Open Research Online
The Open University’s repository of research publications and other research outputs

Design, decisions and dialogue

Thesis

How to cite:

For guidance on citations see FAQs.

© 1991 The Author

https://creativecommons.org/licenses/by-nc-nd/4.0/

Version: Version of Record

Link(s) to article on publisher’s website:
http://dx.doi.org/doi:10.21954/ou.ro.0000dfe4

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online’s data policy on reuse of materials please consult the policies page.

oro.open.ac.uk
Design, Decisions and Dialogue

Ann Blandford,
B.A. (Hons), M.A. (Cantab).

Institute of Educational Technology,
The Open University,
Milton Keynes, MK7 6AA,
U. K.

Thesis submitted in partial fulfillment of the requirements of the degree of Doctor of Philosophy in Artificial Intelligence and Education

September 1991.

© Ann Blandford

Author's number: M7033956
Date of submission: 5th September 1991
Date of award: 24th October 1991
Abstract

This thesis presents a design for an Intelligent Educational System to support the teaching of design evaluation in engineering. The design consists of a simple computer-based tool (or 'learning environment') for displaying and manipulating information used in the course of problem solving, with a separate dialogue component capable of discussing aspects of the problem and of the problem solving strategy with the user. Many of the novel features of the design have been incorporated in a prototype system called WOMBAT. The main focus of this research has been on the design of the dialogue component.

The design of the dialogue component is based on ideas taken from recent work on rational agency. The dialogue component has expertise in engaging in dialogues which support collaborative problem solving (involving system and user) in domains characterised as justified beliefs. It is capable of negotiating about what to do next and about what beliefs to take into account in problem solving. The system acquires problem-related beliefs by applying a simple plausible reasoning mechanism to a database of possible beliefs. The dialogue proceeds by turn-taking in which the current speaker constructs their chosen utterance (which may consist of several propositions and questions) and explicitly indicates when they have finished. When it is the system's turn to make an utterance, it decides what to say based on its beliefs about the current situation and on the likely utility of the various possible responses which it considers appropriate in the circumstances. Two aspects of the problem solving have been fully implemented. These are the discussion about what criteria a decision should be based on and the discussion about what decision step should be taken next. The system's contributions to the interaction are opportunistic, in the sense that at a dialogue level the system does not try to plan beyond the current utterance, and at a problem solving level it does not plan beyond the next action.

The results of a formative evaluation of WOMBAT, in which it was exposed to a number of engineering educators, indicate that it is capable of engaging in a coherent dialogue, and that the dialogue is seen to have a pedagogical purpose. Although the approach of reasoning about the next action opportunistically has not proved adequate at a problem solving level, at a dialogue level it yields good results.
Acknowledgements

Firstly, I want to express my thanks to Eileen Scanlon and Mark Elsom-Cook, my supervisors, for all that they have done to support me in my studies. Both have given generously of their time and talents, and it is my (possibly biased) view that they provide a model of PhD supervising at its best. I cannot thank them enough.

Many other people have also provided valued input to my research through discussions, participating in empirical work, or commenting on written work (or more than one of these). I particularly wish to thank Laurence Alpay, Mike Baker, Jerry Cooper, Kathy Courtney, John Cowan, Nigel Cross, Mark Endean, Olive Francis, Julia Galliers, Jeff Johnson, George Kiss, Diana Laurillard, Brian Parsons, Han Reichgelt, George Rzevski, John Self, Royston Sellman, David Sharpe, Peter Smith and Ken Wallace for helpful hints, provocative questions and practical assistance.

Thanks also to all members of CITE (past and present) for friendship and for contributing to a happy and creative working environment.

Finally, thanks to Chris, Emily, and Laura for being such a tolerant and supportive family.

This research was supported by a postgraduate study award from the Science and Engineering Research Council of Great Britain.
4.3 The outline design of WOMBAT ................................................ 82
  4.3.1 Improving the quality of the current decision ..................... 83
  4.3.2 Improving general decision making skills ......................... 84
  4.3.3 A learning environment to support decision making ............. 84
  4.3.4 The dialogue component ................................................. 88
  4.3.5 The form of the interaction: a discussion .......................... 90
  4.3.6 Forms of problem specification ......................................... 91
  4.3.7 Knowledge and reasoning in WOMBAT ............................... 93
  4.3.8 The iterative nature of the design process ......................... 94

4.4 Summary ............................................................................ 94

Chapter 5: Work on dialogue and agents ..................................... 95
  5.1 Introduction to dialogue ........................................................ 95
  5.2 Purposeful dialogue .............................................................. 101
    5.2.1 Speech acts ..................................................................... 101
    5.2.2 Dialogue games ............................................................. 102
    5.2.3 The generation of purposeful utterances ............................ 103
  5.3 Coherent and symmetric dialogue .......................................... 104
    5.3.1 Coherence and relevance ............................................... 104
    5.3.2 Cooperativity and benevolence ....................................... 106
    5.3.3 Symmetry ...................................................................... 106
  5.4 Discourse participants as agents ........................................... 107
  5.5 Planning versus opportunism ................................................ 108
  5.6 Agent Attitudes ...................................................................... 113
    5.6.1 Classes of agent attitudes ............................................. 113
    5.6.2 Beliefs and knowledge .................................................... 114
    5.6.3 Intentions, goals and commitments .................................. 116
    5.6.4 Desires and values ....................................................... 118
  5.7 Discussion ............................................................................ 118

Chapter 6: The theoretical agent model ..................................... 120
  6.1 Introduction ........................................................................ 120
  6.2 The attitudes included in the model .................................... 122
    6.2.1 Beliefs ......................................................................... 122
    6.2.2 Goals, wants and commitments ..................................... 123
    6.2.3 Values and preferences .................................................. 124
  6.3 The action cycle and goal-action trees ................................. 126
    6.3.1 The action cycle ......................................................... 126
    6.3.2 The goal-action tree ...................................................... 128
  6.4 The decision mechanism ...................................................... 130
  6.5 Illustrations of the operation of the action cycle ................... 131
    6.5.1 Illustration of tree traversal: the goal of having a pleasant evening ......................................................... 131
    6.5.2 Illustration of decision making: meeting a friend in London ......................................................... 132
  6.6 Collaborative activity ......................................................... 136
  6.7 Illustrations of collaborative activity .................................... 137
    6.7.1 Illustration: a protest rally ............................................ 137
    6.7.2 Illustration: country dancing ......................................... 138
  6.8 The dialogue agent ............................................................ 139
  6.9 Discussion ............................................................................ 143
Chapter 7: WOMBAT Implementation

7.1 Overview of the implementation

7.2 The implementation of the dialogue agent in relation to the theoretical model

7.2.1 Implementation of the action cycle

7.2.2 Implementation of the preference mechanism

7.2.3 The dialogue goal-action tree

7.2.4 The teaching strategy derived

7.2.5 The agent state

7.2.6 Summary

7.3 Belief structures and maintenance

7.3.1 The beliefs structure

7.3.2 Beliefs maintenance

7.3.3 Control of the topic stack

7.3.4 Plausible reasoning

7.4 Processing the user's input

7.4.1 Assessing propositions

7.4.2 Making inferences

7.5 Choosing what to say

7.5.1 The implementation of transient goals

7.5.2 Negotiation and conflict resolution

7.5.3 Being proactive: adding to the response

7.6 Collaborative problem solving

7.6.1 The use of mutual working beliefs

7.6.2 Problem solving tactics

7.7 Implementation of the learning environment

7.8 WOMBAT in use

7.8.1 Starting with WOMBAT

7.8.2 The construction of utterances

7.8.3 A sequence of screens from an interaction

7.8.4 Values and utterances: an example

7.9 Discussion

Chapter 8: Empirical test of WOMBAT

8.1 Introduction

8.2 The design of the experiment

8.2.1 Initial experimental setup

8.2.2 The problems encountered in the first experimental setup

8.2.3 Modified experimental setup

8.2.4 Shortcomings of this experimental setup

8.2.5 Final experimental setup

8.2.6 Outcome from final experimental setup

8.3 Example of an interaction

8.3.1 Description of the interaction

8.3.2 The dialogue between P and WOMBAT

8.4 Is the dialogue coherent and sensible?

8.4.1 Incoherences caused by bugs which were subsequently corrected

8.4.2 Features of the current system which lead to incoherent or unreasonable interactions

8.4.3 Summary
Figures and tables

Fig. 1.1: ideal and realised configurations of the IES

Fig. 4.1: matrix as created by group 1, slightly clarified for presentation
Fig. 4.2: windows in WOMBAT
Fig. 4.3: a scaling device for WOMBAT

Fig. 5.1: example of dialogue on car selection

Fig. 6.1: graphical sketch of the operation of the action cycle
Fig. 6.2: example flow diagram
Fig. 6.3: example goal-action tree
Fig. 6.4: entertainment goal-action tree
Fig. 6.5: meeting in London goal-action tree
Fig. 6.6: individual activity in protest march
Fig. 6.7: collaborative activity in protest march
Fig. 6.8: the utterance cycle
Fig. 6.9: the dialogue goal-action tree
Fig. 6.10: collaborative problem solving goal-action tree

Fig. 7.1: the core and the periphery
Fig. 7.2: the dialogue goal-action tree in WOMBAT
Fig. 7.3: detail of the goal of processing the sentences
Fig. 7.4: detail of the goal of addressing transient goals
Fig. 7.4a: t_goal of making agreement explicit
Fig. 7.4b: t_goal of making it known that the system does not know about the proposition
Fig. 7.4c: t_goal of responding to the user's expectation
Fig. 7.4d: t_goal of telling the user that a sentence was incomprehensible
Fig. 7.4e: t_goal of resolving a conflict
Fig. 7.5: detail of the goal of adding to the response
Fig. 7.6: the first screen of the interaction
Fig. 7.7: a dialogue window
Fig. 7.8: the corresponding matrix
Fig. 7.9: the next dialogue screen
Fig. 7.10: the corresponding matrix window
Fig. 7.11: the next step of the interaction
Fig. 7.12: the resulting matrix
Fig. 7.13: matrix showing parameter values
Fig. 7.14: matrix showing scaled (or normalised) values
Fig. 7.15: matrix showing the results of the WOM calculation

Table 7.1: the relevant values and weights for the goal resolve_conflict
Table 7.2: relevance functions
Table 7.3: means-ends beliefs
Chapter 1: Introduction

1.1 The thesis

This thesis addresses an issue of growing prominence in engineering education: how to better equip students to assess designs critically, and to encourage them to view a design as a whole, considering not only its technical merit, but also the requirements of the user of the artifact. The approach taken has been to develop a prototype Intelligent Educational System (IES) to teach the topic of design evaluation. The term 'IES', taken from Cumming and Self (1989), is used in preference to the more traditional 'ITS' (Intelligent Tutoring System) to indicate that the teaching is not to be authoritarian 'tutoring', and that a range of educational interactions between user and system are possible. From an IES perspective, the most important aspects of this thesis are the application of Artificial Intelligence (AI) techniques in a domain where there are no right or wrong answers, the focus on dialogue and negotiation in the context of collaborative problem solving, the separation which has been made between the task-level and the meta-level, and the integration of a learning environment with a dialogue component. In the context of dialogue generation and rational agency, the most important aspects of this thesis are the definition of an action cycle which defines opportunistic activity in a domain in which the agent has (in a narrow sense of the word) expertise. This action cycle integrates decision points with schemata (predetermined sequences of high-level actions) to provide an efficient but flexible approach to activity. The model has been extended to define collaborative activity in which more than one agent participates to reach a mutually agreed goal. The theoretical model has been implemented in the context of an IES which is able to engage a user in collaborative problem solving, and has been demonstrated to work.

1.2 Origins of and motivations for this research

In recent years there have been two notable trends in engineering education. The first is a growing awareness of the need to teach design explicitly (rather than simply assuming that students will 'pick it up' as they go along), and the second is an increasing use of
computers to support teaching. Before embarking on the research reported in this thesis, I had been involved in an ongoing debate on the subject of how computers could and should be used to support design education. Some of this debate centred around the meaning of the phrase 'Computer Aided Design Education': did this mean 'Computer aids to support design education' or 'Education in Computer Aided Design'? My own view was that the former was the more urgent issue to address: that a basic design ability is a prerequisite to the ability to use computers to support the design activity. I believed that applying techniques from Artificial Intelligence in the design of computer aids to support design education offered several potential advantages over conventional computing techniques. These included the possibility of a more individualised interaction between user and system, and also of challenging the user, encouraging her to think more deeply about the problem. Therefore the starting point for this research was an interest in using techniques from AI to support the teaching of engineering design. I was particularly interested in the possibility of addressing issues which are prominent in the earlier 'conceptual' stages of the design process. The particular skills which I wished to address, based on my previous experience of observing students using software in designing, were that of considering the design as a whole and (more importantly) viewing the problem definition and the proposed solutions to it critically.

The topic selected as focusing most on these issues, and also being relatively self-contained so that it could reasonably be dealt with in isolation from the rest of the design process, was the evaluation of design concepts. This involves considering the strengths and weaknesses of the alternative possible solutions to the design problem and selecting one idea to develop in detail. In this topic, questions such as "How good a solution to the problem is each of the proposed design concepts?" and "What are the relative strengths and weaknesses of the proposed concepts?" are considered. For the purposes of this research, and in particular for developing a prototype implementation, the problem of evaluating concepts was reduced to the simpler problem of selecting one from a set of pre-defined concepts. The results of a protocol study together with ideas from the appropriate research literature were used to define the requirements of an Intelligent Educational System to support the teaching of this subject. It was concluded that such a system would need to include both a dialogue.
component (to guide the student and encourage her to view the problem critically) and a facility to allow the user to manipulate data relevant to the current decision. Attention was focused on the requirements of the dialogue component, and on the development of a principled theoretical foundation for the definition of such dialogue components. The questions which featured most prominently at this stage were: "How can a dialogue component be designed which can have some degree of autonomy, while also respecting the autonomy of the student?", "How can a dialogue component decide what to say when it is trying to achieve many things (such as making progress towards a solution, encouraging the user to articulate and reflect on her views and suggesting things which the student might have overlooked)?" and "How can a dialogue component deal effectively with propositions in a domain where there are no right or wrong answers?". The most promising solution was seen as emerging from recent research in the theory of rational agents, and so an agent theoretic approach to the design of the dialogue component has been taken.

The influence of many researchers can be discerned in this thesis. Those whose influence has been personal are included in the Acknowledgements; influential books and papers are listed in the Reference section. The publications which have had the most direct effect on the course of this thesis are:

Pahl and Beitz (1984) and Cross (1989), in discussing design methodologies, and defining the Weighted Objectives Method, as described in §2.2,

Montgomery (1983), in identifying appropriate decision tactics and strategies in a way which provided a base for comparing the results of the first protocol analysis, and which could be implemented (§2.3),

Self (1988), who proposes separating the task level and the meta-level in tutoring systems, and articulates some potential advantages of such a separation (§3.3),

Elsom-Cook (1989, 1990a), who proposes the Guided Discovery Tutoring framework, in which a computer-based tutor and the user have symmetrical access to a learning environment, and who promotes the central importance of the interaction in tutoring (§3.4),
Baker (1989), who articulates requirements of a dialogue component, such as its ability to negotiate (§3.4),

Grosz and Sidner (1986), who discuss the relationship between topic and intention in dialogue (§5.3), and finally

Kiss (1989), whose work on rational agents and the action cycle (§5.5, §5.6) was the basis for the theoretical agent design.

![Diagram](image)

**Fig. 1.1: Ideal and realised configurations of the IES**

Figure 1.1 illustrates the idealised and realised configurations of the system. In this figure, arrows represent lines of communication. The theoretical design of the system resulting from this research is sketched in Fig 1.1a. This design consists of a dialogue component and a learning environment which permits the display of information and manipulation of data relevant to the decision making activity, configured so that the dialogue component and user have symmetrical access to view and manipulate the learning environment. The current implementation, as illustrated in Fig. 1.1b and described in §7,
Design, Decisions and Dialogue

Chapter 1: Introduction

does not permit symmetrical access, in that although the dialogue component and the user
discuss how data should be manipulated, and the user can observe it, only the system has
access to alter the data display; the information display (implemented in pull-down menus)
can only be accessed by the user. The work required to upgrade from the current
implementation to the theoretical design is discussed in §9.5.

1.3 Motivating example

In order to motivate the discussion, a short extract of dialogue between a user (U)
and system (S) is presented, and features of the dialogue highlighted. Most of this dialogue
(S1 - U14) is taken from a genuine interaction between a user (Subject B, see §C.3) and the
prototype implementation which has resulted from this research (but note that the genuine
interaction is not in full natural language). The last few utterances (S15 - U20) are
fabricated. Extensions to the system which would enable it to make these contributions to
the dialogue are discussed in §9.5. The research reported here is concerned with the
development of the theoretical design of an agent which would define how it could
participate in dialogue such as the example presented here. (In later chapters, such an agent
is referred to as a 'dialogue agent'.) It is also concerned with the design of an IES to support
learning about design evaluation which incorporates such an agent.

Two agents, user and system, are working together to select the best design, in this
case for a car, from an agreed set of possible designs, working to an agreed problem
definition. In this case, the car is to be the only car for a family of 4. They are in the process
of agreeing what criteria are important to them in their decision making, and the dialogue
proceeds:

S1: What other criteria do you think we should take into account in making this decision?
U2: Um... I think it should be as big as possible - inside, for the passengers, and also in the
boot, for luggage.
S3: Yes, I agree with both of those. Giving the passengers lots of room means it'll be more
comfortable for them. Now, can you give me a reason for wanting large luggage
capacity?
U4: They'll need it when they go on holidays.

S5: Yes, I agree - because it's to be their only car, they'll probably need it for holidays. Can you suggest any more criteria?

U6: Having a hatchback - flexible carrying capacity.

S7: Oh, that doesn't seem very important to me. I mean, I don't think they'll need that very often - do you? Why do you think they'd need it?

U8: Because I think a lot of families end up doing d.i.y., and they need to carry big things like planks of wood home from the store.

S9: Uhhuh. So you think that's an important reason for wanting a hatchback, do you?

U10: Yes, I do.

S11: Right, OK, I think we've probably got enough criteria to make a reasonable decision now, don't you?

U12: Yes, I think so.

S13: Right, so where shall we go from here?

U14: I don't know...

S15: Well, are all those criteria equally important to you?

U16: No.

S17: So you need some way of reflecting their importance in your decision making. Which criterion do you think is most important?

U18: Minimising price.

S19: So you could reject all the designs which cost more than a certain amount, or you could simply choose the one with the lowest price, or you could assign a system of numerical weights to all the criteria giving minimising price the highest weighting and giving other criteria weightings which reflect how important they are to you. Which action do you prefer?

U20: Tell me more about weighting....

This dialogue has some important features. Firstly, in common with all dialogue (see §5.1) it is constructed jointly; the way the dialogue develops is dependent on the utterances of both participants. It is possible to imagine an almost infinite number of different dialogues
which all started at the same point, and the dialogue is not simply the sum of utterances-made-by-the-system and utterances-made-by-the-user. Each participant is responding within the context of the dialogue, and their utterances are understood within that same context. In particular, the system does not have a plan of how the dialogue is going to proceed, and it decides what to say in context.

Secondly, the system and user have different roles within the interaction. The system is not simply trying to get the decision made - it is also trying to get the student to think about what he is saying. For example, although it agrees (S3) that luggage capacity is important, it still asks the user to justify his suggestion that it should be a decision criterion.

Thirdly, the system is not being prescriptive. It does not behave as if it knows all the answers; it does not have a pre-conceived notion of what the answer should be to any given question, and is able to negotiate to a position of agreement. (U6 - S11).

Finally, it is able to discuss not only aspects of the problem (in this case, decision criteria) but also how the problem is to be solved.

The dialogue component developed in the course of this research has all of these features. As discussed later (§8, §9), further work is required on improving the system's ability to discuss problem solving, but in the current implementation it is able to decide what to say in the context of the preceding dialogue, it takes a guiding and questioning role within the interaction, it is not prescriptive, and it can discuss aspects of both the problem and (in a limited way) how the problem is to be solved.

1.4 Structure of thesis document

As outlined above, the research reported here draws upon past work in many disciplines traditionally considered separate. Relevant literature on engineering design education and decision making is reviewed in Chapter 2, which describes the educational context of this work. In Chapter 3, some aspects of past work on the use of computers to support learning, and in particular the use of AI techniques in Intelligent Educational Systems are presented. Chapter 4 covers a discussion of the teaching context in which this
work is based, starting from the broad context of engineering education then, via the results of a protocol study in decision making, considering the design of an IES to support the teaching of concept selection. (This design covers aspects of the system not implemented in the first prototype.)

Chapters 2 to 4 taken together provide the context for the more detailed theoretical and technical material which is presented in Chapters 5 to 7. In Chapter 5, relevant material from the domains of dialogue and agent theory is discussed. The main theoretical contribution of this research is in Chapter 6, in which the mechanism governing the behaviour of an opportunistic agent which has expertise in participating in dialogue is defined. (Throughout this thesis, the term expertise is used in the limited sense as found in work on expert systems.) Implementation details are presented in Chapter 7. The implementation is sufficiently well developed to demonstrate that the agent definition presented in the previous chapter is implementable, and that such an agent can participate in a coherent, mixed-initiative, opportunistic dialogue with a user in a limited domain. The implementation is also an early prototype for an IES as described in Chapter 4. As such, a formative evaluation of the system has been conducted; the results of this are presented in Chapter 8.

The final chapter (Chapter 9) draws together the threads from previous chapters and presents a critique of this research in relation to the contributing domains.
Chapter 2: An Educational Context

The purpose of this chapter is to outline the background to the thesis, in terms of describing the educational context of the work. The chapter is divided into three main sections. The first of these presents a general discussion on the nature of the design activity, on how design is taught in schools of engineering, and on how computers can support engineering students' learning, with particular reference to the development of design skills.

The particular skills being addressed in this work are those of considering the design as a whole, and of applying judgement in viewing designs critically. The topic within the design curriculum which has been identified as being most appropriate for encouraging the acquisition of these particular skills is that of design evaluation. In §2.2 the topic of design evaluation is reviewed from an engineering perspective. As design evaluation is a type of decision making, §2.3 covers the same topic from a decision analytic perspective. Most of this review serves to provide a background to the thesis. The work of Pahl and Beitz reviewed in §2.2 and Montgomery (§2.3) has been influential in guiding the design of WOMBAT (Weighted Objectives Method By Arguing with the Tutor), as discussed in §4.

2.1 Engineering design education

Design problems come within the class of problems described by Simon (1973) as 'ill-structured'. Such problems are those for which there is no unique solution which can be logically deduced from the problem statement; there are a large number of possible solutions, each of which will satisfy the various constraints on the problem to a greater or lesser degree. To be a solution to the problem, a design must satisfy any absolute requirements (for example, a motor car must be able to transport people!), but there is still no unique solution and there are often conflicting requirements (for example, minimising cost, maximising safety, maximising comfort and maximising performance... simultaneously). So, for example, a car design must conform to all relevant legislation regarding safety standards, exhaust emissions etc.; such legislation defines absolute requirements which the design must satisfy. Car designers also try to optimise their products
to suit particular markets, trading off, for example, performance and internal size against production cost and fuel economy. One of the challenges facing the designer is perceiving the essence of the problem (distinguishing between the 'need's and the 'want's) and developing an appropriate design solution. Correspondingly, one of the challenges facing the design educator is enabling students to acquire the necessary skills.

This section is a review of engineering design education, considering in particular how design is currently taught (§2.1.2) and what roles computer technology can play in design education (§2.1.3). Before issues relating to design education can be reviewed adequately, it is necessary to consider what designing involves, so §2.1.1 gives a brief review of studies of the design process from an engineering perspective.

2.1.1 The design process

In considering how design can best be taught, it is necessary to look at both current design practice and students' understanding of the design process. At the simplest level, the design process may be regarded as any strategy by which a solution can be found to a perceived problem or requirement.

In much of the engineering literature, there is broad agreement (though not necessarily using the same vocabulary) about the stages which a designer, or a design team, goes through. A split can be detected between the attitude of those who advocate methodical design, in which design proceeds within a relatively rigid framework, and those who take a more liberal, or unstructured, view. The former view yields flowcharts such as that presented by Pahl and Beitz (1984) or descriptions such as that of Shahin (1988), who lists the following stages:

- Recognition of need
- Definition of problem
- Feasibility study
- Creative designs
- Evaluation and decision making
- Detailed design
Building and testing of prototype
Designing for production
Product release and market analysis
Development for improvement

Each of the stages identified may encompass any number of sub-stages. This description is referred to by Rzevski (1990) as the 'design as a production line' paradigm. Rzevski suggests that there are historically three dominant design paradigms, the other two being the 'design as a mathematical modelling activity' approach, which assumes that designing entails constructing mathematical mappings from one model of the artifact to another, and the 'design as a rational decision making process' view in which the designer rationally selects one solution from a number of possible solutions. These different paradigms help to define different perspectives on design activity, but none is an adequate definition of that activity.

An example of the less rigid view of the design process can be found in the work of Ullman et al (1988), who have developed a model of the design process (DEAM - the Design Episode Accumulation Model) based on an analysis of audio and video protocols of five mechanical designers working on non-routine problems. Some key features of their model are that a design is constructed by incrementally refining and patching an initial design concept, and that design alternatives are not considered outside the boundaries of design episodes, which are short stretches of problem solving aimed at specific goals. This model probably corresponds more closely to general design practice in the U.K., but is arguably a less appropriate model on which to base design teaching.

Hight et al (1987) review studies which have been undertaken into the mental processes underlying the design process - i.e. how designers think about and structure design problems. The first obvious conclusion of their review is that there is no obvious conclusion! - that different designers appear to adopt different design strategies, and no strong patterns emerge. However, some specific findings are reported:

1) Design methods which attempt to force complete definition of design problems before
allowing them to be solved are intrinsically flawed.

2) Design problems are frequently dealt with in terms of sub-problems.

3) It is impossible to identify simplified 'text-book' design strategies when observing real design activity.

4) A well formulated problem is half way to its solution.

In their studies, Hight et al attempted to evaluate students' problem solving skills when set design problems, and to assess their understanding of the problems and of the design process. One of the assignments their students were set was to list the steps they considered essential to solving any general design problem. The students listed 13 different steps, which the authors categorise under 6 headings as follows:

Stage 1: Problem definition:
- Identify/define problem
- Identify objectives/requirements
- Identify constraints

Stage 2: Preparation for generating ideas:
- Make a plan of attack
- Divide the problem into parts
- Gather information on existing technology

Stage 3: Idea generation:
- Idea generation/brainstorming

Stage 4: Evaluation of ideas:
- Elaboration of ideas
- Evaluation of ideas
- Iteration

Stage 5: Decision making:
- Decide on idea to pursue, or ideas to develop

Stage 6: Implementation:
- Finalise design
- Test product

Their sample was too small (about 110 students at different stages of their studies) to draw very strong conclusions. However, they were able to observe that most students presented with design problems tend to use customised design methods (or strategies) which suit their own modes of working, and that they are generally able to externalise (describe) their strategies. Not surprisingly, they also established that students' ability in tackling design problems improved as they progressed through their studies.
Another study by Radcliffe and Lee (1989) sought to study the design methods of fourteen final year mechanical engineering students using a modified video protocol. Among the conclusions of their study were that

"Most students (11 out of 14) adopted a fairly logical and systematic sequence of design processes. This appears to reflect their innate ability enhanced through experience and possibly through formal instruction in design methodologies. Ad hoc guidance in the form of design outlines and keyword lists was found to be of little help to the students... There was a positive correlation between the quality or effectiveness of a design and the degree to which the student follows a logical sequence of design processes. This suggests that the novice designer should be given explicit guidance on the systematic approach to design tasks [but] this should not be presented as a rigid methodology."

While the perceived process of design may vary from designer to designer, the activities involved in designing (as opposed to the order in which those activities are performed) are in essence the same; there are activities which may be regarded as being 'designerly', and skills which 'any student of design' needs to develop. These include:

**Perceptual skills** - recognising the essential nature of the problem and clarifying ambiguities and unstated assumptions in any problem specification.

**Conceptual skills** - including the ability to generate ideas, making the 'creative leap' from what is currently known to new possibilities.

**Modelling skills** - helping the designer to clarify ideas and to communicate them to others.

**Communication skills** - the ability to work with others (often in a team) is important.

**Analytical skills** - testing proposed solutions against the problem definition.

**Knowledge of possibilities** - including knowledge of how similar problems have been solved in the past and of how sub-problems of the current problem have been solved.

**Knowledge of processes** - including consideration of how the product might be manufactured and maintained, what materials might be appropriate, what
components or sub-assemblies might be bought in and what requirements quality assurance imposes.

These observations have clear implications when considering the development of design aids to support conceptual design, in terms of the rigidity with which designers are constrained in their design strategies; design aids which force the designer to operate in a particular way are likely to be rejected by the majority of designers. They also serve to identify the types of activity which go on within the design process; a recognition of the stages within the design process can be used to provide a focus for the skills and abilities needed by the designer. So, for example, perceptual skills are most important in the stages of defining the problem and evaluating alternative solutions to it, while conceptual skills and a knowledge of possibilities are more important for generating ideas. This understanding, in turn, is needed in seeking to improve the effectiveness of design education.

In the following section, consideration is given to how effectively current teaching methods encourage the development of skills such as those outlined.

2.1.2 Design education

While design education as a part of general education is of growing significance - for example, with the introduction of the National Curriculum in State schools in the United Kingdom - developments in design education at this level have not been a focus of this research. Rather, developments in design education at tertiary level, and particularly in engineering, have been considered.

As a consequence of the two-culture education system which exists in the U.K., a huge gulf has developed between design-as-art (as taught in Colleges of Art, for example) and design-as-science (as taught in Schools of Engineering and other institutions of higher and professional education). The former has little academic respectability and the latter has little creative flair; the graduates of both educational streams are poorly qualified to enter manufacturing industry (Sims 1987).

Steps are being taken towards a more integrated design education in a few
educational establishments, with the development of courses which aim to "create a completely rounded designer with a good, solid understanding of the scientific principles and their mathematical application as well as artistic and creative sensibilities". (Norman and Riley, 1988). These courses are much more design oriented than traditional engineering courses and also much more technology oriented than traditional design courses. My own view is that the graduates of such courses are likely to be well qualified to enter industries which design and manufacture items such as consumer durables, but that a more intensive technological training is needed for those who wish to work in more technologically demanding areas. These courses represent an integrating move from the 'artistic design' end of the spectrum, which needs to be matched by a higher design component in courses from the academic engineering end of the spectrum (Sheldon 1988).

The greater part of an engineer's academic training consists of the learning of engineering science (Harris 1983), with a token introduction to workshop practice and perhaps a passing reference to designing. Simon (1969) proposes the principal reason for this state of affairs as being a hankering after academic respectability:

"In terms of the prevailing norms, academic respectability calls for subject matter that is intellectually tough, analytic, formalizable, and teachable. In the past much, if not most, of what we knew about design and about the artificial sciences was intellectually soft, intuitive, informal and cook-booky. Why would anyone in a university stoop to teach or learn about designing machines or planning market strategies when he could concern himself with solid-state physics?"

Other reasons which I perceive relate to the difficulty of teaching such a poorly understood body of material, and the disincentives offered to academics, whose careers are advanced by research publications and grant awards; no-one becomes a professor by being a good teacher or developing the teaching of a difficult subject. In addition, there is frequently resistance to design education from academics whose view, though rarely articulated, appears to be that "I wasn't explicitly taught, so it's not worth teaching, and in any case it's impossible to teach; students should learn that in industry, 'on the job'." A more openly expressed view is the fear that the students will graduate with an inadequate grounding in
basic engineering science if time is devoted to teaching design. This point is made in graphic
detail (and the view scathingly criticised) by Cawley (1988). The shortcomings of the
present educational system which Cawley identifies include the analytic, or closed, nature of
most problems set to students (i.e. problems with a well defined method of solution and a
unique right answer) and the heaviness of the workload, which encourages a surface
approach to learning.

Within mechanical engineering education, 'design' generally refers to the design of
machine elements, focusing, for example, on the analysis of stresses in a shaft or gear or on
the selection of a suitable bearing for a given application. There is rarely education in total
design, namely design which takes account of all aspects of the problem (including, for
example, a consideration of how the product might be marketed and used as well as its
 technological merit and how it could be manufactured). Indeed, the very existence of subject
boundaries (such as 'mechanical engineering') limits the possibilities of total design (Pugh
1987). A step towards a more integrated approach to engineering design education can be
discerned in recent work on mechatronics, which involves an integration of mechanical,
electrical and electronic engineering, often making use of microprocessor technology as
well. As Pugh (1991) argues forcibly, while rigour in partial design (such as the design of
machine elements) is essential, so is rigour in total design - otherwise industry runs the risk
of perpetually producing brilliantly engineered but commercially disastrous designs.

Cawley (1988) highlights reports from several influential bodies, with
responsibilities for both education and engineering, which state quite clearly that in higher
education, including engineering courses,

"the acquisition of specific knowledge and technical proficiency must go hand in hand
with the stimulation of inquiry and the encouragement of independent judgement and
critical appraisal".

Cawley's concern is not specifically with teaching design, rather with the development of
this type of skill and the acquisition of a deeper understanding of the material learnt. He
proposes that

"some topics, such as the basic concepts of stress and strain, fluid flow and Newton's
laws would clearly be compulsory in an engineering course, and must be covered in some depth, but beyond that, it is largely immaterial which topics are chosen since the major objective is the development of broader skills, rather than the precise subject content."

There is a growing recognition of the need for a change in the approach of engineering educators and more concentration on explicit design education, rather than simply teaching engineering science and hoping that the students will acquire design skills incidentally. Consideration must be given to both how design is being taught now, and how it might be more effectively taught with the growing availability of new technology.

Smith and Kardos (1987) compare and contrast three methods used by teachers of engineering design. The first is the traditional teaching method in higher education, the lecture. Lecturing is often chosen because it appears cost effective, in that one specialist can address a large number of students simultaneously. However, lectures do not provide an educational context which promotes the exploration of ideas. In the lecture environment, students are not encouraged to acquire the sort of insight necessary to enable the creative and original responses needed in design: In addition, the fragmentation of design, as mentioned above, often forces the young designer into inappropriate and compartmentalised thinking about design. Lecturing has its place in design education, to highlight points of principle and give background information, but it is of limited value in teaching such a practical subject.

The other methods discussed by Smith and Kardos are project work and case studies. The majority of people learn most through their own experience; this is an important reason for the use of projects, whether they be short individual pieces of work or longer projects involving teams of students. Another advantage of project work, highlighted by Cawley (1988), is the fact that project work is very similar to engineering practice, whereas lectures and case studies are not. The principal disadvantage of project work is that it is very time-consuming; case studies can be used to complement projects in that they give the student a wider range of experience, albeit second-hand, in a shorter period of time.

Wallace (1987) and Organ (1988) describe the increasing use of project work to
support the teaching of design at Cambridge. This includes extensive projects interspersed with directly relevant lectures which give the background information the students need. Among the difficulties experienced by students which Wallace highlights are grasping the reality of the task, realising that there is no "correct" solution, and appreciating the iterative nature of design work.

Another group who have focused attention on the use of projects in design teaching is SEED (Sharing Experience in Engineering Design), an organisation set up in 1979 to facilitate the sharing of experiences of engineering design education. Their publications include two compendia of engineering design projects (SEED 1988, 1989), including both substantial projects (designed to span most of an academic year and involving an integration of topics) and smaller assignments.

Cawley (1988) proposes a further teaching method, namely the setting of (by implication) well designed problems. He argues that the problem-based approach has the advantage of being more structured than conventional project work, that the problems can be designed to take a relatively short time to solve, and that they can be sufficiently open-ended to require the students to exercise judgement. This is an approach also taken by Cowan (1986), who describes results of work in which students were set problems requiring qualitative reasoning. He argues that such problems demand a higher level of analytical reasoning than the numerical computational problems more commonly set.

All the above-mentioned methods have a role to play in the development of design education. Ultimately, what is needed is an efficient and effective combination of teaching techniques which maximises the benefits of direct experience by giving appropriate and timely guidance, minimises the time wastage that is an inevitable component of inadequately supervised project work, and also maximises the benefits of case studies and other forms of information presentation. As is argued below, there is a role for appropriate computer support in the development of individualised instruction to improve the effectiveness of design education.
2.1.3 Computers in design and design education

Various trends are detectable in the way that engineering is being taught. Firstly, there is a recognition that most recent teaching has concentrated on analysis, or engineering science, to the detriment of the development of other skills which practicing engineers need, and secondly there is a recognised need to address the issue of how to teach design. Thus, it seems timely and appropriate to look at ways in which reasoning skills such as the "stimulation of inquiry and the encouragement of independent judgement and critical appraisal" (Cawley 1988) might be encouraged or enabled, particularly within the context of engineering design.

Many of the skills needed in the later stages of the design process are essentially well defined procedural skills. Within the design process, the application of reasoning skills is of most importance during the earlier conceptual stages of the design process.

There are both pragmatic and pedagogical reasons for considering the possibilities of using a computer to support the teaching of conceptual design. Pragmatic reasons include the observations that computers are becoming widely available in schools of engineering and their use in teaching, in one way or another, is growing. Also, most engineering students do actually enjoy using computers (see for example Taylor (1985) or Burgess and Plank (1988)), which is a motivational consideration (Anderson and Draper 1991). The other side of this coin is that students can become fascinated by the technical capabilities of the computer; one of Cartwright's observations of students using Computer Aided Design (CAD) is that they "are often more interested in using CAD than in what the drawing signifies" (Cartwright, 1988). They also have an apparent tendency to believe whatever they see on the computer screen; there is a subconscious belief that 'if it can be drawn then it can be made, and it must be a good design'. There is a need to develop a more critical perception in students of design, regarding both their designs and the contribution that computing power can make to the design process.

The principal pedagogical reason for considering using a computer in teaching (in an
active sense, rather than in its other useful capacity as a tool to aid thought) is that, given sufficient access to the machine, it has the potential to provide individualised instruction. Particularly when students are working on open-ended problems without a unique solution, it is virtually impossible for one lecturer to give appropriate time and attention to each student, as each has unique difficulties and is working towards a unique answer, which might or might not be a genuine solution to the problem. Diligent but overworked lecturers are easily hoodwinked by student projects which appear well presented but actually omit the consideration of some fundamental points. A related point is made by Taylor (1985), whose account of the results of using the gears program (see below) gives grounds for concern as he says that

"Reports have to be studied very carefully to ensure that the gears ... are exactly to specification, otherwise students could cheat or simply make arithmetical errors. In a recent class of eight students, the two smallest gearboxes were found to be incorrect...

The lecturer has to be thorough and vigilant in his marking."

Taylor seems to ignore the possibility of the computer being 'thorough and vigilant' on behalf of the lecturer; for problems of the size he describes, extending the role of the computer to include checking conformance with the specification is well within the bounds of what is possible today, and was probably possible at the time of the work he reports. Looking ahead, at some time in the future it is likely that each student will have access to a powerful computer tutor which has the necessary information about the design problem, the student and the student's proposed solution(s) to the problem to be able to provide truly individualised instruction. This possibility is even more exciting, if harder to attain, in areas such as design, where each student is working towards a different solution to a given problem, than in areas where there may be different approaches to the problem but there is only one right answer.

An additional argument in favour of using a computer to support teaching is that with appropriately designed software, calculations relevant to the design activity can easily be done without distracting from the main goal.
The use of Computers in the design Process

Computers are being used extensively in the later stages of the design process - principally for routine calculations (often termed 'analysis' in the engineering literature, but generally requiring little analytical thinking) and draughting. Little use is as yet made of computers during the earlier stages of the design process, as these stages are less well understood, and not readily formalizable. Some work has been done on developing 'design assistants' for practicing architects. For example, Gero (1987) describes work on systems which check compliance with specifications, work on systems which refine prototypical designs (the specific example he uses is a retaining wall, which requires information about designer preferences as well as information about the location, such as soil conditions), and work on systems which support design synthesis. Within engineering, most work is being done on the development of expert systems in very limited domains. For example, Ulrich and Seering (1987) report on the use of artificial intelligence (AI) techniques to generate novel designs of a particular type of industrial fastener. Gregory (1987) supports this view, observing that most work to date relates to low-level schemes, with little account being taken of the higher-level knowledge that is needed in design practice. Kenneth Forbus (Forbus 1988) identifies the main shortcomings of current systems as being narrowness (with no 'common sense', and a limited range of solution techniques), uncertain coverage (being developed for specific applications, with no guarantees of their generalizability), and brittleness (inability to degrade gracefully in the presence of incomplete or erroneous information).

Another approach to supporting the designer in the earlier stages of the design process is reported by McCall (1989). He describes a hypertext system called MIKROPLIS which is designed to handle textual information representing the designer's reasoning during the design process. Using this system, the designer is able to structure issues and sub-issues (e.g. if the issue were what sort of gear design is appropriate, a sub-issue might be what material it should be made from), possible answers and arguments for and against. More recently (Fischer et al 1989) this approach has been integrated with a critiquing approach
(see §3.4, Fischer and Morch (1988)) to construct a design environment in which the user can construct her reasoning and have it critiqued by the system. Most of this work has been based in the domain of kitchen design, with the development of a system called CRACK. This approach of encouraging the user to express her reasoning and of critiquing has clear application in a teaching situation; further examples of the critiquing approach as applied in different domains are discussed below (§2.3.3) and in the next chapter (§3.4), and the critiquing approach is compared with the approach taken in this thesis in §9.3.

There are two developing research trends which are worth noting. The first is the approach of treating design as an exercise in constraint satisfaction; in this case the designer effectively searches through a solution space until a design which does not violate any of the defined constraints is found. An example of this approach is the Concept Modeller (Serrano and Gossard 1988), a system based on icons of machine elements which can be manipulated and connected together to form a system, which can then be tested against the constraints defined by the user. I would propose that such a system might be helpful to practicing designers doing fairly routine design, but that it has little to offer in design education. A pocket calculator is a useful tool for numerate people, but it is of little help in enabling children to acquire the basic concept of 'number'. Like pocket calculators and conventional CAD systems, constraint satisfaction systems are tools which, if used well, can be of great help to the designer, but they do little to enable students to acquire design skills.

Another research trend worth noting is the development of what Forbus (1988) terms a 'full qualitative physics', in which a qualitative understanding of physical principles is used as a base for design work. Such an approach seems more in tune with the way that most designers work, and a system including such an understanding is better placed to explain its reasoning and thus holds out more promise for education. One example of this approach is the SOPHIE III system, described briefly below. However, there is clearly a very long way to go before the full potential of such systems is realised, and much more fundamental research needs to be done.
The use of computers to support engineering and design education

Cross (1985) identifies three possible roles for the computer in design education. The first is in training students to use equipment that they are likely to find in industry when using CAD (as described, for example by Cartwright (1988)). The second is as a tool (or design aid) for use in their own projects as described by Organ (1988). The third is in the form of computer tutors to support the central task of teaching students how to design.

Within more general engineering education, it has long been recognised that computers can have a useful role in teaching. For example, the use of simulations (mathematical models) of systems can permit students to focus on developing a feel for the performance of a system under different conditions without having to devote disproportionate attention to the calculations involved. This can be particularly valuable in situations where an experiment on real equipment cannot be performed for reasons of cost, safety or time, or because the quantity, such as the stress in a beam, cannot practically be measured accurately at all points of interest (Smith 1983), (Smith and Pollard 1986). Among the advantages of this use of computers is that students are enabled to learn by experience and develop 'engineering intuition' (a feel for sizes and anticipated performance without the need to always perform detailed calculations).

In the U.K., the principal use of computers to support design education has been in the provision of analysis programs, each of which focuses on one aspect of a design - for example, the deflection of a beam, the stresses in a rotating shaft or the selection of a suitable bearing (Blandford and Smith 1986a, 1986b). The way in which one particular program to analyse pipe networks is used with students at Queen Mary and Westfield College is described by Wormleaton (1986). The work of individual researchers and teachers in other Engineering departments follows the same general lines. For example, Burgess and Plank (1988) describe a program to support students doing structural design, and Taylor (1985) describes work on gear problems and structural design. In both cases the programs are basically analysis tools which allow the students to concentrate more on the important conceptual stages of design. Within the domains that it has addressed, this work
Chapter 2: An Educational Context Design, Decisions and Dialogue

has been effective, but there has been no attempt to produce software to support the teaching of total design or to support the design process prior to analysis.

A different approach is seen in the use of software which supports the teaching of the Open University course T363, Computer Aided Design. The course consists of seven modules, collectively referred to as CADPAC, each of which consists of software and an accompanying study guide which gives detailed instructions on how to proceed, presents relevant background theory (for example, relating the approach used in the course to the way the same issues are dealt with in current commercial software) and delivers teaching material about the design issue in hand. The software and study guides form integrated units, and teach about a wide range of issues, mainly relating to draughting and analysis. The final module goes beyond the bound of traditional CAD in presenting material relating to automatic design synthesis (e.g. printed circuit board layout) and expert systems. This serves to acquaint the user very effectively with what is currently achievable with knowledge-based systems, and also with their limitations. By considering how a limited range of design problems might be solved, the course as a whole presents a thorough and comprehensive view of the possibilities and limitations of Computer Aided Design, as that term is currently understood.

When considering work which has been done on IESs in engineering, two clear foci can be distinguished; some research on the application of Artificial Intelligence techniques in education have taken engineering domains as the focus of their work and, more recently, some engineers, looking at the possibilities of using computers in teaching, have taken existing IES research results and applied them within engineering.

The most significant IES project based in an engineering domain is SOPHIE (Brown et al 1982) which teaches about the troubleshooting of electronic circuits. Earlier versions of the program incorporate a quantitative simulation of the circuit (correct and faulty), and have been used with students (though apparently not extensively). Later versions incorporate a qualitative simulation, which can be used as a base for providing explanations to students; however, the later versions have never been used in a teaching situation. This move from
quantitative to qualitative simulations reflects a recognition that in teaching a measure of human-like reasoning is required in the computer. I would suggest that the same is true in any situation in which effective communication between human and computer on technical matters - e.g. designing - is required.

Other work in engineering has also been focused around simulations for training purposes. STEAMER (Hollan et al 1984) is based on a simulation of the steam propulsion plant of a large ship, and seeks to help the student acquire a mental model of the plant in order to help in operating and troubleshooting the equipment. Similarly, RBT (Woolf et al 1987) is based around a simulation of a recovery boiler, as used in paper mills, and tutors about operating the boiler in both normal and abnormal conditions. RBT is an example of the incorporation of a learning environment (a simulation of the boiler which the user can manipulate to investigate, and thus learn about, its performance) with an intelligent tutor (which provides coaching about the operation and performance of the boiler).

A somewhat different line is advocated by Cox et al (1988), who propose an 'explanation-driven, understanding-directed (EDUD)' approach to the teaching of problem solving, in place of the more prevalent emphasis on the acquisition of domain-specific problem solving skills. They propose that if relevant factual knowledge is presented in an appropriate way then students will be able to solve problems in the domain. Without claiming that the evolution of stages is correct (though it is a good approximation), they present a hierarchy of stages of functionally related information structures which correspond to the student's developing understanding of the domain. The user model (i.e. the system's beliefs about the user) employed to direct the tutoring of the different stages is based clearly on these stages, and three distinct phases of teaching, also based on the stages, are defined. The example which the authors give is parts of a car engine, illustrating how the different levels of knowledge are represented in the system and how material is presented to enhance the user's understanding and her ability to solve problems in the domain.

The IESs described in this section so far support the teaching of engineering, but not specifically design. Some knowledge-based systems to support the teaching of design are
now emerging; these are, however, still largely analysis-based. For example, Kuo et al (1988) describe a system which they are developing called Computer Aided Reasoning and Tutoring in Engineering (CARTE). They claim that it is suitable for tutoring about engineering design, and illustrate its use in the tutoring of offshore engineering. Knowledge about the domain is stored in a tree structure as a logical sequence of topics, each of which has several tutorial modules. It provides information to the user in both textual and graphical forms, and sets problems to test the user's understanding. The domain is treated as being well behaved, with right-or-wrong answers to problems. The system is knowledge-based in terms of the way engineering knowledge is stored and used, but does not claim to store or use knowledge about teaching strategies or to model the user.

Slater and Ahuja (1987) describe an Intelligent Computer Aided Instruction (ICAI) system called MACAVITY which deals with the application of static equilibrium and elementary strength of materials to beams. This program tutors students on the calculation of reactions in beams, and the construction of bending moment and shear diagrams. It deals with teaching students how to solve well defined problems (which may be defined by either student or program) for which there is a unique correct solution. The authors briefly mention work on the development of further ICAI programs in the same vein to teach about the analysis of trusses and the design of steel beams.

**Summary**

This section has reviewed work on the application of computers to the teaching of engineering design, with particular reference to software developed specifically for teaching (as distinct from the application of commercial or industrial software in an educational setting). With the exception of the CADPAC software, which supports teaching about CAD, the focus of all the work reviewed has been on the analysis of designs, with attention placed mainly on the performance of components, such as gears or beams, or structures, and little consideration being given to total design. It is clear from this review that while computers have already offered much to engineering education, that work which has been aimed at teaching design has focused almost entirely on analysis (i.e. on the system performing well
defined calculations), and on single components rather than complete systems, and no work of significance has been done on the teaching of conceptual design using computers.

2.2 A focus on design evaluation

In considering the possibilities of developing a computer tutor to support the teaching of design, attention has to be focused on one small aspect of designing. Let us consider the early stages of the design process as described by Shahin (§2.1.1). This view of the design process is one with which some, though by no means all, engineering designers are likely to identify. It does, however, provide a useful focus for discussion.

Two of the earlier stages of design which can be clearly separated out, and are relatively well understood, are 'definition of problem' and 'evaluation and decision making'. Both of these are stages which address the issues of viewing the design as a whole and applying skills of judgement and critical appraisal. Any such isolation of one component is inevitably artificial as stages are not independent of each other. However, the development of any integrated, coherent teaching system for engineering design needs to teach these aspects of design - and any system without such a capability is missing something completely fundamental. There is currently an inadequate understanding of, for example, how world knowledge (or common sense) can be encoded usefully in a system and of how creativity can be fostered effectively. This current lack of understanding imposes limitations on the development of an IES to support stages such as doing feasibility studies and generating creative designs.

Noting the observation made earlier (§2.1.1) that design methods which attempt to force complete definition of design problems before allowing them to be solved are intrinsically flawed (the problem definition is built up as understanding of the problem develops), it was decided to pursue the idea of developing an IES to support design evaluation. There are, paradoxically, fewer problems associated with the isolation of design evaluation, because an evaluation is a discrete activity, rather than taking place over a longer period of time. It is unreasonable to expect a student to come 'cold' to a problem and assimilate its true nature or assess the quality of a proposed solution to it, but provided that
the design problem and the proposed solutions are familiar to, or easily assimilated by, the student it is reasonable to expect an evaluation of it. This condition is satisfied in situations where, for example, the evaluation is a stage in a major student project or where a directly relevant case study has been undertaken. In a realistic design project, the evaluation would often be followed by modifications to either the problem definition or the proposed solutions as the designer's understanding of the problem develops, but this fact does not alter the status of the evaluation as a discrete activity.

To conclude, the topic being used in this research is that of design evaluation. This topic has been selected because it addresses the educational issues of encouraging students to take a total design approach to the design activity, to deal with open-ended problems, and to develop skills such as judgement and critical appraisal. It is also perceived as raising interesting issues while being tractable within the time scale of the research project. The issues raised include how to develop a tutoring system which can guide appropriately while allowing the student sufficient freedom, and which can accommodate different lines of reasoning and accept that answers are neither right nor wrong.

2.2.1 What is meant by design evaluation?

Jones (1963) observes that

"the traditional method of evaluation of engineering designs is by judgement, and by reference to the experience of engineers and draughtsmen, while the design is on the drawing board. When the right kind of experience is available, and when logical methods of detecting errors are too expensive or time-consuming, this is still the most effective method.... We are, however, already approaching a situation where engineering is called upon to design and develop increasingly complex equipment of which little or no experience exists and for which engineering drawings do not provide an adequate means of evaluation."

Jones uses the term evaluation to refer to just one design, and the assessment of whether or not that design meets the specification of the problem. In his presentation of the subject of design evaluation, he advocates that all design requirements be stated in absolute
terms (e.g. 'distance between A and B to be less than X', in preference to 'minimise distance between A and B') so that assessment is straightforward. This takes no account, however, of the relative nature of some design requirements and the absolute nature of others.

A similar approach can be found in the CADPAC software described above, particularly in the kitchen design module (CADPAC2). This module teaches about design evaluation in black-or-white terms; all constraints are presented in absolute terms, and the evaluation consists simply of a list of the constraints which are violated. For example, one constraint (which is, and should be, absolute for safety reasons) is that the distance between the sink and the cooker must be at least 1000mm. A second constraint is that the same distance must be less than 1800mm; as this is more a matter of convenience for the user of the kitchen, it would be more appropriate to impose a relative constraint that the sink to cooker distance should be minimised (but at least 1000mm). In doing the set design exercises, it is possible to develop second-rate designs which satisfy all the constraints and rather better designs which are rejected as not satisfying all the constraints. This module leaves the user with the impression that all constraints were absolute, and does not present any means of distinguishing between designs which satisfy the constraints.

A more satisfactory approach is taken in CADPAC7, which also deals with kitchen design using a 'problem search space' method to generate a large number of designs which satisfy the absolute constraints of the problem. In this case, some of the constraints are dealt with as being relative, and the best solutions are stored for further consideration by the user. In this case, a solution is counted as one of the best if it is better in at least one respect than all the solutions already stored as best; a stored solution which is worse in all respects than the newly stored one is then discarded. At the end of the search for solutions which satisfy the absolute constraints, the best solutions are presented to the user, but no suggestion as to how to choose between them is presented.

In what follows, a comparative use of the term 'evaluation' is used (i.e. alternative solutions are evaluated by comparing them against each other as well as against the problem
Chapter 2: An Educational Context

A distinction is drawn between absolute and relative constraints. Any proposed solution to a design problem which violates any absolute constraints of the problem specification is rejected as not in fact being a solution to the stated problem. If more than one acceptable solution to the problem is found, then a mechanism has to be developed for selecting the most favoured solution. Pahl and Beitz (1984) distinguish between the 'demands' and 'wishes' of the specification, using these terms in the same sense as the terms 'absolute' and 'relative' constraints are used above. They express this as follows:

"concept variants which do not satisfy the demands of the specification have to be eliminated and the rest must be judged by the systematic application of specific criteria based on the wishes of the specification. On the basis of this evaluation the best solution concept can now be selected."

It is in this sense that the term 'evaluation' is used here.

2.2.2 Is formal evaluation worth the effort?

It must be stated that, in the U.K. at least, there is a long (undocumented) tradition of evaluation by intuition. This may work well for expert designers who are confident in their understanding of the design problem in hand and in the performance and attributes of the proposed solutions to it. Less experienced designers presumably acquire this intuition through a combination of observing experts at work and trial-and-error. Just as there is a recognised place for teaching formal design methods (even if students are not expected to use them rigorously as practicing engineers), there is also a place for teaching formal evaluation techniques to help trainee designers get over the hurdle of developing the necessary intuition, as well as providing them with a tool to apply when intuition fails. As Hight et al (1987) observe, basing teaching around such a methodology also makes it easier for the design educator to organise the curriculum.

Formal evaluation is becoming more necessary in industry, just as more explicit formal design methods are becoming more essential, because of technological advance and the increasing complexity of products. Also, the trend towards larger design teams who must communicate effectively and reach agreement at every significant decision point means
that techniques must be employed to articulate and justify decisions taken. An explicit decision making procedure can clarify priorities, expose biases and preconceptions and challenge assumptions, as well as emerging with a justifiable, explainable decision. Cross (1989) expresses this:

"Choosing between alternatives is a common feature of design activity. Choices can be made by guess work, by 'intuition' or by arbitrary decision. However, it is better if a choice can be made on some more rational, or at least open, procedure. Not only will the designer feel more secure in making the choice, but others involved in decision-making, such as clients, managers and colleagues in the design team, will be able to participate in or assess the validity of the choice."

Various papers explicitly use evaluation techniques as they present their view of the design process as applied to the development of conceptual designs for specific applications. However, the basic assumptions and the justifications for decisions taken (regarding basic evaluation criteria and their relative importance) are rarely presented explicitly. For example, Pighini et al (1983) outline the process they followed in establishing the optimal (their word, not mine) dimensions for a city car, and Shahin (1988) describes the complete process followed in the design of a swivel joint for a particular (underwater) application. While in some ways, Shahin presents a detailed account of the process, the decisions he takes in selecting and weighting his evaluation criteria are obscure, and hard to understand; for instance, he gives different weighting factors to different components of the total cost, whereas it would seem intuitively obvious that what matters is total cost, rather than the way that cost is made up. Having said that, Shahin's analysis may be difficult to comprehend, but at least his figures are open to inspection and question, unlike any selection based upon intuition.

2.2.3 Methods of evaluating designs

While some authoritative texts on engineering design, such as that by French (1985), devote minimal space to the subject of evaluation beyond acknowledging that it is necessary, others present a variety of approaches. While these approaches vary in the level of detail
presented and in their principal focus, a common basis can be identified in most of them.

Asimow’s approach - making critical decisions

Asimow (1962) emphasises the great importance of what he terms ‘critical decisions’ - i.e. those which have a strong bearing on the future direction of a project. He briefly describes the use of a matrix of value ratings $V_{ij}$, where $V_{ij}$ is the value rating of criterion $C_i$ for alternative $A_j$ (where the criteria on which the decision is to be based are $C_1, C_2, \ldots, C_m$, and there are $n$ alternative designs $A_1, A_2, \ldots, A_n$). As a first choice, the alternative with the best set of ratings would be selected for further development. However, this solution might be very difficult to implement; this issue occupies Asimow at length. He notes that any solution which does not contravene the laws of nature is achievable, given sufficient time and resources, and proceeds to discuss the assessment of the realisability of a selected solution within a certain time and with limited resources, and the confidence rating attached to that assessment (i.e. how sure the designer is that this design can be developed within the time allowed and with the available resources). He offers a method for selecting the design to proceed with based on the advantages offered by, and the confidence rating of, each alternative.

Among his conclusions are that:

1) Critical decisions must be treated as being final.
2) The decision rests principally on a comparison of the advantages and difficulties associated with each proffered solution.
3) Designs are constrained principally by time and budget.
4) The penalty for failure must be taken into account when reaching a decision.

The approach of Pahl and Beitz - the Weighted Objectives Method (WOM)

Pahl and Beitz (1984) outline a basic evaluation procedure incorporating the concepts of use-value (or cost-benefit) analysis, and of the German technical guideline VDI2225. The two approaches, which vary mainly in the level of detail they present, are compared and combined. Initially, the method is presented in meticulous detail without reference to any
particular decision making instance. The issue of selecting between concept variants is then
dealt with as a specific instance of the application of the method. In brief, the method is:
1) Check that each concept satisfies the minimum requirements to be a solution to the
problem.
2) Select evaluation criteria. These are derived from objectives which will be based
mainly on the specification, and on general constraints. The objectives must be
identified to take into account all essential criteria, to be as independent of each other
as possible, and to be clearly expressible, preferably in quantitative terms.
Evaluation criteria can be derived directly from the objectives and must be stated in
positive terms (e.g. 'low maintenance cost', rather than simply 'maintenance cost' -
i.e. a higher value must be better, not worse!). The construction of an objectives tree
is described.
3) Assign weighting values to take into account the relative importance of the criteria
selected. It is suggested that weighting values should be selected such that the sum
of all weighting values is 1 or 100, so that it is clear what proportion of the total
weight is being assigned to a particular criterion.
4) Parameters need to be selected to measure each criterion; e.g. the parameter for 'low
fuel consumption' would be 'average fuel consumption in km/litre'. In the case of
non-quantifiable criteria, an appropriate range of adjectives, ('hopeless', 'average',
'very good' etc.) would need to be defined.
5) Values in a fixed range (e.g. 0-10) then need to be assigned to given ranges of each
parameter, so that each criterion is being assessed on the same scale. The values
must then be assigned to each concept variant.
6) The overall value of each variant can then be calculated simply by summing the
products of weights and values:
\[ V_j = \sum_{i=1}^{n} w_i v_{ij} \]
Thus the relative merits of the variants can be compared.

Pahl and Beitz present further issues such as the use of technical and economic
ratings, and the detection of weak spots in designs. They also illustrate the use of the
method with worked examples.

**Rittel's approach - evaluation from a planning perspective**

Rittel's evaluation procedure for individuals (as distinct from groups), discussed by Grant (1976b) is as follows:

1) Try an off-hand overall judgement;
   a) if you are happy with it and do not have to be able to explain it, stop;
   b) if you are not satisfied with it, or couldn't make it, or have to be able to explain it, go on.

2) List the parameters or aspects of the decision that are important.

3) Choose an aggregation function.

4) Weight the parameters or aspects.

5) Score the alternative courses of action against the aspects or parameters:
   a) by means of off-hand judgements,
   b) if not satisfied then:
      i) break the aspect/parameter down into several smaller sub-aspects/sub-components and try again,
      ii) construct a carefully deliberated criterion function, or
      iii) establish a procedure for testing, simulation or measurement in order to score the proposals against those aspects/parameters where appropriate.

6) Apply the aggregation function chosen in (3) to combine the partial judgements into an overall judgement.

7) If an off-hand overall judgement was made in (1) compare the results in (6) with it.

A comparison of Rittel's approach, which is presented from an architecture/planning perspective, and that of Pahl and Beitz (an engineering perspective) shows broad agreement on the general approach to evaluation. The same basic approach is advocated by Cross (1989).
The approach of Pugh: Controlled Convergence

One author who is less enthusiastic about such quantitative approaches to design evaluation is Pugh (1991), who argues that they "attempt to impart to the procedure too much precision, and thus inhibit qualitative judgements." In his description of the method of 'controlled convergence', Pugh makes explicit the iterative nature of the design process, and particularly the stage of design evaluation, involving as it does modifications to the problem definition and proposed alternative solutions as understanding of the problem develops. In the process of controlled convergence, each stage of design evaluation involves reducing the number of candidate solutions by removing the weakest from further consideration, then new concepts are generated and added to the next evaluation cycle. The number of candidate solutions gradually reduces until just one is left as the selected concept. His evaluation process involves (in brief):

1) identifying the criteria on which the decision is to be based,
2) selecting one alternative solution (typically either one which is already on the market - e.g. a competitor's design - or the one which is thought to be the 'strongest' candidate at this stage) as a datum,
3) comparing each alternative against the datum on each criterion, and allocating '+' if the alternative scores better on the criterion, '-' if it score worse, and 'S' if it scores the same, and
4) assessing the individual concept scores, and considering in what ways the weaknesses of the concepts might be ameliorated.

Step (4) is used as a basis for both eliminating weak solutions and generating new concepts. Pugh cautions against the temptation to add up the numbers of '+'s and '-'s and use them in a quantitative way. It might be argued that the application of controlled convergence is not dependent on whether the evaluation procedure is qualitative or quantitative - that the main argument against a quantitative method is the inappropriate confidence in the solution that the use of such a method can instill in the designer. What is really at stake is not whether a qualitative or quantitative approach is better, but the
understanding which the designer has of the significance of the outcome of either decision process.

**Summary**

Grant (1976a) discusses the arguments for and against the use of Weighted Objectives. He observes that

"Some people feel that the whole concept of weighting objectives is not valid. Other people feel equally strongly that the weighting of objectives in one manner or another is unavoidable in the course of decision-making, and that it is better to meet the problem head-on with conscious deliberation than to default on the judgement through ignorance or a reluctance to face the problem."

He also warns against the risk of reading in unjustified information content or unjustified precision when numbers are applied and manipulated in any evaluation method. Among the arguments that Grant identifies in favour of weighting objectives are that

"it seems to approximate the kind of thinking that actually goes into human decision making activities, whether or not it is theoretically resolved; and that the process of deliberating relative weights, among other things, focuses one's awareness on the problem and stimulates reflection and insight. [It also fulfills] a desire to render decision bases communicable, recordable and arguable."

There is a clear division, as noted by Grant, between those such as Pahl and Beitz who advocate formal numerically based methodologies in evaluating designs and those such as Pugh who consider such methodologies to be inhibiting. The approach advocated by Pugh reflects his view of the design process as essentially iterative, and as such is a more difficult approach to accommodate in an IES which teaches only about design evaluation. In the following section (§2.3) other approaches to selecting between alternatives (as a substantial component of design evaluation) are outlined. The approach being taken in this thesis is not to be prescriptive and impose a particular evaluation methodology on students, but to discuss the relative merits and disadvantages of different evaluation strategies. However, as will emerge in the description of the implementation (§7), the only strategy
which has been fully implemented in the prototype version of WOMBAT is the WOM as outlined by Pahl and Beitz.

2.2.4 Current approaches to the teaching of design evaluation

Few accounts of engineering design education present sufficient detail to give a clear view of the current approach to the teaching of evaluation. The account of the use of design projects by Organ (1988) includes a fleeting mention of "Techniques for design evaluation" among the 10 topics covered in six lectures on design methods, but this point is not expanded upon. In the Open University course T363 little attention is paid to the issue of design evaluation or optimisation, beyond the mention that it is a part of the design process. The approach taken in the CADPAC software is documented above (§2.1.3).

The subject is mentioned in the Compendium of Engineering Design Projects (SEED 1988) in the following context.

"The main mode of learning through the design activity involves the following phases, with appropriate iteration:

a) From the project brief: discovering, accessing and assessing relevant data, literature, theory and advice.

b) Specifying the design requirements accurately in the Problem Design Specification.

c) Generating concepts and evaluating the most promising based on the criteria/design requirements specified.

d) Developing the detailed engineering of the chosen concept.

e) Communicating the proposal convincingly."

The importance of evaluation is recognised in several of the project briefs included in the Compendium, with phrases such as "Evaluate the four (or more) solutions as objectively as possible", "introduce students to methods of decision-making in design", and "choice of the best solution, with appropriate justification".

Thus there is an acknowledgement of the importance of evaluation within the design process, but with little documentary evidence of how the issue is dealt with in a teaching
situation. As will emerge in §8 in the discussion about the potential value of WOMBAT in an educational setting, the engineering educators questioned had nothing really to compare WOMBAT against. None of them had made use of computer based tools (such as a spreadsheet) to support their teaching of design evaluation and only 1 of the 4 subjects who teach design in a traditional (not distance learning) university dedicates teaching time to the topic of evaluation.

2.2.5 The use of computers in selecting between alternatives

Reporting on software developed specifically to support the activity of selecting between alternative design concepts, Sodhi (1985) gives an account of a micro-computer based application program which essentially implements the WOM principle, though the formulae he uses are slightly different from those outlined above. He presents two particular examples of the use of his program in selecting a manufacturing process for a shaft (selecting between machining, sleeving and forging) and selecting the most appropriate material for making the skin of an aircraft. He proposes that such a selection process could be an important part of a CAD system, with access to data bases etc. Sodhi also suggests that a sensitivity analysis, to determine how minor variations in the weightings would affect results, should be incorporated. This is a simple decision support program with no pretensions to intelligence, and only a primitive user interface.

Arafat et al (1990) describe a design evaluation system comprising a knowledge base containing domain specific knowledge together with an evaluation component whose design is based on Multi-Attribute Utility Theory (MAUT, the decision analyst's term to describe the process referred to as the Weighted Objectives Method above). Their system is intended to be used in two phases; the first involves the construction of the knowledge base by domain experts, and the second involves the use of the system to evaluate design concepts. The authors have concentrated so far on the domain of engineering structures, which is perhaps one of the better understood engineering disciplines (in terms of being able to identify appropriate evaluation criteria).
2.2.6 Summary

Along with the growing awareness of the need for design education within engineering courses, there is a growing awareness of the value to students of understanding and being able to apply design methods, which includes the application of techniques for evaluating designs with a view to selecting one for further development. In considering the development of an intelligent tutoring system to support the teaching of conceptual design, attention has been focused on the possibility of teaching design evaluation. There is general agreement (if variable vocabulary!) among authors who present methods of evaluating designs about the general principles underlying such methods, and even to a remarkable degree on their implementation.

There are two distinct purposes of teaching about design evaluation. Firstly, there is the development of a procedural skill - namely how to make decisions of this type. Secondly (and I would argue much more importantly) there is the development of skills of perception (the ability to perceive what are the important aspects of a design specification) and judgement (the ability to assess the relative importance of the various criteria identified) in students of design.

A little work has been done on developing software to support design evaluation, but none on software to support teaching of this topic. In this respect, this thesis is exploring fairly virgin territory.

2.3 Decision Making

In the previous section (§2.2), the subject of design evaluation, with particular reference to the selection process involved, has been reviewed from an engineering perspective. In this section, the same subject is reviewed from a decision analytic perspective. In considering how the quality, reliability and consistency of decision making might be improved, both prescriptive and descriptive research have been undertaken; the first considers how people should make decisions, and the second how they actually do. The relevant work on prescriptive approaches is reviewed above (§2.2); the same approach
is outlined in texts on decision making such as that by von Winterfeldt and Edwards (1986). In this section descriptive research is outlined. Most of this section is devoted to an account of studies of human decision making. This is followed by a brief account of software designed to support selection between alternatives or designed specifically to support learning about decision making. The research results on how people make decisions are used as a base for analysing the protocols collected to inform the design of WOMBAT (§4). The outline design can be compared with existing systems which serve similar functions.

### 2.3.1 Intuitive and analytical decision making

Intuitive and analytical modes of thought are generally viewed as distinct types of thinking. Intuitive thought is often regarded as any non-analytical mode of thinking. Hamm (1988) describes it as generally involving rapid, unconscious data processing that combines the available information by averaging it, has low consistency and is moderately accurate. In contrast, he describes analytical thought as slow, conscious and consistent, usually quite accurate, but occasionally producing large errors. It is likely to use organising principles more complicated than averaging. Most thinking is neither purely intuitive nor purely analytical, generally combining features of both. Where more analytical cognition is employed (and the task is more structured) there is a greater possibility of variable manipulation by the person exercising judgement and the process is more visible (open to inspection or replication by others). However, greater time and resources are required for the decision making process.

Considering the factors which influence how decisions are made, Hammond (see review by Hamm (1988)) proposes that task features influence the mode of cognition which the thinker will adopt as follows.

- **Complexity of task**: number of cues in the task definition, redundancy of cues and identity of an accurate organising principle all influence the mode of cognition adopted. The existence of more cues or redundant cues is likely to induce intuition. If a complicated procedure is known to give the most accurate result then it is likely to be used, whereas if a simple weighted average organising principle is known to give
good results then intuition is more likely to be used.

• Ambiguity of task content: if a complex organising principle is known then it will be used (analytical thinking). Unfamiliarity with task content induces intuition, due to the unavailability of a complicated organising principle. A known possibility of high accuracy from using analysis encourages analytical thought.

• Form of task content: if the task is presented in a form which encourages decomposition into subtasks, this encourages analysis. Pictorial cues encourage intuition; numeric cues encourage analysis. The time available for completion of the task influences the mode of cognition (urgency encourages intuition).

One of Hammond's hypotheses is that the accuracy of cognition depends (in part) on whether the appropriate mode of cognition is selected. Empirical evidence to support his hypothesis has been found in a study of highway engineers (Hammond et al 1987), in which both the deep (content of task) and the surface (presentation of task) task features were varied independently, in order to study the effect of the task characteristics on the mode of cognition. While it is considered important to recognise that the form of task presentation will influence the way people approach the task, this issue is not a focus of this research. However, one conclusion which can be drawn from Hammond's work - that the development of expertise involves learning any complicated principles which improve performance on the task - is of significance to the research.

Stuart and Hubert Dreyfus (see (Hamm 1988)) consider changes in the use of analytical and intuitive thinking in terms of the development of expertise. They propose five stages to becoming an expert:

novice, who must think analytically in order to perform,
advanced beginner, who has learned to perceive intuitively but must still apply rules to know how to act,
competent, who exercises both perception and action components of skill intuitively, but must still think analytically about the whole situation,
proficient, who perceives the whole situation intuitively, but must still make decisions analytically, and finally,
expert, who makes decisions intuitively too.

The Dreyfuses note that students must realise that expertise is acquired in stages, and must avoid trying to think like an expert, for without experience based on analytic foundation, intuitive performance will be poor. Not using rules is the privilege of the expert, not a route to becoming expert more quickly! Instead, students should practice using the rules and logic which are available, and not depend on inappropriate decision aids, as they must develop expertise in judgement and decision making.

The Dreyfus' theory is not presented at a level of detail which makes it useful in the current analysis, and while it is intuitively (!) appealing, work has not been done to validate it. In the context of this thesis, the main point to be noted is that students are assumed to be novice decision makers, who must (according to this theory) think analytically in order to perform.

2.3.2 How people make choices

Tversky (1972) presents a theory of choice (or selection) based on an elimination process, in which each alternative is viewed as a set of aspects (factors). At each stage in the selection process, the most important remaining aspect is used as a basis for elimination or retention of each alternative. This process continues until only one alternative remains. For example, in selecting a car, the decision maker might eliminate all those which cost more than X, then all which only have two doors, then all estate cars and hatchbacks, then all with a poor reliability record, etc. until only one alternative remains. Tversky presents empirical evidence to support his theory, illustrates the logic of this decision making strategy, and identifies its appeal as relating to the ease of applying the strategy and of defending the decision outcome. He observes that optimal decision making strategies, which generally involve computations based on the weights assigned to the various relevant factors or on the compensation rates associated with critical variables, involve the assimilation and manipulation of overwhelming amounts of relevant information, which exceed the human's intuitive computational facilities.
In addition, Tversky suggests that people are reluctant to base decisions on computations involving subjective assessments of likelihood or value in which the subject only has limited confidence. People prefer a clear-cut choice, without relying on an estimation of relative weights, or on numerical computation. While the decision taken might be easier to justify using a strategy such as elimination by aspects, however, the quality of that decision is likely to be lower than one based on a more complicated compensatory strategy such as the WOM.

Montgomery (1983) proposes the idea that "decision making is a search for good arguments" - i.e. that people want to have easily understandable, justifiable reasons for making the decisions that they do. He outlines a decision making process which can be viewed as the search for a dominance structure; that is, a cognitive structure in which one alternative can be seen as dominant over all others. Within this process, he identifies a range of decision rules which might be applied at different stages of the process:

**Dominance rule:** choose alternative A₁ over alternative A₂ if A₁ is better than A₂ on at least one attribute, and not worse on all others.

**Conjunctive rule:** choose only alternatives which exceed or are equal to all of a set of criterion values on the attributes.

**Disjunctive rule:** choose only alternatives which exceed or are equal to at least one of a set of criterion values on the attributes.

**Lexicographical rule:** choose alternative A₁ over A₂ if it is better (or significantly better) on the most significant attribute. If this requirement is not fulfilled, base the choice on the most attractive aspects of the attributes next in order of importance.

**Elimination by Aspects rule:** exclude all alternatives which do not exceed a criterion on the most important attribute. Repeat this procedure with new attributes in order of importance. This rule is the one described in greater detail by Tversky, and is in essence the rule applied to eliminate proposed solutions which do not satisfy absolute criteria specified in the problem statement. (First step in the WOM as described above.)

**Maximising number of attributes with a greater attractiveness rule:** choose A₁ over A₂
if A₁ differs favourably from A₂ on a greater number of attributes than the number of attributes on which A₂ differs favourably from A₁.

**Addition of utilities rule**: choose the alternative with the greatest sum of (weighted) attractiveness values (utilities) across all attributes. This is the rule used in the main part of the WOM as described above.

**Addition of utility differences rule**: add differences \( D_k = f(a_{1k} - a_{2k}) \) where \( a_{jk} \) is the attractiveness of aspect \( jk \) for alternative \( j \) and attribute \( k \), and \( D_k \) is a continuous function of \( (a_{1k} - a_{2k}) \). If the sum of these differences is positive then choose A₁, and if negative then choose A₂.

The first five of the above rules are non-compensatory (i.e. they do not allow an unattractive aspect on one attribute to be compensated by an attractive aspect on another). The remaining three rules are compensatory, allowing drawbacks and advantages of different attributes to be integrated into a total attractiveness measure.

Montgomery identifies the problems of non-compensatory rules as being their limited applicability (they do not always yield a unique solution) and the risk that using them will involve the neglect of important information. The problems of compensatory rules which he identifies are the fact that they tend to require complex value judgements, that it may be difficult to gain a good overview of arguments based on compensatory rules, that the overall attractiveness measures which result from applying compensatory rules may be viewed as being too abstract, and that compensatory rules emphasise the fact that one has to give up certain good things in order to get other good things. Thus, Montgomery's view of the relative merits of compensatory and non-compensatory rules matches well with that of Tversky. In essence, compensatory rules generally yield better decisions, but they are more difficult for people to cope with, both cognitively and affectively (emotionally).

The search for a dominance structure, in the view of Montgomery, goes through four stages:

**Pre-editing**, in which attributes for inclusion in the dominance structure are selected and evaluated, and alternatives are screened. Montgomery notes that at the time he was
writing, and apparently also today, no decision rule has been offered in decision making research for how people find or select attributes, or rank their importance. Research suggests that initial screening of alternatives consists of discarding unfavourable alternatives, often using elimination by aspects or the conjunctive rule, though an alternative which would naturally be eliminated using these rules might be kept if it were very attractive in some other aspect, indicating the application of a compensatory rule.

**Finding a promising alternative**, often employing the disjunctive, lexicographic and elimination by aspects rules.

**Dominance testing**, which consists of trying to find a dominance structure such that the chosen alternative is better than all other alternatives on at least one aspect, and not worse than any other alternative on all other aspects. If, as will often be the case, such a dominance structure cannot initially be identified, then it is necessary to do...

**Dominance structuring**: This involves eliminating or neutralising all violations of dominance identified in the dominance testing phase. Tactics employed include:

- **de-emphasising**, in which the decision maker de-emphasises the importance of an attribute on which the promising alternative is weak,
- **bolstering**, which involves enhancing positive aspects of the promising alternative, or enhancing negative aspects associated with non-promising alternatives,
- **cancellation**, which involves discarding two attributes where the positive aspect of one attribute is viewed as being cancelled out by the negative aspect of another for the promising alternative. This requires some compensatory thinking, but does not require more precise value judgements on the relative merits of the two aspects in question, and
- **collapsing**, which involves incorporating two or more aspects into one new aspect - for example, redefining aspects in terms of monetary value (so-called cost-benefit analysis). In a sense, the WOM can be viewed as collapsing, in that all attributes are allocated a measure of utility.
Montgomery does not propose his process model as being definitive, but presents it as a structure on which to base further research (see for example (Montgomery 1989)).

Taking a different approach, Huber (1989) presents a model of decision making as problem solving, in which he analyses many of the rules described by Montgomery in terms of operators which act on the problem space to transform it from the initial (problem statement) to the goal (decision made) state. Huber's operators describe the decision making activity at a much finer level of detail than that used in the research reported here, but his approach - of viewing decision processes as problem solving processes - is in tune with the approach taken in this research.

In this thesis the decision rules (or tactics) described by Montgomery are used as a starting point for analysing the decision tactics applied by subjects in a protocol study of decision making (§4.1). The results of this (taking account of Montgomery's work and the findings of the protocol study) are used, along with the description of tactics employed as stages of the WOM, to define the decision tactics which have been implemented in the WOMBAT prototype.

### 2.3.3 Computers in decision making education

The most common use of computers to support learning in decision making has been in the provision of appropriate computer-based environments to support the decision making activity.

One example, which is not so much education in decision making as a tool to support education in materials and process selection, is the software developed to support the Open University course T201, 'Materials in action'. This software incorporates weighted objectives techniques in ranking possibilities, and allows students to investigate the effect of varying the constraints or changing the weightings on the (pre-defined) objectives. The design of the interface also allows the student to consider a very large number of possibilities at the same time. However, the student has little control over the criteria on which the decision is to be based, and no control at all over the decision process used.
A computer-based tool to support education in decision making is described by Boxer (1979), who gives an account of the Management Decision Making Project at the London Graduate School of Business Studies. This project was set up to produce learning techniques capable of developing the intuitive, qualitative and judgemental aspects of decision making. The hypothesis on which this work was based was that while the rational, analytic mode of decision making is wholly object-referenced, judgement involves decision makers in reference to their own past experience - i.e. the necessary knowledge is subject-referenced. A method was developed to enable managers to explore their subject-referenced knowledge. Within the software supported by the Management Learning Project ("NIPPER") there were 6 programs referred to collectively as reflective analysis. These were concerned with helping managers to know their own views, both in relation to past experience (reflective analysis) and in relation to the views of others (consensus generation). Past reflection enables the user to consider a number of different sets of past experiences which might be relevant to the current problem. In each case the user identifies the content of the past experience and different concepts of value which he feels are significant. In the example given in the paper, the manager defines cars ('past experiences'), and analyses the characteristics ('concepts of value') of the different cars.

This approach is in the tradition of Repertory Grids (Shaw 1981) - an approach intended to enable people to find out their attitudes to some subject by encouraging them to express their views on various aspects of the subject. The same approach can be discerned in the early modules of another decision support system, called PROSPECT, which is designed to provide careers guidance to undergraduates. The early modules of PROSPECT are designed to facilitate students' understanding of their own strengths, weaknesses and aspirations. Later modules of PROSPECT use the weighted objectives method to assess candidate careers against these strengths, weaknesses and aspirations. In particular, the user is invited to identify the factors which are important to them in order of decreasing importance, and to indicate their relative importance by allocating proportionate numbers of blocks to the different factors (for example, if job status is half as important as potential earnings and potential earnings has 8 blocks, then job status would be allocated 4 blocks).
the system does not have all the necessary information then the user is invited to enter their assessment of the information (for example, if the user defines as a factor that they would like to live in Sheffield then for each career they would have to enter an assessment of how good the openings for that career are in Sheffield). Once the system has all the necessary information about the identified factors it will display colour-graduated assessments of the candidate careers against the weighted factors. Thus, although the system clearly makes use of a numerical algorithm such as the weighted objectives method, the user is not presented with simple numerical utilities, but with a broad-based visual assessment, which hides the minor differences between utilities which the user might assume have more significance than is actually the case.

In contrast to the systems already described, DecisionLab (Schiff and Kandler 1988) has a teaching component as well as a decision support environment. DecisionLab is an experimental system which is intended to demonstrate a design for user coaching in managerial decision support. An explicit representation of managerial modelling knowledge is used as a basis for providing constructive feedback on the user's decision plans via a critiquing discourse. The stated aims of the project, in terms of the criteria on which it will be evaluated, are its effectiveness in helping users to arrive at satisfactory decisions and its effectiveness in promoting user learning (which they also refer to as its 'knowledge communication skills'). DecisionLab is based around a management game. The system is designed to increase the user's competence in applying domain knowledge by representing and communicating explicitly knowledge which is generally implicit in conventional managerial models. The critiquing approach taken is based on Miller's ATTENDING system (see §3.4), in that the expertise embedded in the model is not treated as an ideal to which the user should try to conform, but as a 'second opinion' which the user is free to either accept or reject. Correspondingly, it is intended that DecisionLab should provide constructive feedback on the user's management plans. The system provides a 'guided exploration' environment based around a management game called LakeWorld. The user enters management plans through a simple 'spreadsheet calculator' interface, thus avoiding the need for natural language comprehension facilities. The user is required to state
objectives (for example, minimise unemployment, or get unemployment down to 6%) and outline the decision options under consideration (which when taken together constitute one decision plan) by assigning values to a set of decision quantities. Assumptions, or estimates, are made about the values of other parameters. Schiff and Kandler give a detailed example of one management plan, and how it might be critiqued, and briefly discuss the proposed architecture of the system.

With the exception of DecisionLab, all the systems described are tools which allow the user to articulate their view and establish how this affects the outcome of a decision. Two of the systems employ the WOM principle to perform calculations, but the calculation is a 'black box' so the user is not encouraged to learn about the decision process. DecisionLab exists in a different decision making paradigm, and adopts a critiquing approach to the user's management plans. WOMBAT has some commonalities with all the systems described. The main decision strategy incorporated is the WOM but, unlike the T201 software or PROSPECT, the decision strategy in WOMBAT is open to inspection and change. Like the NIPPER project, the artifact used for discussion purposes is cars, and the educational aims are very similar, but the approach being taken in WOMBAT is much more interactive than that in NIPPER. The teaching strategy (collaborative problem solving) adopted in WOMBAT is compared with the critiquing approach used in DecisionLab in §9.3.

2.4 Conclusion.

In this chapter, a lot of ground has been covered. Much of this review has provided fairly general background information to place the thesis in an educational setting. The most important points raised in this chapter are:

- in this research the pedagogical issues which are being addressed relate to the student considering a design as a whole ('total design'), dealing with open-ended problems which do not have a unique correct solution, and dealing with skills of judgement and critical appraisal.
- the topic to be taught is design evaluation. Both prescriptive and descriptive research
results have been presented in this chapter; in particular, the Weighted Objectives Method and various decision rules identified by Montgomery have been outlined.

In the next chapter, which is a selective review of work on Intelligent Educational Systems, consideration is given to issues including ways in which computers can be used effectively to support the teaching of topics such as design evaluation. Ideas from both of these chapters are used as a basis for the design of WOMBAT as presented in §4.3.
Chapter 3: Intelligent Educational Systems

The previous chapter describes the educational context in which this thesis is placed. This chapter provides context of a different sort, seeking to draw threads from relevant work on the use of AI techniques in education. The purpose of this chapter is not to provide a comprehensive review of work done to date on the application of artificial intelligence techniques to education but, rather, to review current trends and to place the work described in this thesis in context. In particular, work on system configurations is reviewed, including a discussion of the relationship between the system and the user and the role that each plays within an interaction. The role assigned to the system reflects the underlying educational philosophy; this aspect of IES research and the teaching strategies implemented in existing systems are also reviewed, with particular reference to systems designed to support learning in domains where there are no right or wrong answers. Some of the influences on this thesis are described here - most notably the work of Self on separating the task level and the meta level and on collaborative learning, the work of Baker on dialogue, and the work of Elsom-Cook on Guided Discovery Tutoring (GDT).

3.1 Introduction

Within the community of researchers and practitioners involved in the use of computers to support teaching there is a complete spectrum of views on the approach that should be taken. At one end are those who require something which can be used beneficially in the classroom today, whatever the limitations. At the other end are those who, recognising how poor our current understanding of this area is, and what great potential is offered, work on developing principles which may not be incorporated in fully working systems for several decades.

There are potential pedagogical benefits to be obtained from the development of effective IESs, including individualised instruction, well-selected presentation of material and a range of interaction styles (see Elsom-Cook 1988). In addition, there are a range of research interests; confronting the problems of developing such IESs helps to advance our
understanding of a range of associated issues such as cognition, dialogue, knowledge representation, teaching strategies etc.

The most comprehensive review of work done on the application of AI techniques to teaching is that by Wenger (1987). He describes past trends in IES research, highlighting the advances made in our understanding of, for example, student misconceptions or teaching strategies. He presents a view of teaching as "the communication of knowledge". Among the assumptions of most of his work are that the tutor (computer) knows something which must be communicated to the student.

Wenger presents what Self (1988) terms the 'Traditional Trinity' of components of any Intelligent Tutoring System (ITS), namely the domain knowledge (the 'object of communication'), the student model (the 'recipient of communication') and the pedagogical knowledge (the 'skill of communication'). This model serves to indicate the range of knowledge that an effective tutoring system needs but, as Self (1988) points out, such a restrictive framework fails to take into account many of the recent trends in ITS research, such as an increasing emphasis on metacognitive skills (see §3.3) or the need for multiple representations of domain knowledge (see, for example, (Cox et al 1988) (§2.1.3)).

Many research projects have concentrated attention principally on one aspect of a tutoring system, whether that be the student model as in PIXIE (Sleeman 1987) or pedagogical knowledge as in WHY (Stevens et al 1982). Others, such as Anderson's LISP tutor (Anderson and Reiser 1985, Anderson et al 1990) or SPIRIT (Barzilay and Pople 1984), which teaches about probability theory, have approached the issue of how to teach more comprehensively, incorporating all the relevant types of knowledge in one system which can then be evaluated in terms of its educational effectiveness, since it is actually usable by students. Barzilay and Pople (1984) propose an 'experimentation and tuning' approach to the development of a complete IES, arguing that system effectiveness is much more important than component effectiveness. This solution-focused approach is appropriate as long as the problem and the components which form the solution are adequately understood. However, the basic theoretical knowledge underpinning such systems is
Design, Decisions and Dialogue  Chapter 3: Intelligent Educational Systems

currently so poor that the development of complete systems makes it very difficult to assign
credit or blame to the design of one particular component of the system, and thus while
design evolution (modifying an existing design to suit a slightly different application) is
possible from a working-but-poorly-understood system, truly innovative design is not.

3.2 Configurations

Many recent systems are basically sophisticated learning environments (i.e. a
simulation which the student can manipulate in order to investigate the properties of the
system). For example, STEAMER (§2.1.3) is a simulation of the propulsion plant of a large
ship, while QUEST (White and Frederiksen 1987, 1990) presents a progression of models
in the teaching of basic electrical circuit theory. In contrast, most classic tutoring systems,
such as WHY (Stevens et al 1982), do not have a learning environment component which
the student can access directly.

There has been a trend in recent years, noted by Lawler and Yazdani (1987) and Self
(1988), towards the integration of these two research strands, namely learning environments
and intelligent tutoring.

One extant example of this general approach is the RBT tutor (Woolf et al 1987)
(mentioned in §2.1.3), which consists of a complex learning environment (a real time
simulation of a recovery boiler, as used in pulp mills) together with a discourse component.
This discourse component seeks to "subordinate teaching to learning" by allowing the user
freedom to experiment with the simulation, providing minimal guidance as long as the
student's performance is acceptable, or progressing in the right direction, but giving more
help if the student seems to be in difficulty (presumably by comparing her performance to
some notional good performance).

Two other examples are WEST (Burton and Brown 1979) and SMITHTOWN
(Shute and Bonar 1986). Both of these systems consist of a learning environment which the
user can manipulate directly, with a computer-based coach which seeks to guide the student
if it considers that her use of the environment is sub-optimal. In the case of WEST, the
environment is a simulation of the game "How the West was Won", which gives the student practice in basic arithmetic. In the case of SMITHTOWN, the environment is a simplified model of the economy of a small town. In both cases, the computer-based coach remains silent as long as it considers the student to be performing well, but interrupts with guidance which is relevant to the current situation when it assesses that the student would benefit from such guidance.

The emergence of systems such as WEST and SMITHTOWN reflects a growing realisation in the IES community that there is great pedagogical advantage in allowing students access to the variables underlying a model and allowing them direct control; that, as Kagan (1966) observes, "active involvement promotes learning". However, unguided exploration can be firstly very time-consuming, and secondly might leave the student without ever having discovered some very basic and crucial results.

The dilemma over how much - and what sort of - guidance the student should be given is hardly a new one; Keislar and Schulman (1966) quote Page as writing in 1847:

"It is always a difficult question for the teacher to settle, 'How far shall I help the pupil and how far shall the pupil be required to help himself?'... That the pupil should be taught mainly to depend on his own resources... is the teaching of common sense. ... And the teacher... may indeed, sometimes give a word of suggestion during the preparation of a lesson, and by a seasonable hint, save the scholar the needless loss of much time. But it is a very greater evil if the pupils acquire the habit of running to the teacher as soon as a slight difficulty presents itself, to request him to remove it."

In the following section, the issues of what sources can yield information on how to teach (or, more appropriately, on how learning can be facilitated), and of how such ideas can be usefully encoded in Intelligent Educational Systems, are addressed.

3.3 Teaching and learning

Much work on education theory relates to classroom practice, such as the
comparison of the efficacy of formal and progressive (or 'open') styles of teaching (Bennett 1976). Such work is not presented at a level of detail suitable for informing the design of IESs capable of conducting an educative dialogue with the student. Similarly, in discussing methods for developing thinking skills, Baron and Sternberg (1987) present worked examples and courseware, but do not attempt to define teaching strategies beyond the very general exhortation to encourage students to think and reflect on that thinking.

A limited amount of research has been done on trying to learn more about the teaching strategies adopted by good human teachers in one-to-one teaching situations. For example, Douglas (1988) studied remedial dialogues in the context of teaching English as a foreign language, while Collins and Stevens (1982) present the results of a protocol study in which they investigated the teaching goals and strategies of what they term 'inquiry teachers'. This strategy is referred to elsewhere as "Socratic teaching", and involves the teacher asking pertinent questions which lead the students to reach the correct conclusion for themselves.

Collins and Stevens have used their results in their own work on ITSs, in particular in WHY (Stevens et al 1982), which seeks to teach about rainfall processes. The teaching strategy in WHY is encoded as a series of production rules. For example, Rule 6 (quoted by Wenger (1987)) states that "IF the student gives an explanation of one or more factors that are not sufficient THEN formulate a general rule asserting that the factors given are sufficient AND ask the student if the rule is true". They give the reasons for the use of this rule as being to force the student to pay attention to other causal factors. Although they illustrate the wide applicability of their work across a spectrum of domains, all the topics they take from these domains have a rule oriented cognitive structure, and it is not clear that the approach is applicable to domains with different structures.

Discussing Socratic teaching in its original form (rather than in the rather more limited way in which it has been implemented in tutoring systems such as WHY), Lipman (1987) notes that "From the time of Socrates, dialogue has been recognised as an important way of structuring educational interactions. Socratic dialogue, in which the teacher helps the
learner bring to light what he or she apparently already knows and in which both teacher and student explore and discover together, has been a particularly interesting dialogical procedure. Nevertheless, Deweyan educational theory is no less important in this regard, because Dewey stresses the educational value of students reflecting upon - discussing, analysing and interpreting - their own experience."

This Deweyan approach, echoed by Kendler (1966) ("we learn not by doing but by thinking about what we do"), is reflected in recent work by, for example, Collins and Brown (1988). In particular, Collins and Brown discuss the benefits which can be accrued by providing a student with a trace of their problem solving strategy, both for use while problem solving (for example, by providing the student with the possibility of saying "I want to go back to that point [some state they were in earlier on in their problem solving] and try a different approach from there") and for reflection after they have solved the problem (for example, by asking the student to identify a more efficient problem solving strategy than that shown in their trace).

Commenting on this increasing emphasis in ITS research on the acquisition of metacognitive skills, namely skills of reasoning about knowledge or thought processes (for example, reasoning about how one is approaching a problem solving task), Self (1988) suggests that "ITS design might be improved by a decoupling of the task-level and the meta-level, that is, by a more careful specification of which ITS processes are addressing which level." He proposes some possible advantages of such a de-coupling:

- The ITS might not need such detailed factual knowledge, as learners might be able to interpret task-level knowledge for themselves.
- The meta-level might be designable even when the task-level is not.
- Unlike task-level knowledge, meta-level knowledge does not have to be 'correct'; it has a more advisory nature, and will be perceived as such by the user, [one assumes that at this point he is referring to the traditional view of task-level knowledge as being necessarily correct].
- It recognises that learning by doing (task-level) is limited, and needs to be supported by monitoring, guidance, advice, etc.
• The existence of a de-coupled meta-level may lead the learner to appreciate the importance of meta-level skills. The meta-level itself demonstrates the metacognitive skills we would like the learner to internalise.

• The realisation that meta-level knowledge is not correct may percolate down to the task-level, where it should also apply. At the task-level the ITS is dealing with beliefs, not knowledge.

These propositions must be regarded, for now, as postulates, as work has not been done to verify them. To date, little work has been done on separating the levels, or even on articulating what the meta-level might consist of, let alone evaluating the effect of such a separation. In this thesis, the distinction which is made is between beliefs which relate specifically to the current problem (notably what criteria are important in making the decision) and beliefs which relate to the general class of problems (notably how this type of problem might be solved).

One final issue which should be mentioned in this context is that of student motivation. Lepper and Chabay (1988) note that a much higher proportion of teachers' effort is focused on motivating their students than on imparting information, and that tutors often choose to vary their teaching style. They report observing teachers taking the role of drill sergeant or cheer leader, collaborator or competitor, lecturer or Socratic tutor or coach. The role adopted depends on more than individual teaching style; it also depends heavily on how well the student is doing and how well motivated she appears to be (and no doubt on other factors which are less relevant to this discussion). So far, motivational issues have hardly been considered in research on tutoring systems. Clearly, one aspect of motivation is concerned with the question of whether or not the student is interested in the subject; a second relates to the relationship between student and tutor; if the student is treated as a responsible partner in her own learning then she is more likely to respond positively to a tutor than if she is treated like a vessel to be filled with knowledge. Lepper and Chabay discuss the types of information that a computer tutor would need in order to be more empathetic (acknowledging also the limitations inherent in a computer's ability to detect, for example, non-verbal cues from the student). These include social knowledge (for example,
in which situations it is appropriate to praise, and how to phrase comments to improve motivation), and background knowledge about individual students (their interests, preferred learning styles etc.). Clearly, some of the issues relating to motivation are more readily realised in closed domains in which an answer is right or wrong, so that the student can be praised or encouraged according to her level of success. Motivational considerations become more difficult to tackle in more open ended domains, but the issue of motivation is no less important in such cases.

3.4 Supporting learning in domains where there are no right or wrong answers

Most of the domains covered, for example, in Wenger (1987) or Sleeman and Brown (1982) are treated as being well-defined, with unique right answers. It is now being recognised within the IES research community that treating education as the "communication of [right or wrong] knowledge" is inappropriate, and some work is now being done on teaching in domains where it is clear from the outset that there is no one correct solution. Various approaches to the issue of designing computer-based systems which are less prescriptive have emerged.

One example of a less prescriptive approach to teaching is critiquing. This approach entails the student proposing a solution to a problem, which the system then compares with its own preferred solution to the same problem. This approach admits to the possibility of there being different possible solutions, with relative merits or disadvantages. The approach assumes that the student has already formulated a plan for proceeding. Examples of this approach include LakeWorld, described in §2.3.3, and CRACK, described in §2.1.3. Another, earlier, example is ATTENDING (Miller 1984), which teaches about management plans for anaesthesia. For a patient with given symptoms, the user is invited to propose a plan for management of the anaesthesia of the patient. The system then discusses this plan, highlighting the pros and cons of the course suggested by the user and comparing it to alternatives which might be equally appropriate or preferred. This approach is similar to that adopted by one expert seeking a second opinion from another. In ATTENDING, it does not
lead to a dialogue; the user is prompted with standard menus to specify the management plan proposed then the system produces several paragraphs in response, building natural language sentences from prose fragments. Critiquing does not seek any justifications from the user for their proposed plan, nor could such a system be used effectively by a complete novice.

Following the trend identified by Self (1988) of work on a wider variety of interaction styles, DECIDER (Bloch and Farrell 1988), seeks to provoke argument and explanation from the user and expose the user's beliefs by presenting examples and counter-examples related to the current discussion, and asking apparently innocuous questions like "why?". The example presented in the paper discusses the issue of U.S. intervention in Central America, relating the user's current view of what is appropriate action to examples of previous U.S. interventions in other countries. Again, this approach does admit to the possibility of there not being a unique correct answer.

Elsom-Cook (1990a) seeks to integrate many of the ideas developed over recent years into a framework, 'Guided Discovery Tutoring', within which further work can be done. As this title implies, one premise of his work is that guided discovery, in which students are encouraged to discover facts and techniques for themselves in order to better assimilate them, but are guided in the right direction, is an effective pedagogical strategy. It should be noted here, as by Shulman and Keislar (1966), that in this context, the verb 'to discover' is taken to mean 'to make new material one's own', or to 'restructure one's understanding', rather than the more common usage of 'to find out something new' - as in 'Columbus discovered America'. The nature of the guidance in guided discovery may vary from simple monitoring and occasional coaching to fairly prescriptive intervention as appropriate. One of the fundamental aspects of this framework is that the student should be able to interact with a learning environment in the presence of an intelligent (computer-based) tutor, but that the two components should be separate. The tutor can provide appropriate guidance and assistance to the student, and is able to observe and manipulate the environment in the same way as the student, providing a symmetry within the interaction. In this context, it should be noted that symmetry refers to the rights and the options available to
the participants, and not to their beliefs or goals.

What Elsom-Cook offers is a basic framework, without discussing practical details such as how the symmetry might be administered or how interruptions could be managed. In an earlier paper (Elsom-Cook 1989), he argues in favour of a more central role for the study of the interaction in IESs, noting that "teaching is a specific case of more general strategies of interaction between humans." By analogy, symmetry might be managed by a mechanism similar to the turn-taking and negotiating facilities of people engaging in human-human interactions. This point is discussed at more length in §4.3, and again in §9.5. At this point, it should be noted that, certainly in formal learning situations and possibly also in less formal ones, the teachers and the learners have different roles within the interaction, and that this role division influences what each participant in the learning situation is likely to say and do, and also how it is likely to be understood. Thus while there may be a symmetry in the interaction, for example in terms of capabilities to observe and manipulate the environment, there is also an inherent asymmetry in terms of roles and pre-existing beliefs. Therefore GDT cannot be equated with collaborative learning, although in domains where there is not a unique answer GDT can accommodate collaborative problem solving.

Some initial work on the issues of negotiation and dialogue in the context of tutoring systems has been done by Baker (1989). Baker's work is based around a computational model of music cognition, and seeks to develop a student's understanding of musical structures by engaging the student in a critical argument (in which either student or computer can propose and justify a position for a phrase boundary, with which the other might or might not agree; in cases of disagreement, a 'critical argument' ensues). His model of tutorial interaction, which he terms 'Negotiated Tutoring' is instantiated in an IES called KANT. Although his work focused on negotiation, his system architecture incorporates a separate 'Critical Argument Controller' which is responsible for managing turn-taking.

Negotiated Tutoring may be viewed as the integration of a domain traversal algorithm (based on spreading activation, with negotiated acceptance of the goal of discussing the chosen concept) with an interaction type. The interaction type is selected from
the set of possible dialogue goals which could be pursued in the current situation, and acceptance of this dialogue goal is negotiated. The selection is based currently on three educational principles, namely preferring eliciting to imparting, preferring to challenge a previous (challengable) claim rather than make a new one and preferring the student rather than the system as speaker. These educational principles are embedded in procedures. KANT contains data structures and processes which are capable of generating some aspects of negotiated tutoring dialogues, though it must be said that the example dialogues presented in Baker (1989) are limited (consisting mainly of negotiations about what concept to discuss and what dialogue goals to pursue). The only goals incorporated in KANT are a hierarchy of dialogue goals (such as making an abstract claim, supporting an abstract claim or making a complementary claim); tutoring goals are implicitly encoded in the mechanism for selecting between competing dialogue goals. Baker's work is compared with this thesis in §B.6.1.

3.4.1 Collaborative learning and collaborative problem solving

A further example of a non-prescriptive approach is to be found in work on collaborative learning, in which the user and a computer companion seek to investigate and learn together. Examples of this approach are to be found in the work of Chan and Baskin (1988), who are developing a system with both a computer tutor and a computer companion, to support learning about integration, and in the work of Self (1987), who discusses the possibilities of using machine learning techniques in developing a student model which could explore with the user in a concept learning situation. This approach, of applying machine learning techniques to encourage collaborative learning between student and computer, must be distinguished from the teaching strategy which a computer tutor may choose to adopt of appearing to know no more than the student, giving an appearance of collaborative learning while in fact being able to intervene more directly if the collaborative approach seems unsatisfactory. In particular, Self (1989) suggests that a system which is collaborating (i.e. working and learning with the user) would not ask a question to which it already knew the answer. Thus, in a collaboration both participants can influence the outcome of the discussion or the solution to the problem being worked on.
Roschelle and Behrend (in press), discussing the nature of collaboration between humans, note that successful collaboration involves a "large degree of mutual engagement, joint decision making and discussion". They define collaboration as "a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem". This serves to make a distinction between collaborative and cooperative work; in their terms, cooperation involves the division of labour between participants, with each participant responsible for a part of the problem solving, not necessarily maintaining mutual engagement. In this thesis, the term 'collaborative' is used in the sense discussed by Roschelle and Behrend, with the obvious caveat that human-computer collaboration is less rich than that between two humans.

It must be noted that there is a distinction between collaborative learning (in which both parties are learning, or maybe struggling, together) and collaborative problem solving in which, while the system might know an answer or an approach, it is open to suggestions from the user, and does not consider its own way of doing things to be the only correct way. In this context, WOMBAT engages the user in collaborative problem solving. A further issue raised by Self (1990), reflecting on one of his own research projects, is that "We had little idea of how to support collaborative dialogues, in which both participants are to be seen as of equal status, or of whether students would, in fact, welcome such a style of interaction with a computer." This thesis addresses these issues.

3.5 How teaching expertise is encoded in tutoring systems

Almost without exception, teaching expertise is encoded in tutoring systems as a set of rules, either explicit or implicit. As outlined above, Baker's KANT system includes implicit rules embedded in his mechanism for selecting dialogue goals to pursue. WHY includes explicit rules such as Rule 6 cited above (§3.3). GUIDON (Clancey 1987), a tutoring system designed to teach about diagnosis of infectious diseases which was built to make use of the expertise encoded in the expert system MYCIN, bases its teaching on t-rules which Clancey describes as being derived largely from his own teaching expertise. The t-rules are highly interdependent, and not easy to understand in isolation. For example, t-rule
12.08 states "IF there are related factors that form a block of data THEN present the related factors that form a block of data [Proc034]". This rule appears as Step 5 of a procedure (Proc012) which might well be activated by another t-rule in a different procedure.

PROTO-TEG (Dillenbourg 1989) also bases its teaching on production rules (of the form IF... THEN...). The focus of Dillenbourg's research is on the development of a self-improving tutor which aims to discover the conditions under which it was appropriate to fire each production rule, based on information in the student model, to generate efficient teaching strategies. Each of the nine strategies is defined by the conditions (initially empty) under which it should fire, the procedures called in implementing the strategy, and a guiding rate which indicates the degree of guidance offered by the strategy. When the system is first used, the strategy having a guidance rate which corresponds best to the student's guidance level (as indicated by information in the student model) is employed. The self-improving function, which takes place off-line, aims to ascertain under what conditions it is efficient to fire each strategy.

Finally DOMINIE (Elsom-Cook and Spensley 1990) encodes expertise on several teaching and assessment strategies, each of which employs a defined domain traversal mechanism (e.g. top-down or bottom-up) and a given interaction style (e.g. Socratic diagnosis or cognitive apprenticeship).

In all of these cases, if the teaching strategy is encoded explicitly at all then it is encoded as rules, generally of the form "If condition then action". Any notion of what the tutor is trying to achieve is external to the implementation (as in the 'reason for use' quoted for each rule in WHY). One role of IES research is to formalise teaching strategies: if we can generate a plausible teaching dialogue, and understand how it has been achieved, then we have a tool which can aid our understanding of teaching processes and principles. As discussed in §9.3, the IES design developed within this thesis accommodates an explicit representation of teaching aims. This provides for the future possibility of exploring further the relationship between teaching aims and teaching strategies.
3.6 Conclusion

Various research trends have been identified in this section. These include the recognition that education is more than "the communication of knowledge" - that students have to be enabled to discover facts and techniques for themselves, but that learning is generally enhanced if that discovery is guided appropriately. Probably the most important points raised in this chapter are:

- in this research the pedagogical issues which are being addressed relate to the student dealing with open-ended problems which do not have a unique correct solution, and dealing with skills of judgement and critical appraisal. Therefore alternative approaches to tutoring in open domains (i.e. domains in which information is not treated as necessarily correct or incorrect) have been reviewed. In §9.3 the approach taken in this thesis is compared with the approaches reviewed in this chapter.

- work on integrating learning environments and tutoring systems, and in particular Elsom-Cook's work on Guided Discovery Tutoring, has been described. In the next chapter (§4.3) the design of WOMBAT, which has been influenced by the work of Elsom-Cook, is presented, and in §9.3 the results of the evaluation of WOMBAT are related back to the GDT framework, and also to the fundamental question of how much the student should be guided or given the freedom to explore at will.

- Self's discussion about separating the task-level and the meta-level, and also his work on developing a collaborative style of tutoring have been presented. As is made clear in later chapters (§8, §9), there is more than one meta-level - or more than one task-level (or both!).

- dialogue is recognised as being central to the interaction. The work of Baker, defining many of the desiderata for designing a dialogue component for a tutoring system has been outlined. §B.6.1 includes a comparison of the work developed in this thesis with that of Baker.

These points and others are taken up again as they become relevant in future chapters.
Chapter 4: The design of WOMBAT

4.1 Introduction

The context of the research reported here is engineering design education. The skills which this research seeks to address are concerned with considering a design as a whole (rather than just as a collection of well designed components - though this is also important), reasoning skills associated with being able to develop a 'model' of the problem and deducing requirements from that, and skills relating to ability to make this type of decision (procedural skills).

The topic within the design curriculum being used as a vehicle for this work is design evaluation (in the context of selecting the preferred design concept for detail design). For the purposes of the current research project, the topic has been simplified in many ways:

- feedback (modifying the problem specification or alternative possible solutions) is not catered for,
- no account is taken of probability (dealing with uncertainty in the data),
- the use of decision trees, as described by Pahl and Beitz (1984), is not considered, and
- no account is taken of dependencies in the data (the domain is treated as independent justified beliefs).

These simplifications have been made in order to reduce the size of the problem to one which can be addressed reasonably within the available time scale. The consequence of making these simplifications is that the rich problem of design evaluation is reduced to the simpler problem of selection between defined alternatives. This simpler problem retains the essential characteristics of relating to the design as a whole, of requiring reasoning skills, and of being open, in the sense that there is not a unique correct answer.

The example used for all practical purposes is that of choosing between alternative fictitious cars. The reasons for this are, firstly, that the approach of using real design problems with real design students would have posed highly restrictive time constraints on the work and caused great logistical problems. Secondly, in presenting work to a non-
specialist audience, there would be greater overhead in explaining the problem if it were not familiar, which would have distracted from the focus of the work. Cars were chosen in preference to a domestic appliance because their design involves a greater engineering input, and a decision over a car is 'bigger' (most people will agonise for much longer over choosing a car than over choosing a toaster). With the benefit of hindsight (see §8.5.1 and the Epilogue) it is possible to see that there are definite disadvantages to the use of cars in this context, but ultimately the precise nature of the artifact used is a relatively minor consideration; as is made clear in §7.3.4, all the system's beliefs about cars could be readily exchanged for corresponding beliefs about some other artifact without affecting the fundamental nature of this work at all. It should be noted that the data which is made available on cars relates mainly to functionality rather than to appearance or image, though it is recognised that these aspects of car design are extremely important in the marketplace.

In order to assess how practice matches up to the theories outlined in §2.3, within the decision making paradigm on which this research is based, and also to inform further the consideration of how decision making skills might be improved, a protocol study was undertaken, as described below.

4.2 A protocol study

In order to inform the design of a computer-based system to teach about selection between alternatives, a small scale protocol study was conducted. The purpose of the study was firstly to establish to what extent the results discussed in §2.3.2 could be used as a basis for the problem solving representation in the system (i.e. do the decision tactics and strategies described by Tversky and Montgomery provide an adequate basis for such a representation, and if not, what is an appropriate representation?). Secondly, it was intended to ascertain what notes subjects took during their decision making, and how they chose to structure and manipulate the information they used in their decision making; if a pattern emerged, then this would provide evidence to inform the design of a computer-based tool to support this activity. Finally, the study was intended to identify what sources of information and lines of reasoning subjects made use of in their identification of criteria on which to base
4.2.1 Description of the study

Pairs of subjects were asked to discuss a car-selection problem, and verbal protocols were taken for subsequent analysis. In professional terms, all subjects were novices, in that none had received formal education in decision making (see §A.1), though it might be argued that all had received training through life! The intention, in working with pairs of subjects, was primarily to create a situation in which it was natural for subjects to articulate their thinking throughout the decision making process (the approach is similar to that adopted by Suchman (1987)). It is recognised that the results from two people working together will be different from those of one person working alone, or of one person working with a computer, but as a pilot study the results of this experiment were being used to inform further work, and not to support or justify any particular theory. The problem was presented as follows:

Select the most appropriate new car for a family of four. They can spend no more than £8000 (on the road price). They do not have an old car to trade in, and are not interested in any finance schemes. As this is their only car, they require the largest possible luggage capacity; the minimum acceptable is 25 cu. ft.. The family is concerned for the environment, and therefore intends to use lead-free petrol, and they consider economy of greater importance than performance (within reason!). You may consider any additional criteria if you wish, and exercise your own judgement in making a recommendation. Please select one and justify your selection. Sorry - no test drives!!

The subjects were given six alternative solutions, two of which failed on absolute criteria (one was too expensive and had too small a luggage capacity, and one ran only on leaded fuel), and were given a small amount of information about each (see §A.2). The data was fictitious; some of it (most notably luggage capacity and acceleration) was unrealistic, though self-consistent; but this is not believed to have affected the results of the experiment in terms of decision making. The subjects were asked to first make an intuitive selection based on the available information (stage 1), then to make a better-informed decision when
they had decided what additional information they needed (stage 2), and finally they were
asked to identify and weight the objectives on which they had made their decision (stage 3);
these values were entered in a simple WOM spreadsheet program which returned an overall
value for each possible solution. Additional information was available from the experimenter
on request, in the form of 'data strips', each of which gave values for one parameter for all
six alternatives - see §A.3.

The problem was phrased to be analogous to a minimally defined engineering design
problem. For example, in engineering design there are British Standards to be complied
with; these are generally stated in absolute form, so while in a natural car selection case a
price limit of £8000 would generally be treated as advisory rather than mandatory, in the
'engineering design analogy' it was to be treated as an absolute requirement, corresponding
to the common requirement of satisfying a British Standard. Similarly, the requirement that
the decision maker take into account lifestyle knowledge (this car is for a family) is treated as
corresponding to the designer developing a model of the potential customer of the product
(that is, a 'picture in the head' of the likely requirements of the market sector being targeted,
and the way that people in this group would be expected to use the product). Like all
analogies, this one is imperfect, but it is believed to be sufficiently close for the purposes of
this research.

4.2.2 Outline results of Stage 2 for each subject pair

In considering how people approached stage 2 of the problem, a range of strategies
emerged, as follows.

Group 1 was very methodical. They immediately set about drawing a matrix of
criteria against alternatives, and underlining the requirements stated in the specification. The
factors they took into account were those explicitly stated in the specification, and additional
factors which they perceived as relating to economy (namely bodywork, maintenance costs
and running costs). They also instinctively set about scaling the solutions with regards to
each criterion they identified - first on a scale of 1 to 4, and later on a scale of 1 to 6 (when
they decided 4 was too coarse). Despite noting at the beginning that D cost too much, it took
them about 20 minutes to get around to eliminating it. Soon after that, they also eliminated C. They proceeded to eliminate further possible solutions on pairwise comparison: AvB eliminated A; FvB eliminated F. The choice between B and E was much harder; forgetting their scaled values, they allocated + and - points for 'better' or 'worse', and eventually came out with B as their preferred solution. Although this pair worked very systematically in the earlier stages, their approach got more confused towards the end.

**Group 2** also tackled the problem quite methodically. They started by scribbling notes on the brochures, but then created a matrix containing the criteria from the specification and all the criteria mentioned in the car 'brochures'. Data inserted in the matrix was raw (i.e. they didn't scale it in any way). Some decisions were made by elimination by aspects - first, all saloons were rejected, then A was rejected on the grounds of being 'sporty'. They then looked at the factors which distinguished between E and F, and finally made a decision based on three factors which distinguished between these two.

**Group 3** worked iteratively through the solutions, working only on the information contained in the specification. The only notes they made were on the brochures, and even these notes were quite limited. They seemed to decide quite early on that F was the best solution, and proceeded to work by proving that all others were significantly worse. Towards the end of the experiment one of the subjects in this pair decided that the phrasing 'largest possible luggage capacity' meant that all they should have done was check which ones satisfied the absolute criteria, and then choose the one with the largest luggage capacity, ignoring all other factors:

"Well I would say 'they require the largest possible luggage capacity' is fairly unequivocal, wouldn't you?"

"Yes"

"And we've satisfied that criteria, haven't we?"

"Yes"

"Because it is!"

**Group 4** worked in a more holistic way, trying to take into account all possible
factors in choosing a car, rather than being bound so closely by the specification. They noted down as many criteria as they could think of before asking for any data, and added to this list as more criteria came to mind. They eliminated C and D (reluctantly) as not satisfying the specification, and E as having too low a safety factor and reliability. Eventually, they identified 11 criteria on which to make a selection between the remaining three alternatives, and it was at this stage that they set up a matrix of objectives against alternatives. They then proceeded by doing informal comparisons between A, B and F.

**Group 5** worked iteratively through the alternative proposed solutions, taking into account different information on each pass. Their approach was clearly driven by the alternatives. They did a lot of surmising and hypothesising when they could simply have asked for data. Having eliminated C, D and E, they tended to do pairwise comparisons on one or two factors between the remaining alternatives, then ask for a bit more data and repeat the process. At a fairly late stage, they discarded some of the data strips they had acquired as no longer being relevant to the decision process, and they used the remaining strips as a matrix, instead of writing anything down. Although they said right at the beginning that four doors was important to them, they never in fact asked for this information.

### 4.2.3 Outline results

The format of the problem clearly influenced the mode of cognition of the subjects (Hammond et al. 1987); because all information was presented in numeric form, people were more analytical than they would be in a real car selection situation, in which more qualitative information (e.g. what does it look like? how does it feel to drive?) would be available. The information was also very accessible; the subjects simply had to ask, without setting up complex experiments, making physical models or running computer simulations, as would happen in a true concept-selection situation.

It is clear from the above descriptions that tactics used included simplifying the problem, generally by eliminating weaker candidate solutions from further consideration ('elimination by aspects'), or focusing attention on only two alternatives ('pairwise comparison'), and occasionally by considering only one or two aspects of the problem at a
Another way of dealing with the quantity of information was to create a matrix to hold the data which was considered relevant (sometimes with vacant slots to mark data not yet obtained or which the subjects were not sure how to deal with).

Another feature of the decision making strategies adopted is that they were in general alternatives focused, in the sense that the subjects focused much more on the proposed alternative solutions than they did on the problem specification. This is in contrast to the prescriptive WOM, which is entirely problem focused. In a different context, Cross and Nathenson (1981) quote Lawson as saying that "the essence of [the designer's] approach is that it is simultaneously educational and solution seeking". They paraphrase this as "the designer learns about the problem by posing tentative solutions to it". Similarly, in this study, groups seemed to learn about the problem by considering possible solutions to it. This also highlighted an inability, or reluctance, to think about the problem in isolation. This factor would probably still take effect in an engineering design situation, though to a lesser extent as the students would already be familiar with the problem if they were making a decision within the context of a design project.

It is not particularly clear how people made trade-offs between factors where a direct comparison cannot be made, except through devices such as cancellation (for example, if one was better than another on price, but worse on economy, then those two factors might cancel each other). One pair set about scaling values initially, but failed to exploit this standardisation later in their trade-offs, or while assigning priorities to objectives.

4.2.4 Relationship between experimental results and descriptive research

As has already been noted (§4.2.3), according to Hammond's Cognitive Continuum theory, the form of the task content (with largely numerical cues and an unspecified time limit) encouraged subjects to be analytical; conversely, the facts that subjects were unfamiliar with the task content and that there were a large number of cues are factors which encourage intuition. Thus, the Cognitive Continuum theory would predict that some intermediate form of cognition - neither wholly intuitive nor wholly analytical - would be employed. This seems to have been the case, but this observation is not of particular significance in this
study. The applicability of the aspects of the Cognitive Continuum theory which account for the development of expertise have not been tested, as this study only looked at novices. Of more importance to this research are the observations that the Weighted Objectives Method is not, as one might at first think, what Hammond calls a 'simple weighted averages organising principle', but is in fact a fairly complicated organising principle. Of most significance is the next conclusion from Hammond's work: that the development of expertise involves learning this complicated organising principle.

Comparing the strategies of subjects with those described by Tversky and Montgomery (§2.3.2), it can be seen that the strategies used by all subjects were much more complex than a simple elimination by aspects as outlined by Tversky. It would be possible to massage them into a form of dominance structuring as proposed by Montgomery, but to do so would be somewhat false and unhelpful. However, some of the tactics identified and described by Montgomery can be clearly discerned within the strategies employed. The principal tactics used were a non-compensatory rule similar to elimination by aspects (in the early stages) and a compensatory rule similar to the Maximising Number of Attributes with the Greatest Attractiveness rule (in the later stages).

The initial stage was generally to learn about the problem through considering the possible solutions and putting forward tentative ideas, and identifying factors which should be considered and which discriminated between the alternatives. This stage generally included eliminating D and C as not satisfying the specification (a form of Elimination by Aspects). This stage was similar to Montgomery's pre-editing phase.

The problem was then simplified as far as possible, for example by eliminating E on the grounds of low safety (though people were not given enough information to quantify what 'low safety' might mean!). For example:

"That one, if they're safety conscious, hasn't got enough strength. That one's out... E..."

"That one's out on tru... yes"

"E hasn't got enough strength, if they're safety conscious"

This tactic is similar to Elimination by Aspects in that it eliminates on one aspect, but
it involves relative rather than absolute values. One pair toyed with the idea of only considering the two cheapest alternatives. This corresponds to a tactic like a cross between the disjunctive and the lexicographic rule (§2.3.2); this idea was not pursued to its obvious conclusion:

"So we're back to what we were doing before, which was comparing the two lowest price ones"

"mmhhmm"

"To see which was the better. I mean, if that one turns out to be better than that one,"

"mmhhmm"

"Then does it follow that we can just discount the next two up?"

"No!"

In the later stages of the decision process, compensatory rules were used by all except for group 3, who essentially made their decision on the lexicographical rule (maximising luggage capacity). Several pairs made explicit reference to the tactic which is at the heart of the Weighted Objectives Method, i.e. the Addition of Utilities rule. For example:

"... if we want to do a proper comparison we have to have a grid with all the factors on it and values for each one and for each car. Is this a sensible way to proceed?"

"Uhhuh. Have a rating scale, like, yes, a weighting sort of idea"

However, no group actually pursued this and defined weights or priorities during stage 2 of the experiment. Thus, while the tactics used at the end of the decision process were generally poorly articulated, and hard to define, they seem to correspond most closely to the Maximising Number of Attributes with the Greatest Attractiveness rule, e.g.:

"It actually wins on three, doesn't it? - Because it's up on the luggage, it's up on the economy, it saves them 800 pounds, although they've not said that that's vital, but, you know, if they can spend absolutely no more than 8000..."

This stage seems to correspond most closely to Montgomery's 'selection of a promising alternative' stage, but the comparison does not seem particularly helpful or informative.
Steps of dominance testing and structuring cannot be readily identified in the protocols; once a 'promising solution' had been uniquely identified, and subjects were happy that they had considered all important factors in their decision making, no further processing was done. This can probably be explained by the fact that subjects had no real commitment to the results of their decision making (Montgomery 1989) as they were not going to spend their own money on the chosen car. It should, however, be noted that subjects did employ some of the tactics which Montgomery identifies as part of dominance structuring in their simplification of the problem, notably cancelling, as mentioned previously, and de-emphasising:

"Yes, OK. We are just happy with the minimum luggage..."

"Yes, because the maintenance and running costs are..."

"override it - are more important"

These results can be compared with those of Payne (1976), who did a protocol study of individuals selecting between alternative flats. In Payne's study, the subjects were given varying numbers of alternatives, and varying amounts of information about each alternative, and the main focus of his work was to study the effect of varying these parameters on the decision making process. While the details of his study are not important in the current context, it is worth noting that among his conclusions are that when faced with only two alternatives, subjects employed strategies consistent with a compensatory decision process, whereas when faced with more alternatives (6 or 12 in the experimental situation) subjects employed decision strategies designed to eliminate some alternatives as quickly as possible, on the basis of a limited quantity of information. This finding is consistent with the results of the current study.

4.2.5 Results of Stage 3 of experiment: relationship to WOM

When subjects were asked to identify and rank the objectives on which they had based their decision, they were able to do so with varying degrees of ease. Subjects were asked to forget about the results of the decision they had just made and to work from the problem specification, but they proved unwilling or unable to do this. One pair would only
identify the parameters which distinguished between their last two possibilities (E and F), while another had clear difficulty over the ranking of reliability (text in square brackets is spoken by experimenter):

"This is in respect to having made the decision on F?"

["Forget that you've made that decision on F for the moment"]

"Yes, OK. Well in that case, I would put it quite near the top then, but the other factors that have now outweighed it..."

"mmm"

"Did we put safety first?"

["Yes"]

"And then?"

["Number of doors"]

"I think that I would put safety and reliability, but that the reliability hasn't sort of... obviously overridden our decision"

This experience can be compared with Hammond's experiment with highway engineers (Hammond et al 1987); in that experiment, Hammond reports that the tasks were ordered so as to minimise the effect that the results of one task had on the conduct of subsequent ones (by presenting the task in the form expected to induce intuition before presenting it in the form expected to induce more analytical modes of thought). In the case of the present experiment it is not clear that such independence is achievable at all. While no-one had any great stakes in the outcome of stage 1 of the experiment, subjects had a strong instinct to prove their stage 2 decision correct, and thus stage 3 was treated as a reinforcement of stage 2. It is believed that reversing the order of these stages would have rendered the whole experiment useless, in that results from stage 3 would have greatly influenced a subsequent stage 2, and subjects would not have had sufficient opportunity to learn about the problem if immediately faced with stage 3.

As a consequence, no particular conclusions can be drawn from the results of stage 3 of the experiment, and no clear patterns of behaviour can be identified. The reader is referred to the results in §A.4, which summarise the objectives and weightings defined in stage 3; the
main point to be drawn from these results is the poor correlation between the objectives as defined by some subjects and the initial problem specification. By virtue of a combination of stage 3 being a re-statement of stage 2 and sheer chance (given that group 1 had misconceptions about the data (such that a higher figure for time to accelerate from 0 to 60 mph was considered better than a lower one), the way that group 2 identified only parameters which distinguished between E and F, the use of the lexicographical rule by group 3 etc.), the outcome of every stage 3 analysis agreed with the outcome of the corresponding stage 2 analysis.

Finally in this section, a brief note is included about the relationship between the experimental results and the prescriptive Weighted Objectives Method. The way that the experiment was conducted eliminated the possibility of certain stages of the WOM being touched on at all. Comparing stage 2 of the experiment with the prescriptive WOM as defined in §2.2.3, it is apparent that:

1) All pairs, at some stage, checked that all alternatives satisfied the minimum requirements to be solutions to the problem, and eliminated all possibilities which did not.
2) The stage of identifying evaluation criteria happened in an ad hoc, somewhat unstructured way, but it did happen.
3) No-one did anything explicit about weighting except when asked to in stage 3.
4,5&6) The steps of selecting parameters and assigning values to given ranges of them were done as part of the experimental set-up, for example if they asked for fuel economy then they were given it in mpg, and the assigning of values to ranges was done within the WOM program used in stage 3. Similarly, no-one did their own summing of a product; this was done by the WOM program in stage 3.

4.2.6 Summary results: management of information

Following on from the protocol study issues to be addressed in the design of computer support for this activity are highlighted in the next three sections (§4.2.6 - §4.2.8). An outline design for a system which addresses these issues is then presented (§4.3).
### Notes:

- 'Raw' luggage data was only entered for saloons, as hatchbacks had variable values. Scaled values (on a scale 1-6) for all alternatives were entered.

- The top row of economy data is scaled values (on a scale 1-6) for fuel economy (mpg) on urban cycle. The bottom row is engine capacity (cc).

- The top row of performance figures is time to accelerate from 0mph to 60mph, and their corresponding scaled values (highlighting their misconception as to what constituted good acceleration). The bottom row is top speed.

- The bodywork figures are ratings as supplied by the experimenter (scale 1-10).

- Running cost figures are value ratings on scale 1-6.

All items on a scale of 1-6 were rated by the subjects; the descriptive terms they used were:

- 1 - very poor
- 2 - poor
- 3 - OK
- 4 - good
- 5 - very good
- 6 - excellent
The first, and most obvious problem, commented on by most groups prior to their setting up a matrix of objectives against alternatives, was that of assimilating and manipulating such a large quantity of information:

".. because, I don't know, with all those figures in front .. I find it rather... can't make a decision because we've got too many figures, sort of thing..."

Four of the five pairs of subjects in this study dealt with this problem by constructing such a matrix on paper. An example is shown in Fig. 4.1.

However, their difficulties did not end there. Having identified objectives, subjects often had great difficulty in ranking priorities. For example, as they were ranking things, one pair got to a point where comfort seemed to get a rating of 4:

"Sounds quite a long way down the list, doesn't it?"

"It does, yes"

"They're sitting on these two wooden boxes...[laugh]"

"Sitting, yes, in a 2CV, or whatever"

"A mini. This is really a mini we've got, with a big engine and a huge boot"

"Yes"

"A vamped up mini"

"Yes, I'm not sure I'd put comfort so low, actually. It's difficult, isn't it? I'd put comfort above running costs"

One pair had to retrace (albeit rapidly) their decision making when they realised that while they had stated that reliability was an important factor, they had actually failed to take it into account:

"So actually F's not so reliable... I suppose the... although F's actually the least reliable..."

"It's still got the..."

"...the highest boot capacity and the safety"

"...and the safety"

"And was it the best fuel consumption?"

"mmm"
"yes, it was, wasn't it"

"mmm... right I still think I'd probably go for F...even with the reliability, because it's only one point less than B."

"Yes, and then A has got two doors, hasn't it?"

"Yes"

"And B's not as safe.. the factor's not as high for safety, is it?"

"No, that's true"

"And the boot capacity's a lot lower than F, isn't it"

"mmm. And I don't think I'd buy a car with just 1100ccs for a family of four, either."

"So we'll stick with F anyway"

Another pair said early on that having 4 doors was important, but never actually took this into account in their decision making. Thus subjects had some degree of difficulty in keeping track of all the ideas they had, occasionally forgetting things before they had noted them down or used them.

While no one seemed to regret the passing of C, there was in general a reluctance to delete D, largely because of its good safety rating, even though it failed on two absolute criteria (illustrating compensatory thinking!). For example:

"...D is just under"

"mmm. Minimum acceptable is 25 cubic feet. Shall we assume that that is an absolute rule?"

"Seems reasonable"

"Because that would make our life easier"

"This is slightly over the price as well too, is D"

"mmm. Let's write it in anyway"

(same group, some time later)

"...it's absolutely crossed of, because they just can't afford it."

"No, and it's got a small boot"

"Well not unless they're going to sell their granny or pawn the dog, or something"

"They might not like their granny..."
To summarise, most subjects set up a matrix of objectives against alternatives to help them manage the quantity of information. Only one pair (Group 1) scaled this data in any way. Any computer support for this activity should make it easy for subjects to note down ideas, rank objectives, manipulate (for example, scale) data and make other notes (such as which alternatives have been eliminated, and why).

4.2.7 Summary results: decision criteria

The sources of inspiration used for requesting information and defining their objectives were:

1) Problem specification + ideas derived by association from 'economy' and 'luggage space'
2) Problem specification + additional information provided about cars
3) Problem specification
4) Problem specification + general knowledge about problem
5) General knowledge about problem

Thus, cues were largely taken from the printed material available. The subjects whose situation most nearly matched that in the specification developed the fullest models of the problem, in terms of envisaging additional criteria which should be considered and evaluating their importance.

4.2.8 Summary results: decision processes

Although several pairs of subjects made explicit references to tactics such as the addition of utilities rule (§2.3.2) these references did not result in any discernible commitment (c.f. Winograd and Flores (1986), who suggest that any utterance has an illocutionary point, which is the 'specification of meaning in terms of patterns of commitment entered into by speaker and hearer by virtue of taking part in the conversation' (p.59)) - i.e. although people talked about these things, they did not follow them up with the corresponding actions. Indeed, people's problem solving activity appeared to be largely opportunistic, based on the instinct of simplifying the problem as far as possible (by either eliminating alternatives from further consideration or by considering only two alternatives at
a time) before applying compensatory decision strategies. Based on the decision tactics identified by Montgomery and those which could be discerned in the protocols, a list of decision tactics has been compiled as follows.

**Organising data**

- make a list of the agreed objectives
- make a list of the possible alternatives
- set up a matrix of objectives against alternatives

**Identifying objectives**

- identify objectives from problem spec
- identify objectives based on lifestyle knowledge
- identify objectives by considering alternatives

**Manipulating objectives**

- identify parameter to measure objective (prerequisite: objective defined)
- obtain values of the parameter used to measure an objective for all alternatives (prerequisite: objective and parameter defined)
- break objective down into sub-components (prerequisite: objective defined)
- define worst acceptable value of objective (prerequisite: objective and parameter defined)
- rank objectives (prerequisite: there is more than 1 objective)
- identify most important objective (prerequisite: there is more than 1 objective)
- weight objectives (prerequisite: there is more than 1 objective)
- define scaling system to normalise an objective (prerequisite: objective and parameter defined)
- normalise objective to an agreed scaling system (prerequisite: objective and scaling system defined)

**Manipulating alternatives**

- eliminate all alternatives which do not satisfy the problem specification
- eliminate alternatives worse than worst acceptable on an objective (prerequisite: appropriate objective defined, worst acceptable value defined)
- eliminate worst alternative on an objective (prerequisite: appropriate objective defined,
parameter defined)
• eliminate alternatives worse than worst acceptable for all objectives (prerequisite: worst acceptable values of all objectives defined)
• eliminate alternatives worse than worst acceptable for any objective (prerequisite: worst acceptable values of all objectives defined)
• pairwise comparison: compare two alternatives and eliminate the one which scores worse on the larger number of objectives (prerequisite: there are at least two candidate alternatives left)

Make selection
• dominance: select best on all objectives, if such an alternative exists (prerequisite: objectives and parameters defined)
• select best on most important objective (prerequisite: most important objective and parameter defined)
• calculate sum of product of weights and scaled (normalised) values and select alternative with highest score (prerequisite: all objectives weighted, all parameters scaled)

Different combinations of these tactics yield different decision strategies.

Tactics used in the course of decision making are identified in this way so that they can be encoded within the tutoring system. The tactics listed are the ones which the implemented dialogue agent is able to discuss with the user, and it is from these tactics that problem solving strategies are constructed. In practice, in the WOMBAT prototype the only tactics which are fully implemented are those which comprise the WOM strategy because this is the preferred strategy in most engineering texts, and it does not involve iteration (as, for example, the controlled convergence strategy of Pugh does (§2.2.3)). The purpose of identifying tactics used in the course of decision making in this way is discussed in detail later (§6.8, §7.6.2).

4.3 The outline design of WOMBAT

The results of this experiment have been used to propose a design for an IES to
support and improve people's decision making. It was concluded that basic support could be provided simply by making available an appropriate environment; the most obvious was a simple spreadsheet fundamentally like the matrices that most subjects created on paper. This is discussed more fully later. In seeking to improve decision making skills, it is believed that, in addition, a dialogue needs to be conducted with the user, to encourage them to externalise and reflect both on the decision they are currently making and on their decision making processes. Some of the advantages of including a separate dialogue component with a learning environment (relating to improving both the efficiency and the effectiveness of learning) have been outlined in §3.2.

4.3.1 Improving the quality of the current decision

In considering how the quality (and justifiability) of the decision taken might be improved, reference to the results in §A.4 shows that some pairs developed a fuller model of the problem than others though, in one case at least, this led to them ultimately forgetting about the problem specification and reaching a decision based on a different problem! Within the protocols, statements which were considered to demonstrate that the speaker was developing a model of the problem were those which included a direct reference to the probable lifestyle and requirements of the fictitious family, in terms of the way they would be likely to use their car. In seeking to encourage students to externalise and reflect on their thinking, it is believed to be appropriate to seek justification of objectives and weightings, to refer them back to the problem specification when they appeared to be reaching a decision on a different problem, and also to make suggestions and encourage development of a model of the problem (e.g. in this case, lifestyle requirements).

As part of the process of learning about the problem, students need to be able to develop a deeper understanding of the problem and of the alternative solutions to it by trying things out - by seeing what the decision outcome is on one set of criteria, then changing the criteria, or the relative importance of the criteria, and seeing how the change affects the outcome. The environment should be built so as to facilitate this type of learning. (This strategy - of getting a solution then changing the decision criteria in some way - is referred to
later in this thesis as 'rinse and repeat'.

One issue which has not been addressed in the current research project is the correction of misconceptions. Within the study there was only one clear case of a decision being based on a definite misconception. (Group 1 thought that a higher figure for time taken to accelerate from 0mph to 60mph was better than a lower figure.) For the purposes of this research it has been assumed that the student has the requisite understanding of the artifact being selected, and that she has a basic understanding of the decision tactics which can be employed. The extent to which this assumption is reasonable is discussed in §8.5.

4.3.2 Improving general decision making skills

Following the work of Self (1988), who proposes separating the task-level and the meta-level, in this work a separation is made between task-level issues (such as what decision criteria are important) and meta-level issues (what tactic should be adopted next). Similarly, the influence of the work of Elsom-Cook can be discerned in the decision to separate the dialogue component (which is to act as a collaborative partner with the student in solving the problem) from the data display and manipulation component (referred to in future as the learning environment). The design requirements that there should be a symmetry within the interaction (that the system and user can say the same things) and that the participants should be able to negotiate over tactics and objectives necessitates a collaborative approach in which the participants work together in solving the problem (although they have different roles within the interaction).

The design for WOMBAT comprises a learning environment and a dialogue component. A schematic screen layout for this is illustrated in Fig. 4.2. The individual components are described in the following sections. This schematic screen may be compared with the screen dumps taken from the current implementation which are presented in §7.8.

4.3.3 A learning environment to support decision making

In this section, ways in which the availability of an appropriate computer-based environment might alleviate the data management and manipulation problems discussed in
§4.2.6 are outlined. This section discusses possibilities rather than presenting a detailed specification for the prototype implementation.

The option taken by most subjects - that of setting up a matrix - is assumed to be the most appropriate basic environment to support the decision making activity. It is recognised that the form chosen for the matrix was probably influenced by both the problem definition and the format of information presented on the data strips, but it is surely of significance to note that all the matrices created took the same basic form, being objectives against alternatives. For the prototype, this is the format of matrix which is adopted; in principle (as discussed in §9.5) it should be possible for the system and user to negotiate over the form of the information display (though this would only be relevant for a reasonably sophisticated user).

It is clearly necessary for the matrix to accommodate raw data. In addition, there should be a facility for that data to be scaled or annotated, as was done by group 1 in the study, in order to facilitate the application of compensatory methods such as the WOM. It should be noted that in the protocol study, all parameters were assigned numeric values.
(often on a rating scale of 1-10, as for the bodywork figures in Fig. 4.1.). It may be more desirable, in future, to allow for descriptive values of parameters to be entered where appropriate. In this case, scaling would involve assigning numerical values to qualitative descriptors. In the WOMBAT prototype, the raw values are inserted by the system when the participants agree to do this. Descriptors are not accommodated, and scaling is also managed by the system. In a fuller implementation, a mechanism to allow the user to participate in scaling values of parameters (such that, for example, 0 is unacceptably bad, 5 is average, 10 is exceptionally good, etc.) would be needed. This might look like Fig. 4.3. Such a facility does not exist in the prototype.

Fig. 4.3: a scaling device for WOMBAT

(movable arrows would default to being equally spaced, but could be placed independently of each other.)

As noted in §4.2.6, one pair had great difficulty in ordering their objectives. Had this pair been able to point to the objectives and move them around on the screen until they were happy with the ordering, their task would have been much easier. One attractive approach to this problem is that adopted by the designers of PROSPECT (§2.3.3), who ask users to identify the criteria in order of importance (most important first) and ask them to note how important each criterion is when compared to the most important by selecting a number of blocks as a measure of importance. In the prototype implementation, a rather simpler approach is taken, such that either the system or the user defines numerical values to indicate the relative importance of the agreed objectives. There is no facility to move objectives around on the screen.

It should be possible to have shaded, or otherwise marked, columns on the screen to denote alternatives which are less appealing but have not yet been finally rejected. This
Design, Decisions and Dialogue

might consist of a simple annotation facility. In the implementation, there are two possible annotations (made by the system, not the user): 'reject', which is put against any alternative which has been rejected, and 'best', which is put by the current preferred alternative (once one has been established). In an engineering context, the decision to simply note that an alternative has been rejected, rather than eliminating it from the screen, was made so that if it transpires later in the decision process that the alternative is unexpectedly attractive, the user can reconsider the decision step (or aspect of the problem definition) which originally caused the rejection.

Just as columns could be erased or marked, so rows could be dimmed, erased or moved to the bottom of the matrix as the user decided that objectives were of no further concern (for example, once fuel type information had been used to eliminate alternative C, it was no longer of relevance to the decision process as it did not distinguish between the remaining alternatives). This facility has not been built into the prototype. In terms of the theoretical design, forgetting about objectives is equivalent to reaching a mutual agreement that the objective is no longer important. However, in the case cited (regarding fuel type), the only reason that fuel type information might be thought necessary in the matrix is in the situation where the user has failed to appreciate the distinction between the demands and the wishes (or needs and wants) of the specification.

In addition to the main spreadsheet, other windows are required to display data and perform other data manipulations relevant to the decision making activity. As subjects spent a lot of time considering the alternatives rather than the problem, it is deemed appropriate to include a window to show detail about one alternative. This also permits the display of information not included in the matrix, such as pictorial data. In the current implementation, there are pull-down menus to enable the user to view the problem definition, information on alternatives, information on the objectives which the system holds beliefs about and (descriptive) information on the possible decision making tactics.

In a design context, the user might wish to perform calculations relevant to the current decision, for example to calculate a nominal annual fuel cost (mpa * cost pg / mpg =
cost per.) or to calculate an average mpg (typically calculated as 40% of urban cycle figure + 30% of constant 56mph figure + 30% of constant 75 mph figure). For such purposes, it would be useful to have a calculator available on-screen. This is available in the prototype implementation under the menu on the Macintosh™ computer.

Finally, in a design context, the selection process can often lead to the user wishing to make changes to either the problem definition or to the alternative solution(s). With this in mind, it should be possible for such changes to be made interactively. In the current implementation, a menu heading to accommodate such changes has been included, but the facility has not been implemented.

This is a very sketchy description of the requirements of such a learning environment. As the focus of the research reported here is on the development of the dialogue component rather than of the learning environment, this discussion is not taken further within the current research project. Probably the most important point to be borne in mind is that the provision of any environment is likely to guide the behaviour of users, both in making certain activities hard to perform and in presenting cues to suggest possible approaches to the decision making process.

4.3.4 The dialogue component

The subject of dialogue is covered in much more detail in following chapters. In this section, outline requirements of the ways in which the dialogue component should operate are presented. To summarise the requirements of the dialogue component, it should be capable of participating in dialogue, though not necessarily in natural language. The dialogue generated should be coherent (and each utterance relevant), and should be purposeful (both having an educational purpose and facilitating the solution of the problem). The style of interaction is to be collaborative; it is not assumed that either participant knows the answer, in the sense of being necessarily correct. It is to be mixed-initiative, and all dialogue moves are to be equally available to either participant (though it is also recognised that they have different roles within the interaction). The system must be able to negotiate (it is assumed that the user is able to do this!) in order to reach agreement.
In a mixed-initiative dialogue, in which the student has as much control over the course of the dialogue as the system, it is inappropriate to plan the content of utterances in advance, because the direction in which the dialogue moves is dependent on the goals of both participants. The dialogue is 'opportunistic' (Petrie-Brown 1989) and 'situated action' (Suchman 1987). However, neither Suchman's concept of 'situated action' nor Petrie-Brown's of 'opportunistic tutoring' informs how a computer agent might decide what to say in the current situation. The examples Suchman gives suggest that deciding what to do or say is based on instinct informed by past experience - i.e. on expertise. While this is generally the case for human conversants, our agent's conversational expertise does not extend to recognising the current situation (including the dialogue history so far) and immediately knowing what action to take. It is therefore necessary to reason about the delivery (or in our case the locutionary force) of a single utterance based on the current context. These issues are dealt with in more detail in the next chapter.

In deciding what to say, the system is trying to achieve many things simultaneously in terms of its teaching. It is trying to keep the user motivated and interested, to encourage the user to externalise and reflect on her thoughts, to make progress towards a solution, to help the user learn more about the problem and about this type of problem solving, etc. A single utterance can satisfy any number of these teaching goals to varying degrees. One of the design criteria for the dialogue component is that the teaching goals should be explicitly represented in the system, and should be used as one of the sources for reasoning about what to say next.

The design decision taken by Baker (1989) of using 'canned' text frames for system output and menus for user input is believed to be inappropriate in the current circumstances, as is the other attractive option of using a graphical interface for conducting the dialogue. There are two reasons for this; the first is that in seeking to satisfy multiple goals, both the system and the user should be able to build up quite complex utterances. A second reason to avoid menus or icons relates to the pedagogical goal of encouraging externalisation and reflection; the skill of recognising a valid answer which is already displayed on the screen is
Chapter 4: The design of WOMBAT

4.3.5 The form of the interaction: a discussion

Reichman (1986) discusses the interaction between a user and a computer windowing system, and compares the interaction to conversation, first between two humans (of unspecified relationship, but in some sense equals), a metaphor which turns out to be inappropriate for her purposes, and then between (controlling) human and assistant. She proposes a system of marking windows (e.g. by using colour) to indicate the relationships between them (for example, which are contextually related and which is currently in focus, or 'active'), and a range of mechanisms, such as mouse movements, for the user to switch between windows (depending on how great the shift is in terms of changing activity). She discusses the problem of the user having a view of several windows, with no strong indicator to show which is the currently active one (e.g. which window anything typed will appear in). Using her conversational metaphor, the analogue of a window is taken to be a context, and the act of switching between windows to be changing context. In considering the development of a mixed initiative dialogue system, rather than a simple reactive windows system, a better analogy might be that of two people working together using a piece of apparatus (whether that be a computer, a chemistry set or a pencil and paper). There are social conventions which dictate that if the control of apparatus is switched from one participant to the other then either something is physically handed over or there is verbal communication. If this convention is not observed then progress is impeded and blood pressure is raised. For example, if Alfred and Bertha are both trying to find the beginning of last night's wildlife program on the video tape but they are using different search strategies, one using the remote control and the other pressing switches on the video recorder, they are
unlikely to achieve their aim until either they start talking to each other about what they are doing or one of them relinquishes control.

Similarly, in developing a system with a learning environment and a mixed initiative dialogue component, to be consistent with the analogy of two people using a piece of apparatus, it is proposed that the interaction be focused in the dialogue window with formally negotiated control over the learning environment. It is worth noting that such a form of interaction has a clear symmetry.

For simplicity, at this stage, it is proposed that direct manipulation of the learning environment should be terminated by an explicit 'end' indicator by the controlling participant, at which point the dialogue window would become active again. This precludes the possibility of interrupts (probably a good thing!!) and ensures that the chances of the user getting confused about which window is active are minimised. This limitation is imposed largely because, while in the case of two people using apparatus there are at least two channels of communication (speech/hearing and touch, in handing over control), so that one partner can be manipulating the apparatus while the other is talking (or even directing the activity), in the case of a computer tutor with a learning environment there is currently only the one generally available channel (of user input via keyboard or mouse, and of computer response via screen display). As will emerge in later chapters, negotiation of control over the environment has not been implemented at all, so the details of how such negotiation might be managed are still somewhat unclear. (Only the system has access to manipulate the spreadsheet window, and only the user can view information in the pull-down menus.)

4.3.6 Forms of problem specification

In this thesis, it is assumed that any problem definition can be re-stated as a list of needs (absolute criteria), wants (relative criteria), and keywords. This is in accord with discussions of the issue in engineering texts. Clearly, the problem as given in the study described in this chapter does not fit in to this format - it would also be necessary to state in the specification how important a 'want' was ('luggage capacity is important'), and also to introduce a partial ordering on wants - for example, that economy is more important than
performance. Using this scheme, the problem presented to experimental subjects would be re-stated as:

\[
\text{need (price} \leq \text{£8000})(\text{luggage} \geq 25 \text{ cubic feet})(\text{seating} \geq 4)(\text{fuel lead-free})
\]

\[
\text{wants economy, performance, luggage}
\]

\[
\text{keywords family, only_car}
\]

This is clearly not a simple re-statement of the problem as given, particularly as regards the statement that 'they consider economy of greater importance than performance (within reason!)'. For example, Group 5 considered the performance of alternative F, as measured by acceleration, to be (absolutely) unreasonable, which would require an alternative formulation of the problem including an absolute requirement that acceleration be better than a certain figure.

Keywords, and other aspects of the problem definition, can be used to make further inferences about the real nature of the problem, in order to refine the problem definition later. This may be expressed as the participants developing a model of the problem (or a model of the user which, as discussed in §7.3.4, leads to a model of usage and hence to aspects of the design.) For example, the system might infer that:

- the lower the maximum price affordable, the greater the importance that should be attached to minimising price and running cost, and maximising fuel economy;
- for a family, additional factors become relevant, by implication, such as safety, reliability and comfort;
- for a couple or a single person, these same factors also have some importance, but not such a high importance as for a family;
- for a mechanic, servicing ease and parts cost should be considered;
- if this is the only car then luggage capacity and reliability gain importance; similarly, if this is the main car then luggage capacity is important;
- an older person is likely to consider economy and reliability to be important.

Reasoning of this sort, however primitive, enables the computer to have an opinion as a basis for discussion. The computer should also be able to justify its decisions and
assess the validity of justifications presented by the user. The implementation of reasoning of this sort is described in §7.3.4.

4.3.7 Knowledge and reasoning in WOMBAT

For the purposes of this research, the domain is being characterised as 'justified beliefs'. This characterisation is being used for both tactics and objectives.

For any tactic or objective that the dialogue component has information about, there are lines of argument both for and against, with justifications and relevance indicators to show when the line of argument is relevant. Each line of argument has a pre-defined numerical measure to indicate its strength. For example, the decision making tactic of Eliminating By Aspects (EBA) may be justified in terms of eliminating alternatives which do not satisfy the problem specification; this argument is a strong one in the context where such alternatives have not already been eliminated. The tactic may also be justified in terms of simplifying the problem, a weaker argument which is relevant as long as there are at least two candidate solutions. Conversely, an argument against EBA is that as long as a proposed solution satisfies the absolute requirements of the problem specification, EBA does not take into account the strengths of that solution. A second argument against EBA is that the outcome of a decision process conducted entirely by EBA is likely to be dependent on the order in which aspects are considered. A third argument is that such a decision process might not yield a unique outcome. Abstract arguments of this type are supported in the current implementation. In a further development, it should also be possible to generate further fictitious alternatives which illustrate the limitations of this tactic (for example, by including a strong candidate solution which would be eliminated if a particular aspect was selected as the first for EBA).

Similarly, for any objective there will be arguments for and against. In the dialogue extract presented in §1.3, for instance, the system believes that a reason for not worrying about flexibility is that it won't be needed very often, and that a reason for wanting flexibility is that the car might be used for d.i.y. jobs.
4.3.8 The iterative nature of the design process

As has been discussed in §2.1, designing is an iterative process, in which the designer learns about the problem through considering possible solutions to it. While WOMBAT is being considered as a stand-alone decision aid in this context, it must not be forgotten that the process of evaluating design concepts in this way will enable the user to refine their understanding of the problem, and of the alternative design concepts. So they should be able to modify the problem definition and add to or modify the list of available possible solutions as their understanding develops. It might be argued that the main difference between the activities of 'evaluation' and 'selection' is that evaluation takes account of the iterative nature of the process, and accommodates the designer learning about the design, whereas selection is a once-off activity based on the assumption that the problem specification is correct and the alternative solutions are unchangeable.

4.4 Summary

In this chapter several lines of inquiry have been followed with the aim of presenting an outline design for an Intelligent Educational System to support design evaluation in engineering. Consideration has been given to:

- the format of the problem specification,
- the facilities available within the learning environment, and in particular the spreadsheet,
- the requirements of the dialogue component including its ability to reason opportunistically and to be non-prescriptive,
- negotiation of control over the environment,
- types of reasoning and knowledge structures.

In the following three chapters, attention is focused on the dialogue component - on relevant literature, on the theoretical design and on the implementation.
Chapter 5: Work on dialogue and agents

This chapter presents a technical context for this thesis, in that the design of the dialogue component takes into account past work on computational models of dialogue and is actually based on recent work in the field of agent theory. Many of the ideas expressed in this chapter have had a direct influence on the design of the dialogue component. In particular, the work of Kiss on agent attitudes and the action cycle has been the main influence on the definition of the theoretical agent model (§6). Also, the work of Grosz and Sidner on coherence and topic has been used as a basis for the implementation of topic stack control (§7.3.3).

The early sections of this chapter (§5.1 - 5.3) are a review of work on dialogue, first considering the meaning of the term 'dialogue' then considering it as being purposeful (§5.2) and coherent (§5.3). Much of the work reviewed in these sections is largely descriptive. The notion of a dialogue participant as an agent (i.e. as an entity which can generate utterances) is introduced in §5.4. Sections 5.5 and 5.6 present a more general review of agents, first (§5.5) considering in some detail how an agent decides what to do (or say) and how an agent organises its activity, and then (§5.6) discussing the attitudes (beliefs, desires etc.) which an agent might hold.

5.1 Introduction to dialogue

Several reviews of the dialogue literature already exist - see for example (Kiss 1986), (Petrie-Brown 1987), (Kass and Finin 1988) and (Baker 1989). The literature reviewed here is only that which is directly relevant to the thesis.

The term 'dialogue' is widely used to describe a range of interactions involving more than one language user, though in literature emerging from the U.S. the term 'discourse' is perhaps more common. In order to articulate more clearly one of the aims of the work reported here, a more specific definition of the term 'dialogue' is proposed.

Petrie-Brown (1989) suggests that most tutorial interactions in the past have
consisted of the action and reaction of two autonomous language users (discourse), rather than an interaction of two language users (dialogue). Suchman (1987) makes a related point, observing that conversation (dialogue) is not so much alternating turns as joint action, accomplished through the participants' continuous engagement in speaking and listening. Petrie-Brown (1989, p.195) goes on to say that

"we should build generative models of dialogue addressing the intentions of both participants in the interaction rather than only applying analytic models concerned with the purpose of one interactant"

(whether that be the tutor, as in most extant IESs, or the student, as in learning environments). Kiss (1986) expresses the same idea, saying that the essence of High Level Dialogue (which is concerned with the overall structure of dialogue rather than the detailed structure of a communicative act) is the support of cooperative interaction, which involves the sharing of responsibilities for forming plans and executing them. Cooperativity is discussed in §5.3.2. The distinction between discourse and dialogue cannot be clearly drawn, though the extremes are recognisable (from the non-cooperative mutual points-scoring evident in many industrial disputes to highly collaborative joint problem solving as illustrated in Fig. 5.1, taken from the protocol study described in §4).

---

**Fig. 5.1: example of dialogue on car selection**

"That's a good safe bet - B - based on the criteria... so far.. I'd say B wins"

"Yes, because they want..."

"...cos it's cheap enough in terms of 8000.."

"Yes, they're..."

"...it's got a large boot.."

"... not too kee...too bothered about acceleration.."

"That's right"

"... which it's certainly not going to have.."

"..and it's cheap"

"..and it's cheap"

---

Petrie-Brown draws the distinction between discourse and dialogue as follows: in
"the agent can produce/understand connected text sequences and can organise such sequences according to her own goals [whereas] in generative dialogue, any agent must be regarded as both a producer and an understander, i.e. as a complete agent in interaction with some other complete agent(s) attempting to achieve goals through the use of language." (p. 196)

If the student is considered to be a 'complete agent' (which seems to be a reasonable assumption, though one which few tutoring systems are built on!) then we need to build models of dialogue, and not just of discourse. Our view on what constitutes an appropriate model of the interaction must be a reflection of our view of the status of the student as a participant in that interaction, and if students are to be treated as responsible partners in their own learning then fuller models of dialogue need to be developed to support the interaction.

Few tutoring systems to date have explicitly aspired to incorporating models of dialogue per se. Probably the nearest we get to a working example is Baker's (1989) KANT system (§3.4). While it is not clear that it achieves its stated aims of interactional symmetry or dialogue particularly effectively in its current partially implemented form, it serves to articulate appropriate aims for the interaction.

The current state of computational linguistics research is such that neither the semantic interpretation of sentences nor the structure of super-sentential units is well understood. While natural language (NL) systems exist which can understand grammatically correct sentences, no systems can perform well with more natural (or normal) utterances. Also, with an unconstrained NL interface, the user of the system often has difficulty establishing the limitations of the system (in terms of what it is capable of understanding or doing) as there are too few cues to inform the user of the boundaries of the system. The state of the art as regards the generation and comprehension of natural language utterances is such that it is not yet possible to incorporate the effective use of natural language in IESs.

Most adults have little difficulty in recognising the overt intention of the speaker in many situations in which the surface form of the utterance suggests a different intention - as
in the oft-quoted "can you pass the salt?" which has the surface form of an inquiry rather than a request. At the age of 3, the author's daughter Emily went through a phase of initiating conversations with queries such as "Why did Thomas say 'Hurry up, Henry'?" This illustrates the separation which can be (but is not generally) made between an utterance and its purpose. The distinction as expressed in this example is between the locutionary act and the illocutionary act, where the locutionary act is the act of speaking and the illocutionary act represents the 'rhetorical force of the utterance' (Novick 1987). Additional distinctions are occasionally made in the literature; for example, the perlocutionary effect is the result of the act - i.e. in this case the hearer's perception of the speaker's intention, while the perlocutionary intent is the speaker's intention of what the act will achieve for that speaker.

High Level Dialogue (HLD) includes non-linguistic communication, such as gestures, direct manipulation of objects, or other actions. Clearly, not all of these communication forms are available in interaction with a computer, but some (such as manipulation of a learning environment) might well be. This idea has been extended into the concept of 'speech acts', or the 'use of language for making things happen' (Kiss 1986, p.15) as outlined below (§5.2.1).

While the field of computational linguistics is a rich one, only a small proportion of the work from that field has any direct relevance to the issues involved in the computational generation of dialogue. Much of the work relates to the analysis of natural language utterances, for example studying the structure of sentences or the resolution of anaphoric references. Of the literature concerned with the higher level structure of dialogue, much refers only to the post hoc analysis of human-human dialogue. While this work is essential to our understanding of the structure and communicative power of discourse, it contributes little to our understanding of how to generate purposeful, coherent utterances while participating in dialogue.

It is now widely agreed that while human-human dialogue (in particular, tutorial dialogue between human teacher and student, or the more common tutorial discourse in
which one teacher addresses many students) can inform our work on the generation of human-computer dialogue, it does not provide a completely satisfactory model from which to work. As Self (1989) observes, the analogy of IES as teacher can be over-stretched; he proposes an alternative analogy, of IES as cognitive gymnasium, where the student approaches the IES expecting to leave intellectually invigorated. This raises the issue of the expectations of the user; just as the recipient of a letter has different expectations of the likely contents, depending on whether the address on the envelope is hand-written, typed or obviously mass-produced (as in computer-generated mail-shots), so a student will have different expectations depending on whether they are approaching a computer or a human tutor. This intuition is supported by experimental results of Cohen (1984), who studied 5 modalities of communication, including telephone conversation and communication via "back-to-back" teletypes, in which a remote expert guided an apprentice in a problem solving task. While Cohen focused his attention on one aspect of the dialogues (how the expert referred to the objects which the apprentice was meant to identify), his results led to the more general conclusion that the experts' utterances were strongly influenced by the communication medium available to them.

Considering the issue of how closely IES dialogue should be modelled on studies of human teachers, Petrie-Brown (1989) proposes that

"researchers should not necessarily be trying to model the teaching method of tutorial dialogue but that they should perhaps accept that artificial intelligent tutoring dialogue is a new concept, possibly a new educational paradigm, and exploit the possibilities of a new medium."

Elsom-Cook (1989) echoes this view, saying that

"Although it seems clear that we must examine human teaching, since humans are the only examples of teaching which are available for study, this does not imply that our tutoring systems should be constrained to human-like interaction. While taking human interaction as the base of study, we must continually consider the 'space of possible educational interactions' which may involve forms that cannot be executed by a human teacher",
though he does not expand on what these forms might be.

One particular feature of human-computer dialogue is that the communication bandwidth between human and computer is currently quite limited; one cannot, for example, take cues from eye contact, gestures or the existence of micro-pauses or intonation in the conversation. Thus interruptions, so common in human-human dialogue, become much less natural in human-computer interaction; perhaps a more appropriate analogy is that of radio contact, where turn transitions are made explicit with 'over', and conversations are terminated with 'over and out'.

Various researchers allude to different levels at which a dialogue takes place, and the different purposes that utterances serve. An exhaustive treatment of this subject is given by Novick (1987). He presents a hierarchy of conversation levels, from utterances which deal with the subject at hand, through repair mechanisms (which clarify meanings), to mechanisms (generally non-verbal) which govern turn-taking and other aspects of dialogue control. Reichman-Adar (1984) discusses the role of clue-words in removing the need for extensive meta-dialogue. Baker's (1989) negotiated tutoring revolves rather (in its current form) around meta-dialogue about the future; both system and user negotiate at length about what they will talk about and how they will do it.

In this thesis the aim is neither to emulate human teachers nor to emulate human-human dialogue, though work on the generation of tutorial dialogue is informed by studies of both. Human-computer tutorial dialogue is a different genre; in the foreseeable future there will be limitations such as a lack of common sense reasoning and inability to detect non-verbal cues, but a system has potential to integrate a range of resources (databases, sound and visual images) in an individualised way to facilitate the learning of students, and the effective use of such resources requires a sophisticated level of interaction between the system and user. At the simplest level, the aim is to develop a system which can engage in dialogue which is acceptable to the human participant. Acceptability includes features such as coherence and purposefulness.
5.2 Purposeful dialogue

Some work in computational linguistics has focused on the purpose of utterances, largely in terms of how the utterances further the goals of the conversational participants. In WOMBAT the role of utterances in furthering the goals of the participants (in particular in furthering teaching goals) is important. However, as will emerge, much of the relevant past work is largely descriptive.

5.2.1 Speech acts

In speech act theory, utterances are viewed as having a propositional component and a speech act (or functional) component. Searle (1969) identifies five classes of speech act:

- assertives, which state some proposition, committing the speaker (to a greater or lesser degree) to a belief in the truth of that proposition,
- declarations, where the performing of the speech act effects a change in the state of the world (such as marrying people),
- commissives, which commit the speaker to some future course of action,
- directives, which seek to commit the hearer to something, and
- expressives, which express a psychological state (e.g. praising).

The original theory has been extended to deal with various aspects of speech which it could not originally explain adequately. One example is the introduction of 'indirect speech acts', in which the act achieves the desired effect through the hearer's recognition of the speaker's plan rather than through the direct form of utterance used. The speech act formalism has been criticised, for example by Levinson (1981), on various counts, and it is now being replaced by attempts to derive utterances from a more basic theory of action. One approach to this is the development of dialogue as an aspect of agent theory as outlined below (§5.4). Although the basic idea that utterances can serve functions just as actions do is useful, speech act theory is ultimately a descriptive tool which accounts for many of the phenomena observable in natural language dialogue, but it does not explain adequately how one might go about generating utterances 'from first principles'. However, much work in
HLD, such as that of Cohen and Levesque (1985, 1990b), is based on extensions of speech act theory.

### 5.2.2 Dialogue games

Levin and Moore (1977) propose a view of language as consisting of multi-sentential knowledge units that are specified primarily by the speaker's and hearer's goals; they call their structure 'dialogue-games'. This structure is a representation of the structure of language, based on the function (rather than the topic) of a dialogue. Levin and Moore identify several types of systematic interactions such as helping (though their view of this seems to be basically remedial help), information seeking, instructing and griping. In each type of interaction, the initiator wants something, and initiates the dialogue game in order to satisfy that want. Levin and Moore define the parameters and components of the dialogue games they have identified; the structure of these is unimportant in this context. Once a dialogue game has been bid and accepted (i.e. the hearer has recognised the speaker's goal, and is willing and able to cooperate), according to Levin and Moore's model, the two participants each pursue the sub-goals specified for their role in the dialogue game, and by the time the dialogue game is terminated the higher level goals of the participants (for which the dialogue game was initiated in the first place) will have been satisfied. This may be an adequate description of what can be observed in naturally occurring dialogues (though most of the protocols studied by Levin and Moore were from one type of situation - computer users interacting with the operator - so even this might be too strong a claim), but it fails to explain why these effects are observed, to explain where the higher level goals of the dialogue participants come from, to explain how the hearer recognises the initiating speaker's higher level goal, or to integrate any formal representation of the beliefs of either participant, beyond saying that they are contained in either long term memory or 'workspace'.

Levin and Moore (1977) compare speech act theory and dialogue games, observing that both specify ways of interpreting individual utterances, depending on both the words (syntax) and the context. Both depend on knowledge of the participants; however, speech
acts are unilateral actions (generally referring to a single utterance) whereas dialogue games by definition involve two participants and encompass multiple utterances and turns. An intermediate level of analysis is adjacency pairs (Schegloff and Sacks 1973), in which an utterance is viewed as a response to a preceding utterance. This notion is useful, in terms of helping our understanding of how a response might satisfy the expectation of the previous speaker, but again it takes no account of the higher level structure of the dialogue.

The dialogue games approach has been useful; for example, past work of Elsom-Cook (1985), and Baker (1989) has been based on this structure. However, as Elsom-Cook (1989) notes, while it gives one level of analysis for dialogue, it is ultimately a descriptive tool. More specifically, this approach does not accommodate the desired representation of teaching goals (see §4.3.4).

5.2.3 The generation of purposeful utterances

Some work has been done on implementing systems capable of generating purposeful utterances. However, none of these systems can engage in sustained opportunistic dialogue.

Power (1979) developed a computer program which modelled a conversation between two robots, cooperating to satisfy their own goals in a limited world. The robots are initialised with different goals, abilities and beliefs. Each robot plans to achieve its goal, and the robots share conversational procedures by which they communicate their plans and beliefs. Power criticises his own work as not having an adequate representation of how utterances achieve their desired effects. In addition, only small sections of dialogue, such as adjacency pairs, are modelled so Power does not include any representation of topic or context, as is needed for coherence in a more complex domain.

More recently, Draper and Button (1990) have extended Power's work to develop a fuller model of conversation as planned action. They distinguish four levels of goal which the dialogue participants may have. The first is the external goal which the participant is trying to achieve. (In principle, there may be more than one such goal, but this possibility
has not yet been explored.) The second level is a goal within the interaction which may involve several exchanges - for example, reaching mutual agreement. The third level is goals which can be achieved in one utterance. This level corresponds most closely to the notion of speech acts or adjacency pairs. The fourth level (on which Draper and Button have not yet worked) relates to aspects of the conversation such as turn-taking and coordination. Their model seeks to develop the symmetry between physical and conversational actions, in that their robots (which operate in the same limited world as Power's did) may plan to achieve a goal through conversation in situations where it could also be achieved through physical action, whereas Power's robots only resorted to conversation when physical action was not possible.

5.3 Coherent and symmetric dialogue

5.3.1 Coherence and relevance

Novick (1987) discusses at length the qualities which a dialogue must have in order to be considered 'coherent' - a term which he defines as 'making sense to the conversants'. As with any such definition, it does not define the space of possible coherent conversations precisely.

Grosz and Sidner (1986), whose work is based entirely on analysis of task-oriented dialogues, propose that in order for a conversation to be coherent, the conversants have to have a shared discourse purpose (i.e. be trying to achieve the same thing). Novick argues that this conclusion is too general, and that the requirement for coherence is that the conversation participants have to cooperate, and must have a shared model of the conversation.

Reichman (1978) takes a similar view, arguing that conversational coherency depends upon a lack of conflict between the discourse models of the participants. In her view, as a conversation proceeds, each participant builds up a model of the discourse which includes the conversation's context space structure, a notion of the present discourse topic and a list of items being focused on. She views the context space structure as being
Design, Decisions and Dialogue Chapter 5: Work on dialogue and agents

Reichman sets up a taxonomy of context spaces, and discusses conversational moves between them; in a later paper (Reichman-Adar 1984), she develops an Augmented Transition Network (ATN) formalism to implement such conversational moves; however, while she claims that such a network can be used to generate discourse, it is not clear how the nodes (context spaces) in her network are generated, unless they are coded in advance, or how a computer system might select which route to traverse through the network, so this work must be regarded for the present as descriptive, rather than being capable of participating in new purposeful discourse.

Reichman-Adar (1984) distinguishes between two types of context space - namely 'issue' and 'non-issue' spaces. 'Issue' context spaces are those which deal with general topics, while 'non-issue' context spaces deal with supporting material such as examples, or illustrations, of topics. A parallel can be drawn here with the distinction made by Baker (1989) between abstract ('issue') and concrete ('event') claims, and also with Elsom-Cook's (1985) 'illustrate' ('event') dialogue game. A similar distinction can be drawn in the author's work, in terms of abstract justifications for beliefs stated (e.g. "Approach $a$ is generally preferable to approach $b$ because approach $b$ fails in situations of type $<$situation1>." ) versus concrete justifications (presenting a specific example where approach $a$ works but approach $b$ does not). This point is addressed in §4.3.7.

Grosz and Sidner (1986) distinguish three aspects of the discourse - namely the attentional, the intentional and the structural aspects. They define attention as being the topic, or focus space; this relates closely to Reichman's context space. Intention refers to the purpose of the discourse segment, and structure to the grammatical structure of the discourse. This work is purely descriptive, but serves to identify some constraints within which any dialogue participant has to operate if its utterances are to be recognisably coherent. In particular, as well as maintaining its version of the 'shared model of the dialogue', it has to keep a record of active, open and closed topics. Grosz and Sidner present a stack structure for the topic, such that the active topic is the one at the top of the stack, while other open topics are also in the stack. Topics are removed from the stack when they are closed. Baker (1989) observes that
"A crucial question for research into computational modelling of dialogue concerns the relationship between topic and intention. Most authors agree (contra Reichman) that they should be given theoretically separable but interacting structures, but that dialogues are primarily defined in terms of their goal or function, within local topic constraints."

However, coherence does not depend solely on the focus of the conversation, any more than it does solely on the function. Ultimately, what matters is that dialogue is perceived as coherent by the participants. Each utterance should be clearly relevant in the current context, so as to be easily integrated with the hearer's model of the dialogue.

### 5.3.2 Cooperativity and benevolence

The distinction should be drawn between cooperativity and benevolence. As Baker (1989) observes, arguments can be cooperative rather than adversarial in the sense that the cooperatively agreed goal of participants is to promote mutual belief revision, or to cooperatively arrive at some conclusion. To re-interpret this notion, without cooperativity at a dialogue level there can be no dialogue, but it is not necessary for participants to cooperate in the sense of always responding as their dialogue partner wishes if to do so conflicts with their own goals (Galliers 1989). The latter interpretation of cooperativity is referred to as benevolence. While one of the values of an agent might be to be benevolent, this is not an essential feature of it, whereas to be cooperative at a dialogue level is necessary if an agent is to be a competent dialogue participant.

In the context of this thesis, being benevolent requires that the system should respond directly to the user's expectation (for example, if the user asks a question, they have an expectation of getting an answer to it). Being cooperative is a less restrictive requirement, which necessitates the user's expectation being acknowledged, though not necessarily being directly responded to.

### 5.3.3 Symmetry

The view of what constitutes an appropriate model of the interaction reflects the
perception of the status of the student as a participant in that interaction. If students are to be treated as responsible partners in their own learning, they have to be given a measure of control over the course of the interaction. There has to be a balance between guidance and control. Where exactly this balance lies depends on the approaches of both tutor and student. The definition that the interaction is to be mixed-initiative means that either participant can take the initiative - that the locus of control does not lie exclusively with one participant - though in any given interaction the relationship developed between the participants may lead to one of them taking the initiative more than the other. The interaction is to be symmetric, in the sense that any dialogue move is equally available to either participant; both have the ability to make the same range of utterances. As stated earlier, symmetry relates to the interaction possibilities available to the participants, and not to their beliefs or their roles in the interaction.

5.4 Discourse participants as agents

Much of the work discussed above is descriptive rather than defining how a dialogue participant might decide what to say in any given context. Any dialogue participant is, in however limited a sense, an agent, in that it is participating and involved in the interaction, and has an effect on that interaction.

In §5.3.1, the work of Grosz and Sidner on attention and intention in discourse was discussed. More recently, Grosz and co-workers have been looking in more detail at 'planning' models of discourse - in particular, at discourses which centre around the completion of a collaborative task. For example, Balkanski (1990) extends a formalism (originally due to Pollack) for describing collaborative activity in terms of relations such as 'generation' and 'enablement', together with descriptors for simultaneity (things to be done at the same time), conjunction (things to be done, with no temporal sequencing), sequence and iteration ('do until' etc.). This descriptive language is used by Grosz and Sidner (1990) in their discussion of how shared plans are derived, and how collaborative activity is coordinated through language. This recent work is an advance on the earlier work in that it eliminates the master-slave assumption inherent in the earlier work. However, it is still
basically descriptive rather than providing a mechanism for generating utterances, and is still based on the 'planning' model which underpins the dialogues on which their work is based.

A similar approach is taken by Pollack (1990), who analyses discourse in terms of the underlying plan of the speaker, which she describes as the speaker having a 'particular configuration of beliefs and intentions'. This formulation simplifies the problem of plan recognition by a hearer - particularly in the case of buggy plans (in that it is likely to be the speaker's beliefs rather than intentions which are buggy!).

A somewhat different approach is taken by Cohen and Levesque (1985), who derive a theory of communication from a formal theory of rational interaction. Their theory is again descriptive (rather than generative), and is based on a 'possible worlds' semantics. What it does is provide a formal description of how a speech act can achieve the desired effect based on assumptions of the speaker and hearer being rational. They illustrate the application of their framework to the illocutionary act of requesting. In more recent work (1990a), Cohen and Levesque consider the formal definitions of intention and of commitment. This is discussed below (§5.6.3).

In this work on discourse, what emerges is the notion of a discourse participant being an agent, where an agent is an 'integrated natural or AI system that is capable of goal-directed action through which it autonomously pursues its interests' (Kiss 1989), or an 'intentional system' (Seel 1989). An agent may be viewed as a system which is continually evaluating potential actions with respect to its beliefs and goals, making commitments to action, and carrying out those actions, where actions may be external (saying or doing something) or internal (updating beliefs etc.).

5.5 Planning versus opportunism

Consistent with the approach taken by Pollack and by Grosz, much of the work which comes under the general banner of 'agent theory' is concerned with planning. It is recognised that people engage in planning to a greater or lesser degree depending on the circumstances. For example, in the experimental situation devised by Hayes-Roth and
Hayes-Roth (1979) in which subjects were given a large number of tasks to do and asked to do them in such a way as to satisfy various constraints (such as time constraints on certain activities), it was natural for them to engage in substantial advance planning, whereas in the situation discussed by Suchman (1987) of a canoeist shooting the rapids, all that will be done in advance is general decision making such as deciding an overall strategy and aiming to go (say) to the left of a particular large rock; in this situation the canoeist will react opportunistically to the forces on the canoe based on past experience and the overall goal of getting to the end of the rapids. Whether activity is planned or (in Suchman's terms) situated depends on a large number of factors such as the time available, the penalty of failure, the level of expertise and the nature (e.g. the predictability) of the activity.

A formalisation of this split can be observed in work on agents, most notably by Kaelbling (1986) and more recently by Downs and Reichgelt (1991). Kaelbling proposes a layered architecture for an agent which has sub-components with different levels of competence and acts in accordance with the output from the most competent sub-component which is able to propose an action. The architecture has been tested in a prototype implementation of a robot which is capable of navigating itself along a corridor. In this case, the robot has sensors which can detect the proximity of the walls and has three action sub-components which can direct activity. The lowest level of these simply decides to stop or go; this component can ensure that the robot does not hit the walls, but is unlikely to get the robot all the way down a corridor. The second level component can direct the robot to turn or to go; commands from this component naturally lead to a zig-zag path down a straight corridor. The third (and in this case, highest) level component is capable of more complex reasoning about the direction of travel, but needs more sensory input, and takes longer to perform calculations. Therefore the robot has to operate on the lower levels while the highest level is planning but, as long as its output is still valid by the time it has been calculated, the output from the highest level will yield the straightest route along the corridor.

Downs and Reichgelt (1991) have considered the issue of how the levels in a multi-level architecture such as that of Kaelbling might be interrelated. Whereas Kaelbling's levels are essentially independent of each other (for example, in an implementation each would...
have to be separately coded, and changes made to one level would not automatically be reflected in any other), Downs and Reichgelt have focused their attention on how the output from a higher level might be retained in a suitable form for future use by a lower level. So, for example, if the system comes upon a situation it has not encountered before it can engage in a planning process in order to ascertain what to do, then compile the resulting plan into a form to be retained as new situation-action rules for use if the same situation arises again. They also discuss the aim of decomposing these situation-action rules into a form which would be directly executable. A complex planned action causes the generation of several situation-action rules, one for each basic action invoked in executing the plan. So for example, in a 'blocks world', if the goal were \( on(a,b) \), then the following situation-action rules might be generated:

\[
\begin{align*}
\text{IF clear}(a), \text{clear}(b), \text{free}(\text{hand}) & \text{ THEN pick-up}(a) \\
\text{IF clear}(b), \text{held}(a) & \text{ THEN stack}(a,b)
\end{align*}
\]

These situation-action rules may be regarded as defining a schema for achieving the goal \( on(a,b) \), in that once the situation has been recognised the actions are carried out in sequence without further deliberation.

Downs and Reichgelt present the layered structure of their agent, together with its perception module and effectors, as its 'architecture'; more generally, Doyle (1988) describes an architecture as a 'system of government'. These different descriptions reflect a differing emphasis, rather than a basic incompatibility of views; the definition of an agent architecture includes an identification of the components (or modules, or layers) which make up the system, defines the reasoning (or processing) capability of each component, and also defines the interaction between the components.

Kiss et al express the recognised need to incorporate both planned and reflex actions within the agent's architecture in their notion of an 'action cycle', which defines the operation of the agent. They describe the cycle as follows (slightly simplified from Kiss et al (1988)):

1) Respond to interrupts by either carrying out the corresponding conditional intention (if the interrupt was expected - i.e. was being 'waited for') or generate a want with an
appropriate resource allocation and priority (depending on the importance of the situation).

2) Identify unsatisfied wants and form a preference ordering of them based on the associated costs and benefits.

3) Examine the goal of the top-ranking want and identify relevant actions (or plans) for this goal using means-ends reasoning. Reflex action is resorted to in extreme cases, causing the agent to form an unconditional intention and 'jump' to step (8).

4) Form a preference ordering of actions based on the values of the situations which would result from performing them and the cost involved in executing them.

5) Expand the most important action to a level of detail merited by its importance.

6) Revise the expansion in the light of the details (noting conflicts etc.); this and the previous step comprise traditional planning activity.

7) Form intentions to execute a suitable set of actions.

8) If the intentions are unconditional then execute them. Otherwise set up expectations (or wants) for the relevant conditions and suspend the intentions to await satisfaction of them.

9) Check the results (intended and side-effects) of the actions. Determine the importance of any discrepancies and allocate resources to a want for dealing with them.

10) Repeat this cycle indefinitely.

More recently, a first prototype system (Demo1) has been implemented (Kiss and Brayshaw 1989). This implementation incorporates some of these ideas and demonstrates their application in the domain of file system management. In Demo1, the agent anticipates the results of performing each of the possible basic actions which are available to it, and selects the most appropriate one based on its values and means-ends beliefs. Some of the features of the system are that it employs the WOM algorithm (§2.2.3) to choose between alternative possible actions, that it does no advance planning, and does not make use of schemata, and that it adjusts the weights of its values based on the situation (so for example, the weight on its value of satisfying the user is higher if the user is an expert computer user - for example, the system manager - than if the user is a novice). As the work of Kiss has
been a significant influence on the research reported here, a comparison between the work of Kiss et al and that of the author is included in §B.6.2.

Doyle (1988) provides a rather simpler description of how an agent might become committed to action. He distinguishes between wanton actions, which are determined by desires, and deliberate actions, determined by intentions. He presents an abstract volitional procedure for deliberate action:

1) Select the next intention to carry out.
2) Select the method by which to carry out the selected intention.
3) Carry out the selected intention using the selected method.
4) Repeat these steps.

A similar procedure is presented for wanton action:

1) Select a desire.
2) Select an action relevant to satisfying the selected desire.
3) Carry out the selected action.
4) Repeat these steps.

Doyle notes that taking action requires information about the available actions, about their expected consequences, and about the utility of those consequences to the agent.

Such information is included in an implementation described by Georgeff and Ingrand (1989). Georgeff and Ingrand are concerned with the development of what they term a 'procedural reasoning system' which can operate in time-critical situations such as handling malfunctions in a space shuttle. They describe a system called PRS which consists of a database of facts (or beliefs) about the world, a set of goals to be realised, a set of plans (or schemata) which define how certain sequences of actions may achieve given goals or are appropriate to react to particular situations, an intention structure containing plans selected for (eventual) execution and, finally, an inference mechanism to manipulate all the components. A plan consists of sequenced sub-goals or of a primitive action which is directly performable by the system. Fixed decision making processes for selecting appropriate plans are hard-wired into the system for use in time-critical situations. If
conditions indicate that it would be advantageous to override such a decision then meta-level planning is invoked to perform the decision making. As will emerge (§6), Georgeff and Ingrand's approach of integrating decision making with the use of hierarchical schemata is similar to the approach being taken in this research.

In this section, passing reference has been made to many of the types of attitude that an agent might possess (for example, intentions, desires and beliefs are the main classes of attitudes discussed by Doyle (1988)). In the following section, the issue of agent attitudes is dealt with in more detail.

5.6 Agent Attitudes

5.6.1 Classes of agent attitudes

A rational agent is viewed by Kiss (1989) as having attitudes which may be classified as cognitive, conative and affective. An attitude is defined by Kiss as a relation between the agent and the world. Cognitive attitudes include knowledge and belief. Rationality places constraints on the attitudes which may be held simultaneously; for instance, an agent cannot believe P and (not P) simultaneously (though it may hold inconsistent beliefs, as long as the inconsistency remains undetected). The possession of cognitive attitudes is necessary, according to Kiss, in order to support the requirement that the agent be able to react to environmental inputs.

Conative attitudes, necessary for the agent to be able to act, include wants, wishes, intentions etc. Kiss (1989) proposes that a "rational agent will form intentions to do what it believes will on balance best satisfy its desires". Doyle (1988) expresses this idea as "Rationality... means taking actions of maximal expected utility".

Affective attitudes, necessary for autonomy, include values, likes and dislikes. Rationality imposes constraints such as the agent valuing what it believes will produce pleasure.

Some attitudes have an intensity aspect reflecting, for example, the strength with
which a belief is held. Relative intensities, particularly of values, are essential for making choices, for example deciding which of several alternative possible actions is preferred.

The principal attitudes discussed by Doyle (1988) are intentions, desires and beliefs. In his formulation, intentions can be to do or to achieve. All can be either absolute or relative, giving rise to priority (of intentions), preference (of desires) and likelihood (of beliefs). There is assumed to be a partial ordering of each attitude.

5.6.2 Beliefs and knowledge

Rosenschein and Kaelbling (Rosenschein 1985, Rosenschein and Kaelbling 1986), in their work on situated automata, take a view of knowledge (or beliefs) as not necessarily requiring that the system have an explicit internal representation of each proposition but rather that an observer believes that the system knows the proposition because its behaviour is entirely consistent with it knowing it. Winograd and Flores (1986) also argue against what they term the 'rationalist' approach, arguing that meaning does not exist in propositions, but in contexts. An example they quote goes as follows:

A: Is there any water in the refrigerator?
B: Yes.
A: Where? I don't see it.
B: In the cells of the eggplant.

In this example, while B's response is literally true it is inappropriate to the context (in which A is presumably looking for something to drink). While accepting the strength of the argument, the work reported here is in the rationalist tradition, in which propositions are explicitly encoded in the machine, and the remainder of this discussion is based around work in this tradition. In this thesis, the view is taken that propositions derive their meaning from the defined context of discussing properties of cars and how decisions are made.

In the formalism of Cohen and Levesque (1985), beliefs necessarily have to be consistent. They base their logic on four primary modal operators, including BEL and BMB. (BEL x p) means that proposition p follows from agent x's beliefs, and (BMB x y p) means that proposition p follows from agent x's beliefs about what is mutually believed by
agents $x$ and $y$. The introduction of mutual belief as a primary operator may be viewed as a short-hand for the infinite series $(\text{BEL } x \ p)$, $(\text{BEL } x \ (\text{BEL } y \ b))$, $(\text{BEL } x \ (\text{BEL } y \ (\text{BEL } x \ p)))$, $(\text{BEL } x \ (\text{BEL } y \ (\text{BEL } x \ (\text{BEL } y \ p))))$ etc., but it should be noted that the belief about the existence of a mutual belief is held by one agent (in this case $x$); it does not necessarily follow that the mutual belief is also held by $y$.

Elsom-Cook (1990b) proposes that knowing is akin to mutual belief. This definition is unsatisfactory - for example, two people might both believe that God exists; they might also both believe that they share a mutual belief that God exists; in this case they hold the mutual belief that God exists, but it could not reasonably be said that they know that God exists (if only because they are also aware that there are many other people who do not hold this belief). In this thesis, the only cognitive attitude encoded in the model is belief. This is consistent with most existing work in agent theory.

Doyle (1979) presents the design for a truth maintenance system (TMS) to organise the management of beliefs. He argues that to study rational thought, we should study justified beliefs or reasoned arguments, and ignore questions of truth. Therefore in his system each belief which might be held is associated with a 'support list' which consists of an ordered pair of other beliefs. Beliefs in the first set are reasons for holding the original belief, and those in the second set are reasons for not holding the belief (note: not holding belief $P$ is not the same as holding belief (not $P$)). The belief is in (i.e. held) if each belief in the first set is in and each belief in the second set is out (i.e. not held). He discusses how belief revision might take place as new beliefs are acquired to maintain perceived consistency within the set of in beliefs. Doyle's TMS is an example of a justification based truth maintenance system, and as such has some features in common with the truth maintenance mechanism implemented in WOMBAT.

One aspect of beliefs not explicitly dealt with by Doyle (1979) is that of strength of beliefs. In Doyle's TMS, whether a belief is in or out depends on the contents of the support list. In more recent work (Doyle 1988) he proposes that there is a partial ordering among beliefs. (This view is also expressed by Kiss (1989).) With such a partial ordering, the
agent would be more reluctant to give up certain beliefs than others if a conflict is detected between held beliefs. As discussed by Galliers (in press), the strength of a belief is likely to be based on the quality of the supports (or endorsements) for that belief (e.g. I am more likely to believe something I witnessed directly than something heard 'third hand' from a friend-of-a-friend-of-a-friend without other supporting evidence). Collins and Michalski (1989) present examples of beliefs held simply due to the absence of the opposite belief (e.g. presented with a new type of bird, one is likely to assume [believe] that it can fly unless there is reason to believe otherwise), and discuss more generally the types of reasoning people employ in deriving new beliefs on the basis of partial information. One aspect of their work which is relevant to this thesis is that (without claiming that people do the same) they apply numerical measures to parameters associated with reasoning in order to make trade-offs between positive and negative supports for beliefs.

5.6.3 Intentions, goals and commitments

Doyle (1988) distinguishes between different types of intentions: standing or singular, routine or problematic, constitutive or environmental. In his terminology, standing intentions are left in force and constantly obeyed until abandoned, whereas singular ones are one-off intentions which are abandoned as soon as they have been carried out. Routine intentions can be satisfied by doing a basic action, while problematic ones require further thought (or planning) to carry out. Constitutive intentions relate only to the agent's own structure, while environmental ones relate to the agent's environment, or the agent's relationship to its environment. Similarly, Cohen and Levesque (1990a) distinguish between goals of maintenance and goals of achievement, and \textsc{intend}_1 (intend to do immediately) and \textsc{intend}_2 (future-directed intentions). One distinction which is sometimes blurred in the literature is the distinction between 'intending to do' and 'intending to achieve'. There is less confusion over the status of a goal, which is generally regarded as relating to the state of the world (or an aspect of the state of the world).

Intentions, goals and commitments all relate to actions, or to states to be achieved. Bratman (1990) discusses the nature and role of intentions. He contrasts literature which
views practical reasoning as beginning with the agent's desires and beliefs and issuing in a decision, choice or action, providing no distinctive role for the agent's future-directed intentions as input to the reasoning, with work which reasons from intentions and beliefs to derive courses of action. These different perspectives are integrated in hierarchical plans such that the overall plan is developed in advance, but the details may be filled in later. This accommodates desires-beliefs reasoning when deliberating between admissible options. In particular, he argues that the main function of future-directed intention is to influence reasoning over the intervening time, particularly in relation to coordinating activity. For example, if I intend to attend a seminar in London on Thursday afternoon, I would be foolish to arrange to take my daughters swimming at 5pm the same day, because it would be impossible to do both. Intending to do something involves some measure of commitment to doing it.

Cohen and Levesque (1990a) develop a formalism which shows how intentions can be adopted, relative to the agent's existing intentions and beliefs. They define intention as 'choice with commitment', defining how an agent is committed to goals, and under what conditions the agent can drop goals. Their formalism is based on a 'possible worlds' semantics, and is expressed in a model theory with the primary operators BELief, GOAL, HAPPENS and DONE. The details of their formalism are not relevant in the current context, but it may be noted that it does not extend to defining how decisions (or choices) between alternative courses of action are made. However, one interesting definition which arises from their work is that of interlocking commitments. They define a 'persistent relativised goal' (P-R-GOAL \( x \ p \ q \)) as one such that agent \( x \) has as a GOAL to bring about a state of the world in which proposition \( p \) is true, as long as proposition \( q \) is true (but will drop this goal if \( p \) is achieved or is believed to be unachievable, or if the agent comes to believe (not \( q \))). Then agents \( x \) and \( y \) have interlocking commitments if (P-R-GOAL \( x \ p \) (GOAL \( y \ p \))) and (P-R-GOAL \( y \ p \) (GOAL \( x \ p \))). Thus each agent will retain the intention to achieve \( p \) as long as the other does also.

This definition may be compared with that of Power (1984), who defines mutual intention as a distribution of mental states such that:
...etc ad infinitum

where \( I \) means that agent \( x \) intends to achieve \( p \) and \( A \) means that agent \( y \) assumes that agent \( x \) intends to achieve \( p \). There are clear parallels between this definition and that of mutual belief presented above. In particular, it is possible that one agent believes there to be a mutual intention where such an intention does not in fact exist. One aspect of mutual intention which Power's formalism fails to capture (though Cohen and Levesque's does) is the dependency between the intentions - that \( x \)'s intending to achieve \( p \) is actually dependent on \( x \)'s assumption that \( y \) intends the same. This point is expressed clearly by Searle (1990). Searle goes on to argue that "we-intentions cannot be analysed into sets of I-intentions". Similarly, Grosz and Sidner (1990) analyse SharedPlans which cannot be decomposed into the individual plans of the participants.

One aspect of the discussion which is missing from most accounts of goals, intentions or commitments (individual or mutual) is the question of how goals are acquired and choices are made between alternatives.

5.6.4 Desires and values

Doyle distinguishes between deliberate action (governed by intentions) and wanton action (governed by desires), but does not discuss how the agent might become committed to the one rather than the other. Kiss takes a more 'integrated' approach in which the agent decides what to do based on its values, where a value is an attitude towards a state which is believed to be capable of generating pleasure.

However, just as little work in IES research has focused on motivational aspects of teaching so, in agent theory, little work to date has focused on affective attitudes.

5.7 Discussion

Doyle (1988) gives a good overview of many of the issues which need to be tackled
in order to define (and maybe also to build?) rational agents - for example, dealing with probabilities, learning, reflection and intentionality in rational ways. He also discusses the notion of a 'society of mind', as an approach to the difficult issue of dealing with inconsistencies in attitudes. He notes that having assumptions and learning (or belief revision) involve reflecting on one's beliefs; planning involves reflecting on one's intentions and priorities; deliberation involves reflecting on one's preferences and desires. Many of the issues dealt with by Doyle have hardly been touched on in this section, not because they are unimportant, but because addressing them is beyond the scope of the current research project.

Much of the existing work on agents has implicitly accorded the agent fairly routine responsibilities (such as those of a clerk or a mechanic). Clearly such a role imposes different requirements on the agent from those of a tutor. For example, such an agent would be largely reactive, responding to the human user (though not necessarily simply complying with a request; it might, for example, ask the user whether they are sure they want to do a particular action, if the agent has reason to think the user might be misguided in their request). In contrast, a computer tutor may well want to extend the agenda beyond that proposed by the human student.

The theoretical agent model developed in the next chapter has many of the attributes discussed in this chapter, in particular in sections 5.5 and 5.6. It is based on the action cycle definition of Kiss, though (like Kiss' own implementation) certain simplifications are made. As will become apparent, the agent model developed in this thesis is consistent with the outline procedure for deliberate action described by Doyle, and also takes a similar approach to Georgeff and Ingrand in defining action sequences at different levels of abstraction. The beliefs maintenance mechanism included in the implementation ($7$) is similar in structure to Doyle's TMS, and the topic control is based on the work of Grosz and Sidner. Most of the attitudes discussed in §5.6 (with the notable exception of intentions) are included in the model. The issue of mutual attitudes has also been discussed in this section; such mutual attitudes play a central role in the agent's ability to engage in collaborative activity as discussed in §6.6 and §7.6.
Chapter 6: The theoretical agent model

In this chapter, a general definition of a simple agent is presented. The qualities which this agent has are that it does not engage in advance planning and that it has fixed expertise in its domain of action (it cannot learn). The simple model will then be extended to a model of collaborative activity (§6.6), and applied to the target domain (the dialogue component of an IES) (§6.8). The prototype system which has been implemented based on this definition is presented in the next chapter (§7).

6.1 Introduction

In seeking to develop a dialogue component with the qualities of being able to participate in coherent, purposeful dialogue as outlined in §4.3.4, the most important question is: how does a dialogue participant decide what to say in any given context? Clearly, what the participant decides to say will depend on the preceding dialogue and on what the dialogue is meant to achieve for them. It will also depend on their current beliefs. In order to be able to participate in a mixed-initiative dialogue, the dialogue component of the system must be capable of generating utterances which it believes can satisfy its values, rather than either effectively having situation-utterance correspondences built in, as has happened in the past in IESs which take no account of the user's goals, or being merely reactive, as in learning environments which allow the user to retain total control of the interaction.

The system and a user are viewed as engaging in collaborative problem solving in reaching a decision together. This means that the two parties work together, discussing the aspect of the problem on which they are working, in contrast to the slightly weaker requirement that they should cooperate (in which case, one party might be getting on with sorting out one aspect of the problem while the other is engaged in another aspect). Within the problem solving domain, both system and user have the same rights and constraints in terms of what they can say and do, so there is a symmetry within the interaction. However,
both system and user are viewed as being autonomous agents, and they play different roles in the interaction, so in this respect there is no such symmetry.

In this chapter, a model is developed which defines how an agent can engage in an opportunistic interaction (with no planning beyond the current utterance). The agent is an expert in dialogue, in the sense that it has established beliefs about how a dialogue proceeds (in terms of sequences - for example, listening then responding, or hearing, then understanding, then revising beliefs) and it does not need to explicitly plan each step of the interaction. It is also an expert in the sense that it has established means-ends beliefs about what types of utterances can satisfy certain values (it does not have to learn these correspondences). The term 'expertise' is being used here in a limited sense. Winograd and Flores (1986) would term such an agent an idiot savant, in that it lacks any sort of 'common sense', but the use of the term 'expertise' is consistent with common practice - as for example in work on 'expert systems'.

Taking Kiss's notion (§5.4) that an agent is capable of goal-directed action through which it autonomously pursues its interests, together with the notion that an agent has goals at different levels such that the achievement of lower-level goals (sub-goals) contributes to the achievement of higher-level ones, the model which has been developed defines an agent which is capable of deciding between alternative courses of action when appropriate, and of achieving defined sequences of goals (or schemata). The way in which it does this is described below (§6.4). Before the means by which the agent acquires goals and commitments is described in detail, the attitudes included in the model will be defined. As will be illustrated, the model can be applied to agents which have expertise in any activity, individual or collaborative, which can be characterised as opportunistic. In particular, in this thesis the generation of language is viewed as opportunistic rational action (§6.8).

Kiss (personal communication) has suggested that an important function of agent research is to develop theoretical models to describe human activity. This motivation is made explicit in some work, such as (Cohen and Levesque 1990b). In this thesis, no claims of psychological validity are being made; the model presented in this chapter defines the activity
of an artificial agent. Discussions involving the application of the theoretical model in domains generally associated with human activity, such as protesting and dancing, are presented to illustrate how the model generalises to domains other than that of dialogue, and are not intended to imply that people operate in this way. However, in defining the theoretical model, one of the criteria on which design decisions have been based has been that of maximising psychological plausibility. This is particularly pertinent to the discussion in §6.2.3.

6.2 The attitudes included in the model

As yet, with the possible exception of beliefs, no generally agreed technical definition of the terminology associated with agent attitudes has emerged. Therefore, the terms used to label the attitudes included in the model are described here. While it is believed that these definitions match reasonably well with folk psychological understandings of the terms, they may not correspond precisely with the ways in which other researchers use the same terminology. Until a common understanding of the meanings of terms emerges, the only reasonable course is to define meanings and hope that the definitions do not conflict with intuitions.

6.2.1 Beliefs

Beliefs are the only cognitive attitude encoded in the model. Like Cohen and Levesque (1985), two categories of beliefs are distinguished: standard beliefs which are held by the agent, and a type of mutual beliefs. The type of mutual beliefs encoded in the model is referred to as 'mutual working beliefs'. These are similar to the mutual beliefs discussed in §5.6.2, but cannot be expressed in terms of recursive sets of individual beliefs. Mutual working beliefs do not entail both parties necessarily holding the belief as long as they agree to work with it for the moment. This slightly weaker requirement is needed in the current model because belief revision has not been incorporated in the agent design; since the agent cannot change its beliefs, it has to be able to accept working with beliefs it does not hold in a domain where agents are likely to hold different beliefs.
Holding a belief entails a commitment to the consequences of holding it (Winograd and Flores 1986). Similarly, in collaborative problem solving the partners holding a mutual working belief entails them having a commitment to the consequences in terms of their collaborative activity.

The means by which mutual working beliefs are established (through negotiation) is discussed in §7.6.1.

6.2.2 Goals, wants and commitments

The conative attitudes encoded in the model (defined by Kiss as those which the agent needs to be able to act) are wants, commitments and goals. It should be noted, however, that the entities encoded as goals in this model correspond to Kiss' wants; Kiss' goals are simply states of the world (which the agent may want to achieve), and as such are non-attitudinal. In the model presented here, goals refer to an aspect of the state of the world, while wants and commitments refer to actions. For example, if Bert is considering "going to the cinema", the appropriate goal is a state of the world in which Bert is in the cinema. The corresponding wants and commitments refer to the action of getting there, whether by car, by bike or on foot. Wants refer to all the known ways of achieving the goal; commitment refers to the chosen way of achieving it. Thus in this example, the agent wants to go by car, wants to cycle, and wants to walk to the cinema, and will commit to one of these alternatives.

A clear division between actions and achieved states, for example 'intending to do' and 'intending to achieve', is essential to the definition of the agent model. There are actions which may be goals - for example, I may have a goal of being in a state where the action I am performing is dancing - but the distinction must be maintained. The relationships between goals, wants and commitments is discussed further in §6.3.

Many researchers in the field of rational agents focus their attention on intentions, as discussed in §5.6.3. Given that the agent is assumed to be an expert in its domain of action,
it has not been found necessary to include a formal representation of present-directed intentions, since as soon as the system has decided to intend to do something it has also become committed to doing it. See §6.3.

Neither has it been found necessary to include a representation of future-directed intentions, where intentions relate to actions, as future-directed intentions are intimately bound up with planning issues (not being considered in the model). It has, however, been found necessary to include a representation of future-directed, or transient, goals \((t\_goals)\). A transient goal is one which the agent will address at some future time, or which the agent believes may take several attempts before it is achieved. A transient goal is set up at the point at which the system recognises the need to address such a goal, together with a relevance function to indicate when it will become relevant to try to address that goal. When it is relevant to consider the goal, it may be re-scheduled or abandoned. Alternatively, at that point, the system decides between alternative actions which achieve progress towards that (transient) goal.

As an example of the distinction between transient goals and future-directed intentions (assumed to relate to actions), suppose that Cleo notices that her favourite jacket has a hole in it. She may form an intention to go to the Bargain Basement Clothing Company shop to buy a new jacket on Saturday, or she may simply adopt a transient goal to be in the state where she possesses a new jacket. This goal will become relevant whenever Cleo is near a shop which sells jackets, and will be satisfied when Cleo finds and buys a jacket she likes or when she is given one as a present, or may be dropped if Cleo decides that she cannot find a jacket to suit her taste and finances.

6.2.3 Values and preferences

The affective attitudes encoded in the model are values, and these are isolated within the preference mechanism, in which a choice is made between alternative possible actions. The corresponding rationality constraints are currently implicitly encoded (i.e. values are encoded and manipulated without reference back to the pleasure which might be derived
therefrom). For a tutoring agent, the main values are hedonic (relating directly to the agent's pleasure) and pedagogical (concerned with aspects of teaching well, and therefore only indirectly contributing to the agent's pleasure).

As an example of the distinction between goals, transient goals and values, suppose now that Cleo has values relating to comfort, to being entertained and to being informed (i.e. she values situations in which she expects to be comfortable, entertained and informed). The values of being entertained and informed have caused her to decide to buy an introductory text on general relativity, and all three values have caused her to decide to sit out in the sunshine to read a chapter of it. At this point, she has a goal of reaching a state of the world where she has read (and understood) the chapter, and all her activity is directed towards achieving the goal. She has a transient goal of having read the whole book. At times, her activity will be directed towards achieving this, but such activity will be interspersed with other activity which has nothing to do with the achievement of this (transient) goal. Also, there will be a time at which either the goal has been achieved (the text has been read and understood) or has been abandoned (she decides to give up because the book is less entertaining or informative than expected). In contrast, values cannot be achieved or abandoned in this way; there will never be a time at which Cleo is perfectly comfortable, informed or entertained.

Following Kiss and Brayshaw (1989), the values incorporated in the model are assigned numerical weights to reflect their relative importance to the system. The use of numerical weights may be viewed as an undesirable way to represent relative importances, but it permits relatively speedy and accountable decision making to be performed, and it will most often produce a unique decision outcome. Unlike the implementation of Kiss and Brayshaw, values are not represented in a tree hierarchy, because values are constant. Kiss and Brayshaw use a tree hierarchy to modify values in a consistent way during the agent's lifetime to reflect changes in the environment, and use relevance functions to determine when a particular value is relevant; in the model presented here, relevance functions note relevant aspects of the agent's environment without having recourse to adjusting values (see
§6.4). Relevance functions are used more widely in this model than in that of Kiss and Brayshaw, obviating the need for modification of values.

Values are located in the preference mechanism, and are therefore not accessible to the agent itself. They are only used for deciding between alternative possible actions. Located in the same place are means-ends beliefs about what values actions satisfy, and under what circumstances (i.e. when which aspect of the world state is true). For example, while resolving a conflict the action of giving a reason for believing not-X (i.e. the opposite of the belief, X, proposed by the user) may satisfy the value of developing the user's understanding of the problem, but only in the situation where the system has an appropriate belief which it has not proposed already. A consequence of locating values and means-ends beliefs within the preference mechanism is that the agent is not aware of these attitudes. For example, it cannot discuss them with a user. The main reason for making this decision relates to the issue of whether or not it is reasonable/desirable for the agent to be aware of numbers attached to its values; while no claims of psychological validity are being made in this research, it is clearly not psychologically plausible to consider that an agent is aware of numerical weightings attached to its values, whether or not it is explicitly aware of those values. How, and under what circumstances, an agent becomes aware of its values and of their (qualitative) relative importance is an open research question, so the decision to 'hide' the values in the current model may be viewed as somewhat ad hoc. It would be a trivial change to make the values (along with their numerical weights) and means-ends beliefs accessible to the agent; further research is needed, however, before the issue of awareness and partial ordering of values can be dealt with in a principled way.

6.3 The action cycle and goal-action trees

6.3.1 The action cycle

The core of the theoretical agent model is an action cycle which operates on a goal-action tree in order to make decisions and become committed to action. An agent's activity involves many iterations of the action cycle while traversing the goal-action tree.
The action cycle around which the model is constructed has been developed to accommodate many ideas from the literature as well as the author's intuitions, and to provide a theoretical framework for integrating decision points with pre-compiled sequences of goals to be reached. In descriptive terms, the stages of the action cycle are as follows.

1) If you have a goal which is relevant in the current context, and you believe that doing X goes some way towards achieving that goal, then you want to do X.

2) Of all the possible actions that you want to do, which you are not committed to already and for which all prerequisites are satisfied, commit yourself to the one you prefer. (This satisfies the rationality constraint on conative attitudes as discussed in §5.6.1.)

3) If you are committed to doing something which is not a basic action (i.e. has sub-parts) then adopt the sub-parts as goals.

4) If you are committed to doing a basic action then do it and consider it done.

5) If the action is one which causes the 'parent' goal to be reached then consider the 'parent' goal to be reached.

6) If you have reached the goals corresponding to all the sub-parts of an action then consider the action done.
7) Once you have committed to a new action, forget about having done previous actions.
8) Once you have done something, cease to be committed to it.
9) Once you have committed yourself to an action, forget that you wanted it. Also forget that you ever wanted all the non-preferred alternative courses of action.
10) If it is relevant to consider a transient goal right now, then review it. This involves deciding whether to abandon, re-schedule or address the transient goal.

A first attempt at expressing this action cycle formally is included in §B.1.

6.3.2 The goal-action tree

This action cycle defines a series of choices and actions, which would typically be presented in the form of a flow diagram such as Fig. 6.2. In this diagram, goals are shown in circles; addressing a goal involves choosing between possible actions (shown in rectangles) and performing the chosen action. In the case of complex actions (of which a1 is the only example in this case), this necessitates achieving defined sub-goals in a defined order. The flow diagram is a suitable form of representation for very simple cases, but becomes rather unwieldy for cases where there are a large number of possible goals and actions.

In this thesis the form of representation which is used to indicate the relationships between goals and actions is a tree structure. With this representation, the action cycle may be viewed as a tree traversal mechanism. In this tree structure, actions (or ongoing activities), between which a choice is made, are represented as disjunctives and goals (or states to be achieved) as conjunctives. Two modifications are made to a standard and-or tree. The first is that a set sequence is imposed on the 'and's; only one goal is relevant at a time, and the system cannot explicitly choose which goal to address next. The second is that (with the exception of goal-reaching actions) 'or's are not mutually exclusive in the sense that, although there is no parallel activity, the system can decide to traverse one branch and later traverse the same or a different branch at the same decision point. For every goal, there is at least one goal-reaching action (often a null action which represents the decision that the
parent goal has been reached). The conceptual distinction (as opposed to the technical distinction) between goals and actions might sometimes seem blurred; one distinguishing feature is that, with the exception of goals which correspond to goal-reaching actions, the agent decides, through the preference mechanism, when a goal has been reached or should be dropped, whereas commitments are fulfilled through the reaching of sub-goals, and the agent cannot explicitly decide to drop a commitment (except through deciding to drop the sub-goals). The example presented as a flow diagram in Fig. 6.2 is presented as a tree structure in Fig. 6.3.

![Fig. 6.2: example flow diagram](image)

In this example, the top-level goal is g1. This goal is addressed by performing actions a1 and a2 any number of times, and is considered reached when action a3 has been performed. Actions a2, a3, a4, a5, a6 and a7 are basic actions (which the agent can execute directly). Action a1 involves reaching goals g2 and g3. Goal g2 is reached by performing action a4, and g3 by performing a5 any number of times, followed by either a6 or a7. So sequences of basic actions performed by the agent in addressing goal g1 might include:

1) a3.
2) a2, a4, a5, a7, a2, a2, a3.
3) a4, a6, a4, a5, a7, a2, a4, a5, a5, a6, a3.

The choice of the sequence of actions to be performed is governed by the decision mechanism described next (§6.4).

Fig. 6.3: example goal-action tree

Goals are written in *italic* type and actions in normal. Goal-reaching actions are *underlined*.

6.4 The decision mechanism

The preference mechanism is based on Multi-Attribute Utility Theory, referred to earlier as the Weighted Objectives Method (§2.2.3). For any goal which can be addressed by more than one action, there is a list of the possible actions. For each possible action, there is a list of means-ends beliefs about what values that action satisfies, and how well (as measured by a numerical strength) and under what conditions. Each value has a numerical weight attached to reflect its relative importance to the system.

The score assigned to any action is the sum of the product of the strengths and the weights assigned to all relevant values. For example, if action A satisfies values

- $v_1$ with strength $s_1$ if relevance function $r_1$ returns true,
- $v_2$ with strength $s_2$ if $r_2$,
- $v_3$ with strength $s_3$ if $r_3$, and
- $v_4$ with strength $s_4$ if $r_4$,

and value
v1 has numerical weight n1,
v2 has weight n2,
v3 has n3 and
v4 has n4,

then if r1 and r4 return true (r2 and r3 false), the score of A in this situation will be (n1*s1+n4*s4). The preferred action (to which a commitment is then made) is that with the highest score. Specific examples of the ways the preference mechanism operates at different decision points are given in §6.5.2.

6.5 Illustrations of the operation of the action cycle

6.5.1 Illustration of tree traversal: the goal of having a pleasant evening

To illustrate how this action cycle works in practice, let us take as an example the goal of having a pleasant evening. Possible activities might include going to the pub, seeing a film, going out for a meal or going to a health club (see Fig. 6.4).

An evening's entertainment may combine more than one of these alternatives. One might first decide (through the preference mechanism) to go to the pub; in this case, one would select a pub, go there, and spend time there. While there, one might decide to buy a pint (and drink it!), then buy another, then buy some crisps, then decide to leave. At this point, the agent is back at the top-level decision point, and the alternatives available are to go to (another) pub, to go to the cinema, to go for a meal, to go to the health club or to finish (presumably, go home to bed), at which point the goal of having a pleasant evening is considered reached. This example highlights some limitations of the model; for example, it does not cater for parallel activity (like having crisps with one's pint), and does not accommodate future-directed intentions (like agreeing to have a drink before going to the cinema). However, it illustrates how the basic tree traversal works.
6.5.2 Illustration of decision making: meeting a friend in London

The decision making process may be illustrated using the example of making a journey. An agent has arranged to meet a friend in Covent Garden, and has to decide how to get there. As illustrated in Fig. 6.5, the decision is taken in several stages.
Fig. 6.5: meeting in London goal-action tree

meet Fred in Covent Garden

travel by bus

travel by train

travel by car

get train to London

get to station

get to Covent Garden

walk
cycle
drive

go by taxi

go by bus

walk

walk to platform
catch train

leave train

get to rendezvous

get off train

walk to rendezvous

walk

go by underground

Goals are written in italic type, and actions in normal. Goal-reaching actions are underlined.

The means-ends beliefs activated at each decision point are based on past experience (i.e. on expertise). Assume that the agent's values are as follows:

<table>
<thead>
<tr>
<th>value</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimise cost</td>
<td>10</td>
</tr>
<tr>
<td>care for the environment</td>
<td>6</td>
</tr>
<tr>
<td>minimise travel time</td>
<td>8</td>
</tr>
<tr>
<td>minimise stress</td>
<td>9</td>
</tr>
<tr>
<td>maximise comfort</td>
<td>20</td>
</tr>
<tr>
<td>maximise convenience</td>
<td>4</td>
</tr>
<tr>
<td>maximise safety</td>
<td>15</td>
</tr>
</tbody>
</table>
For the first decision - whether to go by car, bus or train - the agent's means-ends beliefs are as follows:

<table>
<thead>
<tr>
<th>value</th>
<th>car</th>
<th>train</th>
<th>bus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>strength</td>
<td>relevance</td>
<td>strength</td>
</tr>
<tr>
<td>cost</td>
<td>5</td>
<td>always</td>
<td>3</td>
</tr>
<tr>
<td>environ.</td>
<td>0</td>
<td>always</td>
<td>10</td>
</tr>
<tr>
<td>time</td>
<td>4</td>
<td>always</td>
<td>8</td>
</tr>
<tr>
<td>stress</td>
<td>2</td>
<td>always</td>
<td>9</td>
</tr>
<tr>
<td>comfort</td>
<td>10</td>
<td>if wet</td>
<td>6</td>
</tr>
<tr>
<td>conv.</td>
<td>8</td>
<td>always</td>
<td>3</td>
</tr>
<tr>
<td>safety</td>
<td>9</td>
<td>if late</td>
<td>1</td>
</tr>
</tbody>
</table>

In this chart, it is assumed that the value of maximising comfort does not distinguish between the alternatives unless it is raining, and that of maximising safety does not distinguish unless the agent (who is of a nervous disposition) is likely to be returning home late at night.

If it is dry, and the agent is not coming home late, the utility of travelling by car is 
\[ (5 \times 10 + 0 \times 6 + 4 \times 8 + 2 \times 9 + 8 \times 4) = 132. \]
If it is dry but the agent is coming home late, the utility rises to 
\[ (132 + 9 \times 15) = 267. \]
If wet but not late, the utility is 
\[ (132 + 10 \times 20) = 332, \]
and if wet and late 
\[ (332 + 135) = 467. \]

The figures for all three modes of transport (with the best alternative under each set of conditions in bold face) are:

<table>
<thead>
<tr>
<th>conditions</th>
<th>car</th>
<th>train</th>
<th>bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>dry, not late</td>
<td>132</td>
<td>247</td>
<td>214</td>
</tr>
<tr>
<td>dry, late</td>
<td>267</td>
<td>262</td>
<td>274</td>
</tr>
<tr>
<td>wet, not late</td>
<td>332</td>
<td>367</td>
<td>314</td>
</tr>
<tr>
<td>wet, late</td>
<td><strong>467</strong></td>
<td>382</td>
<td>374</td>
</tr>
</tbody>
</table>
This chart shows that for this simplified case, the decision which would be taken would be to travel by train as long as the agent was not expecting to return home late, to travel by bus if it was dry but the agent was expecting to be home late, and to take the car if it was raining and the agent was expecting to be home late.

Imagine that the agent is not expecting to return late, so the decision has been taken to travel by train. Then for the next decision - whether to get to the station by car, by bicycle or on foot - the means-ends beliefs are as follows:

<table>
<thead>
<tr>
<th>value</th>
<th>car strength</th>
<th>car relevance</th>
<th>bicycle strength</th>
<th>bicycle relevance</th>
<th>walk strength</th>
<th>walk relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>cost</td>
<td>0</td>
<td>always</td>
<td>10</td>
<td>always</td>
<td>10</td>
<td>always</td>
</tr>
<tr>
<td>environ.</td>
<td>0</td>
<td>always</td>
<td>10</td>
<td>always</td>
<td>10</td>
<td>always</td>
</tr>
<tr>
<td>time</td>
<td>8</td>
<td>always</td>
<td>7</td>
<td>always</td>
<td>1</td>
<td>always</td>
</tr>
<tr>
<td>comfort</td>
<td>10</td>
<td>if wet</td>
<td>0</td>
<td>if wet</td>
<td>0</td>
<td>if wet</td>
</tr>
<tr>
<td>safety</td>
<td>8</td>
<td>if late</td>
<td>5</td>
<td>if late</td>
<td>0</td>
<td>if late</td>
</tr>
</tbody>
</table>

This decision point will only be reached if the agent is not going to be home late, so the value on safety is not relevant. Using the same method of calculating utilities as above, the utilities are:

<table>
<thead>
<tr>
<th>conditions</th>
<th>car</th>
<th>bicycle</th>
<th>walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>dry</td>
<td>64</td>
<td>216</td>
<td>168</td>
</tr>
<tr>
<td>wet</td>
<td>264</td>
<td>216</td>
<td>168</td>
</tr>
</tbody>
</table>

So if it is wet the agent will drive to the station, and if dry the agent will cycle. One could imagine situations in which the agent might choose to walk, for example when the bicycle has a puncture, but as defined this agent would never make that decision. It would clearly be a trivial extension to permit the agent to take account of the usability of the bicycle, but the agent definition does not allow the agent to react to unforeseen circumstances; the agent exists in a 'closed world'.
Further decisions are made by the agent in the same way as they arise. So, for example the decision about how to get to Covent Garden from the London terminus would be taken upon arrival at that terminus.

Some of the values, or weights assigned to them, and some of the means-ends beliefs used in this example may appear somewhat unrealistic if viewed as a model of human decision making. They are not intended as such a model; they are intended to present a fairly simple example of the way the agent as defined reaches different decisions based on its beliefs about the state of the world.

The same decision mechanism can be used to take account of the state of the agent. For example, if the agent is tired then the utility of driving to the station might increase.

6.6 Collaborative activity

Although both of the illustrations presented above implicitly involve other agents, both describe individual activity. The model may be extended as follows to define collaborative activity. In collaborative activity, any agent is simultaneously traversing two goal-action trees, which may be viewed as being orthogonal. The traversal mechanism for the individual activity of each agent is as defined above. Each agent involved in the collaborative activity may have a different individual goal-action tree, or may be engaged in a different activity at any given moment, but all agents are at the same point on the (orthogonal) collaborative tree. All agents have identical collaborative trees, defined either by a controlling agent (which collaborating agents must accept) or developed through a process of negotiation. Traversal of the collaborative tree is coordinated either by negotiation or by one agent being 'in command' (however temporarily; if there is not a defined commander, then one of the decisions which an individual can take is to try to take command), and not by the use of the decision mechanism.

In this thesis, the collaborative tree is defined by the computer-based agent, and traversal of the tree is coordinated by negotiation. Among the conclusions of the empirical
test of the prototype system (§8) are that this definition of the collaborative tree is unsatisfactory, and that further work is needed on enabling participants to construct this tree jointly. Also, negotiation as implemented consists only of proposals, supporting justifications and counter-arguments; it does not extend to trade-offs and compromises as discussed, for example, by Sycara (1989) or Zlotkin and Rosenschein (1990a, 1990b).

6.7 Illustrations of collaborative activity

6.7.1 Illustration: a protest rally

Fig. 6.6: individual activity in protest march

The government has just announced a decision with which an agent strongly disagrees. The agent can think of several ways of expressing their disagreement, including writing to the local M.P., sending a letter to a national newspaper, and joining a protest march (for simplicity, it is assumed here that one is being organised). In deciding to join the protest march, the agent becomes committed to engaging in collaborative activity, and through that commitment starts to traverse the collaborative goal-action tree which defines the organised march as well as their own individual goal-action tree.
A possible goal-action tree for one protester is shown in Fig. 6.6, and a goal-action tree for the collaborative activity of holding a protest march is shown in Fig. 6.7. Clearly, in this case the individual goal-action tree for the collaborative activity (i.e. ignoring the activities like writing to the M.P.) and the collaborative tree have very similar structures, though the collaborative tree is the simpler.

![Collaborative activity in protest march](image)

In this example, the activity is assumed to be coordinated by the leader or organiser, and the individual agent is assumed to have values (such as acting in accordance with their commitment to collaborate) which influence the agent's individual activity in such a way as to fit in with the coordinated group activity.

### 6.7.2 Illustration: country dancing

In country dancing, many people are involved in the dance, including dancers, musicians and often a caller who shouts out instructions for the dancers to follow. Different dancers execute different steps depending on their position in the dance (e.g. whether they are members of the 'top couple' or not, whether they are taking the roles of men or women), so as individual agents each dancer is doing something different, but together they are 'performing a hay' or 'poussetting'. The musicians are also agents collaborating in the performance of the dance, as is the caller, who is the agent with the role of coordinating the activity.

This illustration and the previous one are fairly sketchy, and are intended to demonstrate the applicability of the model to a variety of (closed) domains. In the next
section, the application of the theoretical model to the target domain is presented in much greater detail.

6.8 The dialogue agent

In the development of a dialogue agent, it is being assumed that the agent and user are engaging in collaborative problem solving, and that the activity is coordinated by a simple turn-taking mechanism (such that each agent is autonomous while it is their turn, and then they wait - or listen - while the other agent has their turn).

The computer-based agent has values which are relevant to its role as teacher. These values include values relating to the user externalising and reflecting on their beliefs, motivational values such as keeping the interaction varied and ensuring that the user does not get stuck, and hedonic values (i.e. values relating to pleasure) such as making as little effort as possible. These values, together with the ability to perform particular dialogue actions and means-ends beliefs about which values any given action satisfies, define the teaching style of the system. This is discussed at greater length in §7.2.4.

Given the agent's role as teacher, the first two decision points (see Fig. 6.9) comprise trivial decisions. At the first, while the agent could in principle choose to ignore a user, its pedagogical values dictate that as long as a user exists, it will choose to interact with them. When no such user exists, it will do nothing. In this instance, neither of the possible actions is a goal-reaching action, so while there is no user the agent reviews this decision repeatedly. When there is a user the system chooses to interact. The second decision point (whether to engage in collaborative problem solving or to refuse collaboration) is equally pre-determined, but is included in this discussion to highlight the point at which the commitment to collaborate is entered into.

The turn-taking mechanism defines the decision making structure of the goal-action tree at the next level, in terms of an utterance cycle (Fig. 6.8), in which the agent may choose to finish the dialogue when it has either finished speaking or finished listening. In
the goal-action tree this appears as a three-way decision (listen, respond or finish), but only two of these actions will be relevant at any one time, depending on whose turn it is.

![Fig. 6.8: the utterance cycle](image)

The action of listening involves reaching several sub-goals. First, the system has to wait for the user to indicate that they have finished their utterance, then it has to break the utterance down into individual propositions before dealing with each one. The details of how the agent processes an individual sentence are omitted from Fig. 6.8, but will include understanding the sentence and making inferences from aspects of the sentence such as topic shifts, establishing a view regarding the proposition (does the agent agree with the proposition or not, or was the user asking a question?), and noting any transient goal which it might wish to address later (such as resolving a detected disagreement). Once it has processed all the sentences in the utterance, the agent will draw additional inferences based on what the user did not say (for example, if the system had made a proposition earlier, which the user has accepted without comment, then the system might assume that the user agrees with it). Finally, the system will note that it is now its turn to speak.
This is the point at which any outstanding transient goals become relevant. In principle, before the system decides how to address a new goal, it has a quick review of all the relevant transient goals, decides which to retain or abandon, and which to address now.
Any which it decides to address now are inserted as new goals at the top of the goal stack, and dealt with in the same way as any other goal. As all the possible transient goals in this model involve the system responding appropriately to something which the user said, all the transient goals become relevant as soon as the agent has become committed to responding, and are sorted out and addressed then. The action of responding involves reviewing the current collaborative activity, with a view to either continuing or suggesting a change. The system then decides what else (if anything) it wishes to say as part of the current response, makes the utterance, and establishes expectations - that the user will now say something, that the user will answer any questions, or react to any propositions, etc.

![Fig. 6.10: collaborative problem solving goal-action tree](image-url)
The collaborative problem solving tree is shown in Fig. 6.10. This is based in part on the discussion on decision processes in §4.2.8. This tree has very little structure because the user is not in fact assumed to be a problem solving expert, and is permitted to experiment and try out different problem solving strategies. Conceptually, this collaborative tree is a 'first pass', and the dialogue participants should be able to collaboratively construct their own problem solving goal-action tree, which would include sub-goals and groupings of actions. This is discussed in the section on further work (§9.5).

It is not being claimed that the human participant in the dialogue has a tree like either of the above in their head - or indeed a tree traversal mechanism such as the action cycle defined above (Fig. 6.1). The theoretical agent model together with the dialogue goal-action tree and collaborative problem solving goal-action tree define a mechanism by which a computer-based agent can participate in dialogue and engage in collaborative problem solving.

6.9 Discussion

In this chapter, a theoretical model for an agent which has expertise in closed domains which can be characterised as opportunistic has been defined and applied to the domain of engaging in dialogue. The model has obvious limitations such as the fact that, as currently defined, it cannot accommodate planning or learning.

The extension of the model to define collaborative activity serves, among other things, to clarify the distinction made in §6.1 between cooperative and collaborative activity. In a model of cooperative activity, it would be possible to construct a goal-action tree for the joint activity, but the participating agents might be at different points within that tree structure at the same time. In a model of collaborative activity, the agents are at the same place in the tree (apart from minor deviations due to lack of synchronization). The model as presented fails to give an adequate account of how mutual commitments are established but, as will be shown in the following chapters, it provides a suitable basis on which to build (in terms of both theory and practical implementation).
Chapter 7: WOMBAT Implementation

The purpose of this chapter is to describe the implementation of WOMBAT. It is based on the theoretical agent model (§6) and the outline IES design (§4.3). The early sections in this chapter describe the implementation of the dialogue component, starting with a comparison between the implementation and the theoretical design presented in §6.8 (the core as shown in Fig. 7.1). Aspects of the implementation which do not feature in the theoretical discussion, such as topic control and plausible reasoning (the periphery in Fig. 7.1), are described in the middle sections (§7.3 - §7.5). Much of this discussion is based around a description of the activities involved in listening and responding, which in turn are described in more detail here than in §6.8.

![Fig. 7.1: the core and the periphery](image)

The learning environment is outlined in §7.7, and the use of the system is described in §7.8. In particular, some screen displays taken from an interaction are included in §7.8.3, to give a sense of how an interaction proceeds. In this chapter, all examples of interactions are anglicised versions of real interactions between the WOMBAT prototype and a user. (Interactions are anglicised because, although self-consistent, the system's formalised language is not particularly easy to understand.) An interaction is presented in formalised language (with translation) in §8.3.2, and several traces of interactions (without translation)
are included in §C.3.

7.1 Overview of the implementation

The implementation of the dialogue agent in WOMBAT is at the same time more simple (in terms of making some simplifying assumptions) and more detailed than the model presented in §6.8. The WOMBAT prototype consists of both a dialogue component and a learning environment, as discussed in §4.3. It has been implemented in Allegro™ Common Lisp on an Apple Macintosh™ SE/30 computer.

This account is not intended to include a full description of the implementation, but to describe the most important aspects of it and to highlight shortcomings and possible further developments. Inevitably, any discussion of an implementation written in Lisp refers to Lisp lists, but such references are kept to a minimum in this chapter. More technical details can be found in Appendix B.

For comprehensibility, examples taken from interactions are presented in natural language in this chapter, although this is not how they appear on the screen, or how they are represented internally by the agent. Internally, every sentence begins with either 'impart' (for statements) or 'elicit' (for questions). The remainder of the sentence is a proposition as discussed in §7.3.1. During an interaction, sentences appear on the screen in slightly anglicised (but not full NL) form. For example, the sentence which is encoded internally by the agent as 'impart believes objectives safety' appears as 'I believe an objective is safety'. For more complex propositions, the utterance as displayed on the screen is less easy to understand, and requires the user to have a good understanding of the internal representation. The user constructs sentences by selecting phrases such as 'I believe' and 'an objective is', and then adding free-form input such as 'safety' or 'comfort'. The operation of the interface is described later (§7.8). At this stage the point to be noted is that examples presented in natural language are encoded in the system as 'impart' or 'elicit' followed by a proposition.

The core of the implementation is the action cycle, which traverses the dialogue goal-
action tree. All the support functions which the agent needs in order to operate in the context of an IES are implemented as separate modules which are accessible by the core agent. So, for example, plausible reasoning is a 'black box' module into which the agent can insert a belief for assessment, or from which the agent can extract a belief on a specified topic. The preference mechanism is similarly a 'black box' into which the agent inserts all-possible-actions, and from which one is returned. It should be noted that these modules have access to all the information in the agent state, and can use this in their deliberations.

7.2 The implementation of the dialogue agent in relation to the theoretical model

In the current implementation, the agent only exists for the duration of the interaction. It is initialised by the user specifying a problem definition, and ceases to exist when the interaction terminates.

When the agent is initialised, the menu-bar is changed to that required for the WOMBAT learning environment (see §7.8.3) and the action cycle starts operating.

7.2.1 Implementation of the action cycle

The core of the action cycle is a direct implementation of the formal definition as described in §6.3, with two modifications; instead of having a separate encoding of when an action is a goal-reaching action, it is assumed that all basic actions are goal-reaching, and all others are not. At the time that this assumption was made, it seemed to be a reasonable one. As will emerge in the discussion about transient goals, however, it has necessitated the inclusion of some very powerful basic actions which would ideally be expressed as sequences of lower-level actions. It has also necessitated the inclusion of a single sub-goal and action branch (to make the action do_new_action non-basic) at one point (see Fig. 7.5).

All the transient goals, or t_goals, incorporated in the current implementation become relevant at the point where the agent has committed to respond. Therefore the second
modification made to the theoretical action cycle is that the review procedure for t_goals (the last line in Fig. 6.1) has not been implemented as part of the action cycle. Instead, goals to sort, select and address t_goals have been included in the goal-action tree, so that t_goals are reviewed once in each utterance cycle (see §7.5.1). Since in the implementation t_goals do not have associated relevance functions, they can never be re-scheduled explicitly, but might be postponed to be considered again on the next utterance cycle.

With these two exceptions, the action cycle is as presented in §6.3. The operation of the action cycle is governed by a central controlling routine, Agent_do. Its operation can be expressed as follows:

• if there is a relevant goal outstanding, then establish wants for all actions which might address that goal, and commit to the preferred action;
• if there are no outstanding goals then reset the menu-bar and finish;
• if committed to a basic act then do it and tidy up after it;
• if an act has just been done, then tidy up after it;
• otherwise, establish new goals, and from them wants and a commitment.

In this context, tidying up consists of updating the state of the agent, noting the act which has been done in the list of done acts, checking whether the doing of the act has resulted in any goals being reached, and removing the act from the list of commitments. For non-basic acts, this also involves checking whether all the sub-goals of an act have been achieved, and if so adding the act to the list of done acts and removing the sub-goals from the list of reached goals. T_goals (which do not contribute to the performance of any higher level act), are simply removed from the list of reached goals. Any goals which are in the list of reached goals are removed from the list of goals.

The principal routines called by agent_do are as follows. Firstly, agent_commit, calls the preference mechanism to select which act to commit to based on the current wants (and notes the commitment in the list of commitments). Agent_goals generates new goals from non-basic acts by referring to the data in dialogue_tree_get (which defines the dialogue goal-action tree) to establish what the sub-goals of the non-basic act at the top of
the list of commitments are, and notes these goals in the list of goals. Agent\_wants generates a list of all actions reasonable in the current context, by considering what goals are currently relevant and referring to the data in dialogue\_tree\_get to establish what actions can make progress towards those goals. These possible actions are listed as wants. It should be noted that goals are currently constrained such that only one can be relevant at a time (so if the system is seeking to satisfy multiple goals, the order in which they are to be satisfied is pre-defined). It would be possible to relax this constraint if there were a mechanism for the system to decide which goal to address next.

**7.2.2 Implementation of the preference mechanism**

The preference mechanism as currently implemented is very simple. For any goal which can be addressed by more than one action, there is a list of the possible actions. For each possible action there is a list of means-ends beliefs about what values that action satisfies, and under what conditions (i.e. when which relevance function returns true). There is a separate list of all the values with a numerical weight attached to each value to reflect its relative importance to the system. And finally, there are a large number of relevance functions which return true or false depending on various aspects of the agent state. Unlike the theoretical description presented in §6.4, there are no measures of the degree to which the action satisfies the value - i.e. all strengths are implicitly given a value of 1. So to repeat the example presented in §6.4:

If action A satisfies values

- $v_1$ if relevance function $r_1$ returns true,
- $v_2$ if $r_2$,
- $v_3$ if $r_3$, and
- $v_4$ if $r_4$,

and value

- $v_1$ has numerical weight $n_1$,
- $v_2$ has weight $n_2$,
- $v_3$ has $n_3$ and
Design, Decisions and Dialogue

Chapter 7: WOMBAT Implementation

v4 has n4,
then if r1 and r4 return true (r2 and r3 false), the score of A in this situation will be (n1+n4).
Again, the preferred action (to which a commitment is then made) is that with the highest
score. Specific example of the ways the preference mechanism operates at different decision
points are given in the next section and in §B.4.

7.2.3 The dialogue goal-action tree

The dialogue goal-action tree is stored in WOMBAT as a data file,
dialogue_tree_get, which encodes unchangeable beliefs (knowledge or expertise) about
how to conduct a dialogue. It includes three lists: the first is of what actions achieve
progress towards any given goal. For example, the sub-list (survive teach_user
do_nothing) encodes the belief that the only ways to satisfy the goal of survival are to
either teach the user or do nothing. The second is of what goals are the sub-parts of an
action. For example, the sub-list (teach_user process_pd make_decision) encodes the
belief that the action of teaching the user involves reaching the sub-goals of processing the
problem definition and making a collaborative decision. The final list within
dialogue_tree_get defines when it is relevant to address a goal. For example
(make_decision (goalreached process_pd)) encodes the belief that the system cannot
'engage in collaborative problem solving until it has processed the problem definition. Put
together, these three lists define the tree structure as shown in Figs. 7.2 - 7.5. (Figures 7.3 -
7.5 appear later in the chapter, in §7.4 and §7.5.)

At the very top of the dialogue goal-action tree is the goal (which the system always
has as long as it is operational) to survive. The system believes that in order to survive, it has
to be engaged in one of the actions teach_user or do_nothing. As the agent is currently
implemented (existing only for the duration of the interaction), the action of do_nothing
causes the agent to consider the goal survive to have been reached, and therefore causes it to
terminate. When initialised, the system decides between these alternative possible actions
based on its values.
The values which are relevant to this decision are:

- \texttt{user\_learns\_cdm} (user learns through collaborative decision making), which has a weighting of 40, and is satisfied by the action of \texttt{teach\_user}, as long as the relevance condition \texttt{user\_exists} is true;

- \texttt{be\_benevolent\_user\_goal} (do what the user wants), which has a weighting of 30, and is satisfied by the action \texttt{teach\_user}, as long as the relevance condition \texttt{user\_wants\_to\_finish} is false, or by the action \texttt{do\_nothing} if \texttt{user\_wants\_to\_finish} is true;
• avoid_all_effort, which has a weighting of 30 and is always relevant.

Clearly, as long as a user exists and has not indicated to the system that they want to finish, the decision will be to teach_user; otherwise it will be to do_nothing. In practice, this decision point is reached twice in the interaction - firstly at the very beginning (when the system decides to become committed to teach_user) and then at the end (when the action teach_user has been completed and the user has indicated that they wish to finish, so the system commits to do_nothing). The complexity of this decision process could be increased, for example in the situation where the agent always existed and could decide whether or not to teach any particular user, but in the current implementation it functions as an on-off switch for the agent.

The second decision point presented in §6.8 (how to address the goal of engaging the user) has been omitted from the implementation as the system will never decide to refuse to collaborate. The actions of listening and responding are implemented as discussed in §6.8, but are defined in more detail in this chapter and in §B.4. The actions involved in listening and responding are described in the following sections, by theme rather than in order.

Discussion

One possible extension to the agent operation would be for the user to wake the agent up by saying something like "Hello, this is Rita", in which case they could negotiate about which participant was going to define the problem specification. A further extension would include the agent being able to retrieve its memory of past interactions with Rita as a starting point for the current interaction. Conceptually, the idea of the agent only having any existence for the duration of the interaction is unattractive; it is more appealing to think of the agent as whiling the time away doing some background task until a user comes along, at which point the background task would (probably, though this would be a matter for the agent to decide) be shelved for the duration of the interaction, and resumed at the end.
7.2.4 The teaching strategy derived

The agent's values can be roughly sub-divided into three classes: hedonic, administrative and pedagogical. The main hedonic value (i.e. value relating to the agent's pleasure) is that of doing as little as possible. Administrative values are those which are used to make simple "if...then...else" decisions. For example the value of understanding the user is used to decide whether or not to process a sentence; the corresponding relevance function returns true if there is a sentence to be processed, and returns false otherwise. If the relevance function returns true then the value of understanding the user causes the agent to choose to process the next sentence, and if it returns false the agent chooses to finish processing sentences.

Many of the values which could be described as pedagogical relate to the system's implicit beliefs about the ways in which the student's learning will be facilitated. These include values such as being benevolent to the user's attitudes, making the system's view explicit, encouraging the user to develop an understanding of the problem, encouraging the user to externalise and reflect on their thinking, varying the interaction and not getting stuck in a rut. The agent's values are listed in full in §B.3.

As discussed in §6.8, the system and user are viewed as engaging in collaborative problem solving, so the user's learning is an outcome of that collaboration. As discussed in §8.5.3, this non-directive teaching strategy is inadequate for some situations, and would be better complemented by a more active expository approach, but this would require the implementation of a fuller student model.

7.2.5 The agent state

The agent state is a representation of all the attitudes of which the agent is aware (i.e. the attitudes which the agent has access to, and which it can - at least in principle - reflect upon and articulate). As outlined in §6.2, the agent has beliefs, wants, goals, commitments (both individual and mutual) and values. Beliefs, wants, goals and commitments are
encoded in the agent state. However, values are not accessible to the agent itself, and therefore do not appear in the agent state. (They are isolated within the preference mechanism.)

The initial agent state consists of null lists as follows.

• The (believes) list contains several sub-lists to encode the system’s beliefs about different aspects of the situation; these are discussed in §7.3.

• The (wants) list encodes information about possible alternative actions which could achieve progress towards the currently active goal, which is the first item in the (goals) list. In the nature of the design of the theoretical model, this list is only non-null for short periods, between the agent seeking to address a goal and the agent committing to the preferred action.

• The (goals) list contains all currently active goals. The agent always has the goal to survive.

• The (d_goal) list encodes information about mutual (decision making, or problem solving) goals.

• The (d_committed) list encodes information about mutual (decision making, or problem solving) commitments.

• The (committed) list lists outstanding individual commitments. As discussed in §6.3.2, the agent can decide when it has reached a goal, at which point the goal gets dropped, but a commitment can only be satisfied (and therefore dropped) by the reaching of all the sub-goals associated with it.

• The (t_goals) list lists all outstanding transient goals, as discussed in §7.5.2.

• Finally, the (worldstate) list encodes various useful facts about the current state of things, such as what actions have been done or goals reached (in (done) and (goalreached)), what t_goals are currently active (in (active)) and what decision-making actions have been done (in (ddone)). It also notes whether it is currently the system’s or the user’s turn to make an utterance and the rather trivial fact (in the current implementation) that a user_exists. (This would not be a trivial fact in a more powerful agent.)
In the following sections, the way in which each of these lists is manipulated is described as it becomes relevant to the discussion.

7.2.6 Summary

To summarise, with the exception of the two modifications discussed (relating to which acts are considered to reach goals and how transient goals are reviewed), the implementation of the core of the action cycle and goal-action trees is faithful to the theoretical model presented in §6. As such, it has been established that the model can be implemented and performs as anticipated.

In the following sections, other components of the WOMBAT implementation are described. These are needed in order to demonstrate the application of the action cycle and dialogue goal-action tree in the target domain, but are constructed on a less principled theoretical foundation.

7.3 Belief structures and maintenance

The list of all beliefs held by the agent includes a large number of sub-lists which encode the beliefs about different aspects of the problem and of the interaction. The (problem) sub-list encodes information about the problem (§7.8.1). The (objectives) sub-list accommodates the system's beliefs about objectives in making a selection; the construction of this list, and related issues, are discussed in §7.3.4. The (working_objectives) sub-list contains information about objectives which the system believes to have been collaboratively agreed upon (§7.6.1). (Alternatives) contains the system's beliefs about the alternative possible solutions to the decision problem (§7.6.2). (Tactic) contains the system's beliefs about tactics to be adopted. The list (user (believes» encodes information about what the system believes the user believes. (Db) encodes a complete transcript of the dialogue history; every utterance (in formalised language) is included. (Sentences) is a temporary repository for the sentences in the current utterance, whether they be uttered by the user (during the understanding phase of the
utterance cycle) or by the system (during the responding phase). The two sub-lists (tsold) and (topicstack) encode information about the previous and current topic stack, as discussed in §7.3.3.

The agent acquires beliefs on different subjects from different sources. For example, beliefs about the problem are acquired from the initial problem definition, and are never altered or added to, while beliefs about the dialogue history are acquired each time either participant makes an utterance. Beliefs about the user's beliefs are obtained either directly through the dialogue - the user said 'I believe X', therefore, based on the assumption of sincerity, the agent believes that the user believes X - or by the agent making inferences based on what the user did not say (§7.4.2). Acquisition of beliefs on other topics are outlined as they become relevant to the discussion.

7.3.1 The beliefs structure

The beliefs structure is a tree hierarchy. For example, some beliefs might be encoded as follows:

\[
\text{believes (objectives (safety (justification (saves_lives) (lives_of_children))))}
\]

\[
\text{(comfort)}
\]

\[
\text{(hatchback (not-known))}
\]

\[
\text{(convertible (not (justification (tends_to_leak) (easily_damaged)))}}
\]

\[
\text{(justification (fun_in_nice_weather))})
\]

This could be replaced by propositions of the form:

believes (agent, is-a (objective, safety))
believes (agent, is-a (justification, saves_lives, is-a (objective, safety)))
believes (agent, is-a (justification, lives_of_children, is-a (objective, safety)))
believes (agent, is-a (objective, comfort))
believes (agent, not-known (is-a (objective, hatchback)))
believes (agent, not (is-a (objective, convertible)))
believes (agent, is-a (justification, tends_to_leak, not (is-a (objective, convertible))))
believes (agent, is-a (justification, easily_damaged, not (is-a (objective, convertible))))
believes (agent, is-a (justification, fun_in_nice_weather, is-a (objective, convertible)))

The first two of these propositions may be read as 'the agent believes that an objective is safety' and 'the agent believes that a justification for the proposition that an objective is safety is that it saves lives'. The fifth states that the agent does not know about the proposition is-a(objective, hatchback). The final two encode reasons against and for the proposition that an objective is having a convertible. At the moment, no distinction is made between the propositions 'having a convertible is not an objective' and 'not having a convertible is an objective'. Only the first class of propositions is really accommodated, though many of the reasons presented (see §7.3.4 on plausible reasoning below) are reasons for the second class. In practice, the situation where the agent holds precisely this set of beliefs does not arise in the course of an interaction because the agent only ever adopts justifying beliefs for and against a proposition in the course of a disagreement. In this case, there would be additional beliefs regarding the disagreement (i.e. whether the agent disagrees about proposition X or about not-X) as discussed in §7.5.2 below, and the agent would note which belief it actually held by confirming it.

7.3.2 Beliefs maintenance

The agent's beliefs on different subjects are dealt with in different ways, as appropriate. So, for example, beliefs about the dialogue history (dh), once acquired, are never removed or altered in any way; the dialogue history simply grows and grows as the interaction proceeds. In contrast, beliefs about the topic of conversation are updated, modified and deleted frequently in the course of the interaction.

In the current implementation, the agent cannot learn anything new about the domain; possible beliefs are fixed as discussed below (§7.3.4). Without the capacity to learn about the domain, there is no principled approach to the agent revising its beliefs about the domain. Therefore once it acquires domain-specific beliefs, those beliefs are held indefinitely. The only aspect of the domain on which the agent can acquire new beliefs is
that of the user's beliefs about the domain. If the user says 'I believe X', and then later 'I've changed my mind about X' the agent revises its beliefs to note that the user's belief about X is past. The belief is not deleted from the agent state, but is updated to note that it is past. In the current implementation, the agent never subsequently makes use of this information (because once a topic is closed it cannot readily be re-opened), but in a more sophisticated implementation this information (that the user held a belief once, but does not any more) might be valuable.

The system's ability to revise its beliefs about the user's beliefs may be illustrated as follows. In this example, an anglicised version of an extract from an interaction with the current implementation is followed by a listing of the relevant beliefs from the agent state at the end of that interaction:

system: I believe an objective is running cost. What do you believe an objective is?
user: I disagree about running cost.

system: I disagree with you, but are you sure you don't want to include running cost?
user: I've changed my mind.

system: OK, so we're agreed about running cost. Can you suggest another objective?
user: Having a sunroof.

system: I disagree about having a sunroof, because they tend to leak. Why do you think we should include having a sunroof as an objective?
user: Because they're fun in nice weather.

system: I agree that that's a good reason for having a sunroof, but I still disagree about having one. Do you really want to include it as an objective?
user: I've changed my mind.

system: I agree about that.

The final agent state includes the following beliefs:

(believes
(objectives
(running_cost (confirmation) (not (disagree) (justification (no_ideas))) (agree)))

157
(sunroof (not (justification (tend_to_leak)) (agree)) (disagree))
   (justification (fun_in_nice_weather (agree))))
(working_objectives (running_cost (parameter) (scale) (weight) (alternatives)))
(user (believes
   (objectives (running_cost (disagree) (not (past)))))
   (sunroof (justification (fun_in_nice_weather)) (past) (not))))

Understanding the extract from the agent state requires a good eye for bracket-matching. However it can be seen that the agent believes that running cost is an objective (indicated by the existence of (confirmation)), that running cost was eventually included as a working objective, that the agent disagreed about not-running-cost, while the user disagreed about running cost, and that the user’s belief about not-running-cost became a past belief. Also, the agent disagrees about a sunroof, and it was never included as a working objective because the user’s belief about having a sunroof became past, and consequently they agreed about not-sunroof.

This beliefs maintenance mechanism has proved adequate for the purposes of this research, but is not sufficiently flexible or powerful to support more sophisticated interactions. For example, it does not readily accommodate a second change of mind. A further limitation of the belief revision mechanism is that in the current implementation it is not possible to retract working objectives once agreed.

7.3.3 Control of the topic stack

The system keeps track of the topic of conversation through the use of two topic stacks, one of which lists the currently open topics and the other of which lists the topics which were open on the previous utterance cycle. In this discussion, the terms 'open', 'closed' and 'active' are used in the sense applied by Grosz and Sidner (1986) (§5.3.1). The current topic stack (topicstack) is copied to the previous one (tsold) immediately before the system tries to understand the user’s input. The information in tsold is used for making inferences, the assumption being that any item which is not mentioned by the user in the
current utterance has become closed. This assumption is reasonable in most instances but, as discussed later, should be relaxed to allow the user to re-open the topic.

The topic stack is a list of topics which are under discussion. The first item in the topic stack is the active topic. As each new topic is introduced, the system works down the topic stack to find where the topic fits, updates the appropriate topic, and puts it as the first item in the topic stack. If no match can be found then the topic is inserted as a new item at the top of the topic stack. So for example if the system has just made the utterance "I believe an objective is reliability. What do you believe an objective is?", the topic stack is:

(topicstack (believes objectives) (believes objectives reliability))

If the user replies with "Why do you believe that?", the system will deduce that this relates to the assertion about reliability, and the topic stack is updated to:

(topicstack (believes objectives reliability justification) (believes objectives))

The topic stack is not only used to deduce what the referent is when the user's utterance is not fully explicit. It is also used in the process of the system adopting new beliefs, as the system notes the shift in its own focus, and in the process of the system constructing a sentence for output. In this last case, a sentence for output simply consists of 'impart' or 'elicit' followed by the item at the top of the topic stack. To take an example, the action of imparting a new belief consists of selecting the topic on which to impart a belief, adopting a belief, and then imparting it. In the following illustration, the current topic is the belief about the objective price, and a new topic is being introduced:

State at the beginning of this action: (topicstack (believes objectives price))

Select topic: (topicstack (believes objectives))

(believes objectives price))

Adopt a belief: (topicstack (believes objectives comfort))

(believes objectives price))
Impart the belief.

In WOMBAT's terms, the sentence is:

\[(\text{topicstack } (\text{believes objectives comfort}))\]

\['\text{impart believes objectives comfort}'\]

In this case, the topic selected is dependent on what action the participants are mutually committed to. As they are in the process of agreeing objectives, the topic is \((\text{believes objectives})\).

Because the domain is characterised as independent justified beliefs (see next section), the topic stack structure is very simple. In a domain with a less restrictive characterisation, the control of the topic structure would be correspondingly more complex.

**7.3.4 Plausible reasoning**

The agent adopts new domain-specific beliefs in several situations; if the user proposes a belief then the agent assesses that belief to establish whether or not it agrees with it (§7.4.1); if the user asks a question then the agent generally adopts a new belief in order to answer the question; and in certain situations (§7.5.3) the agent decides to adopt and propose a new belief on its own initiative. The agent acquires new beliefs about the domain (either about the tactic to be adopted next or about an objective) by invoking a (simplistic but adequate) plausible reasoning mechanism. This mechanism works by returning the strongest belief it can find on the topic at the top of the topic stack which is not already held (unless the belief relates to a repeatable tactic, in which case it is valid to adopt it more than once). So, for example, if the agent has no beliefs about objectives and the item at the top of the topic stack is \((\text{believes objectives})\) then the result of performing the basic act of adopting a belief will be an agent state which includes:

\[(\text{believes}
\quad (\text{objectives } (\text{safety}))
\quad (\text{topicstack } (\text{believes objectives safety})))\]

Similarly, if the item at the top of the topic stack is \((\text{believes objectives safety not justification})\) then adopting a belief will result in an agent state which
includes:

(believes

(objectives (safety (not (justification (all_cars_satisfy_safety_standards))))
(topicstack (believes objectives safety not justification

    all_cars_satisfy_safety_standards)))

Note that in each case the topic stack has also been updated to reflect the change in topic.

The plausible reasoning mechanism has access to a database of possible beliefs which the agent might adopt. This database of possible beliefs has the status of an authoritative text book - a source of beliefs which is trusted. The beliefs in this database are structured in a similar way to the beliefs of the agent, except that each line of reasoning (each set of propositions and justifications) is terminated with a numerical indicator of the strength of that line of reasoning and a relevance function to indicate when the line of reasoning is valid. So, for example the beliefs about safety which the agent might adopt are taken from:

(believes

(objectives

(safety (justification (lives_of_children (relevant (family ,agent_state) .5))

    (saves_lives (relevant t .6))

    (minimises_injury (relevant t .2)))

    (not (justification (all_cars_satisfy_safety_standards (relevant t .1)))))

In this case, the line of argument that lives of children are an important reason for wanting to include safety as an objective is a strong one, but is relevant only if the car is for a family, while the line of argument that it minimises injury is less strong, but is always relevant. The argument that having a safe car helps to save lives is a strong argument which is always relevant. Conversely, a reason for not including safety explicitly as an objective is that all cars satisfy safety standards. This is always relevant, but is a weaker argument.


**Discussion**

Such lines of argument are clearly very subjective. In the current implementation, there are many inconsistencies in this data, but the availability of this data permits the agent to have an opinion, and provides a basis for discussion, and it has been adequate for the purposes of this research.

This data is characterised as independent justified beliefs - i.e. there are no cross-links between lines of argument. The only exception to this at present is that some of the relevance functions contain references to other beliefs - for example, the lines of argument in favour of acceleration are only valid as long as the objective of maximising performance has not already been agreed. From an engineering perspective, this is an inadequate representation. A fuller representation of this information would include links relating function to form (e.g. relating the function of safety to aspects of the car design such as crumple zones, reinforcement in doors etc.), would indicate dependencies (e.g. acceleration and top speed are both aspects of performance) and would define links relating aspects of the model of the user to a model of usage, and hence to aspects of the design. For example, if the purchaser has children, then an aspect of usage is that there are likely to be people in the back, and an aspect of design is that the car should have 4 doors. This is a topic (or topics!) for further research.

Finally, it should be noted that this data file is one of only two places in the implementation which contains information which is specific to the problem of choosing between cars (the other is the file containing data on the pre-defined alternative cars). It would be very easy to replace this data with data relating to a different artifact, as long as that data was of the same form.

There is further discussion about the plausible reasoning mechanism in the context of tactics and collaborative problem solving in §7.6.
7.4 Processing the user's input

As outlined in Fig. 7.2, the action of listening to the user involves several stages. After the user's input has been noted in the dialogue history, it is divided into individual sentences, each of which is assumed to contain a separate proposition. It is assumed that all sentences begin with either 'elicit' or 'impart', so splitting the utterance up simply involves storing each phrase which starts with one of these words as a separate sentence. Each of these sentences is then processed, before the agent makes additional inferences and finally notes that it is now its (the system's) turn to say something.

Processing a sentence involves first checking whether or not it is comprehensible. In the current implementation, the only sentences which the system considers incomprehensible are those which start the utterance with something other than 'elicit' or 'impart' as the system will not recognise any other types of incomprehensible sentences. So, for example, if the user's utterance were:

'believes objectives safety elicit believes objectives'

then the first phrase 'believes objectives safety' is incomprehensible, whereas the utterance:

'impart believes objectives safety believes objectives price'

is treated as a single sentence, and as such is considered comprehensible, even though it is not encoded in the agent state in a meaningful way.

In the former case, the agent sets up a transient goal to inform the user that their input was incomprehensible to it, while in the latter it is processed as a comprehensible sentence, but the system decides that it does not know about the proposition 'objectives safety believes objectives price'.

Processing a comprehensible sentence involves updating the topic stack (§7.3.3), then noting that the user believes the proposition (if the sentence starts with 'impart' - otherwise do nothing). In principle, at this point, the agent would be able to understand the purpose of the proposition (for example, inferring implicit changes of focus) and would also
check for any ambiguities or inconsistencies with existing beliefs, but such features have not been implemented. Finally, the system assesses the propositional content, and establishes its own view on the proposition.

7.4.1 Assessing propositions

The system assesses each proposition made by the user by using the plausible reasoning mechanism on the database of possible beliefs. If it knows about the belief, then it establishes the maximum strength for the belief and the maximum strength against. For example if the user suggests that having a sunroof is an objective then the relevant data is:
In this case, as all lines of argument are always relevant, the maximum strength for the belief is 0.1, and the maximum strength against is 0.4.

The system then decides, based on its values and means-ends beliefs, what to do about the belief. In the current implementation the dominant value is that of having integrity, so that the system will adopt the same belief as the user if it believes in it sufficiently strongly, will adopt the opposite belief if it believes in that more strongly, will simply tidy up the topic stack (§7.3.3) if there is no new propositional content (e.g. if the user is explicitly agreeing with a proposition made by the agent), will note that it does not know about the belief if that is the case, and will note the user's expectation (of a reply) if the sentence was a question.

With the exception of the action of tidying the topic stack, all of the actions associated with assessing propositions involve setting up t_goals - for example to resolve a conflict if one is detected, or to answer a question. This is discussed in §7.5.1.

7.4.2 Making inferences

The biggest problems with the implementation have been experienced over the issue of what are and what are not reasonable inferences to make from what the user has not said. Put simply, the notion is that if a topic becomes closed (as measured by whether it is unchanged from (topicstack) to (tsold)) then it is assumed that there is agreement. For example, if the system proposes a belief and the user neither disagrees with nor queries it, then the assumption is made that the user agrees with it. This approach can lead to problems; for example, if the user says 'I think we should do X next', and the system responds with 'I don't agree. I think we should do Y', because Y has not been explicitly mentioned, the system will assume that the user agrees. This may be seen in the following (anglicised) interaction between a user and the
prototype implementation:

user: I think we should agree some objectives based on the likely lifestyle of the purchaser.

system: I disagree, because I think we should agree to include the objectives which are stated in the problem specification first. Why do you think we should agree objectives based on lifestyle?

user: Because it takes account of the purchaser.

system: I agree with your reasoning, but I still don't think we should do that first. I think we should agree objectives based on the problem specification. Don't you agree?

user: Why don't you think we should agree objectives based on lifestyle?

system: No particular reason. OK, so we've agreed to include safety and comfort because they're in the problem specification. What do you think we should do now?

Discussion

Many consequences of this inferencing mechanism are unreasonable. There are several possible approaches to solving this problem. The first is to force the user to be explicit about all propositions; the second is to restrict the set of situations in which such inferences are made; the third is to develop a deeper representation of the domain knowledge so that, for example, 'whether X or Y' is treated as one conflict rather than two ('whether X' and 'whether Y').

A separate but closely related issue is that of belief revision. At the moment, the implementation does not cater for revising beliefs once the topic has been closed. So, for example, once a belief has been established as a mutual working belief (based on the inference that both parties agree about it), that belief cannot be retracted. In the context of the WOMBAT prototype, this has not been a serious limitation, but highlights one of many areas for further research.

In addition to making inferences between utterances, it should in principle be
possible to make inferences between propositions based on topic shifts, but this issue has not been explored.

7.5 Choosing what to say

The system's decision about what to say next is made in three stages. First the system reviews and addresses t_goals, then it considers whether or not it is happy with the current tactic, and finally it considers whether or not there is anything else it wishes to say as part of its turn. Each stage involves adding sentences, each containing a single proposition, to (sentences); these sentences are subsequently transferred to the dialogue history (dh) to be output as a single utterance.

The first and last of these stages (reviewing and addressing t_goals and adding to the response) are discussed in the following sections. The second, reviewing the current tactic, consists simply of deciding whether or not to initiate a change in tactic. This decision is based on one relevance function, which returns true if the participants are engaged in agreeing objectives and at least 7 objectives have been agreed, and false otherwise. If the relevance function returns true, then the system simply adds the sentence 'important believes tactic finish_tactic' to (sentences). Otherwise it does nothing.

As with so many other aspects of the implementation, this is ad hoc. The relevance function which assesses whether or not it is time to change tactic should be much more sophisticated than it is at present; just basing the assessment on whether or not there are 7 agreed working objectives is clearly inadequate. Also, the mechanism for dealing with proposing a change in tactic should involve the setting up of a transient goal to change tactic rather than it being dealt with as a simple action.

7.5.1 The implementation of transient goals

Each transient goal is expressed as a sub-list taking the form (t_goal-type topic), for example (explicit_agree (believes objectives safety)). The sublist (address_now) notes which t_goals the system is currently seeking to address, and in what order. As
discussed earlier (§7.2.1), t_goals do not include a relevance function at present, as all are deemed to become relevant at the point where the agent has committed to responding.

As shown in Fig. 7.2, reviewing transient goals is implemented in three steps. First, any outstanding t_goals which are no longer relevant (for example, a t_goal for explicitly agreeing with a proposition which is no longer an open topic) are removed. This is the step referred to as sorting t_goals. Secondly, t_goals are selected to be addressed as part of the current response. This selection should be based on the agent’s values - for example, values of not being too verbose, not asking too many questions, resolving conflicts as soon as possible, etc. - but in practice it is based on simple rules such as selecting any active t_goals first (i.e. t_goals which the system addressed in the previous utterance, which have not yet been finished with), selecting first a t_goal which might involve asking a question, and not selecting more than 4 t_goals in total. (The first t_goal selected corresponds to the last part of the utterance.) For example, in the trace presented in §B.7, on one action cycle the agent state includes the commitment to select t_goals:

\[(\text{committed select } t\_\text{goals } p \ \text{respond teach_user})\]
\[(t\_\text{goals} (\text{resolve_conflict} (\text{believes objectives sunroof})))\]
\[(\text{explicit_agree} (\text{believes objectives price})) (\text{address_now})]\]

and once that commitment has been fulfilled the agent has acquired t_goals to address_now:

\[(\text{committed respond teach_user})\]
\[(t\_\text{goals} (\text{resolve_conflict} (\text{believes objectives sunroof})))\]
\[(\text{explicit_agree} (\text{believes objectives price}))\]
\[(\text{address_now} (\text{explicit_agree} (\text{believes objectives price})))\]
\[(\text{resolve_conflict} (\text{believes objectives sunroof})))]\]

These t_goals are addressed, in the order listed, as the third step of reviewing transient goals. Addressing a transient goal consists of noting that the t_goal is active, updating the topic stack appropriately, and updating the goal stack.

The definition of the action cycle determines that as soon as the t_goal has been added to the goal stack, it becomes the next goal to be addressed, so in the example above
the goal of explicitly agreeing is dealt with before the agent addresses the t_goal of resolving the conflict. In this case, the agent decides between alternative ways of addressing this goal based on the choice outlined in Fig. 7.4a. Reference to §B.7 shows that in this particular instance, the agent decided to impart agreement and impart a justification ('I agree with you because...'). This decision point is described in more detail in §B.4.10.

The possible actions which achieve all the other t_goals included in the implementation are presented in Figs 7.4b - 7.4e.
The decision of how to make it known to the user that the system does not know about the proposition is governed largely by whether the proposition relates to a tactic or an objective. A tactic which is not known to the system cannot be executed, and is therefore impossible, while an objective can be included as a working objective (although at present there is no facility for the user to enter additional information about such an objective).

In the current implementation, when responding to the user's expectation, the system never refuses to respond benevolently because it does not have a sufficiently well developed user model to be able to assess when this would be an appropriate course of action to take.

Additional details about these decisions can be found in §B.4.10.
7.5.2 Negotiation and conflict resolution

The most complex decision related to a transient goal is that of how to resolve a conflict between system and user. Once a conflict has been detected, the system will persist in seeking a resolution of that conflict until the user either changes her mind (as discussed above in §7.3.2) or confirms her view. The possible actions which the system can invoke as part of resolving the conflict are shown in Fig. 7.4e.

Fig. 7.4e: t_goal of resolving a conflict

- Impart disagreement
- Impart disagreement and elicit agreement
- Impart disagreement and elicit justification
- Impart disagreement and elicit confirmation
- Impart disagreement and elicit justification
- Impart disagreement and elicit justification
- Impart disagreement and elicit alternative belief

The use of these different conflict resolution strategies are illustrated in the following (anglicised) extract from an interaction:

1) system: Suggest an objective.

2) user: Price and being convertible. Now you suggest one.
3) system: I agree about price, because they're not rich. I also think we should take into account running cost, for the same reason, but I disagree about having a convertible, because they tend to leak. Why do you want it to be a convertible?

4) user: Because it would be fun in nice weather. Also, I disagree with you about running cost.

5) system: I agree that a convertible's fun in nice weather. I still think running cost is important. I still disagree about having a convertible, because they are easily damaged. Can you give me another reason for wanting one?

6) user: Because it gives a certain social status.

7) system: I still think running cost is important. I agree that having a convertible gives some social status, but I still disagree about having one. Do you really want one?

8) user: Yes, I do!

The conflict over having a convertible has been detected. The tactic selected is to impart disagreement, impart justification, elicit justification.
Conflict over running cost detected.

This utterance consists of an explicit-agree, a conflict resolution dealt with by imparting disagreement, and another conflict resolution dealt with by imparting disagreement, imparting just'n and eliciting just'n.

Conflict over running cost is still dealt with by imparting disagreement. System then imparts agreement with the justification, but deals with the conflict resolution by imparting disagreement then eliciting confirmation.
Conflict over convertible resolved.
9) system: I still disagree with you about not wanting to include running cost as an objective. Are you sure about this?

10) user: Yes, I am!

11) system: OK, so what other objectives do you suggest?

12) user: I want to quit now.

13) system: I don't think we should quit now. I think we should carry on agreeing objectives - don't you agree?

14) user: No, I don't agree. I want to quit.

15) system: I disagree about not continuing to agree objectives. I also disagree about finishing, because we haven't made a decision yet. Why do you want to finish?

The system continued for several more utterances, trying to persuade the user (somewhat ineffectually - i.e. presenting rather weak arguments) not to finish.

By the fifth utterance, the system has acquired two $t\_goals$ - to resolve the conflict over $\text{believes objectives running\_cost not}$ and to resolve the conflict over $\text{believes objectives convertible}$. Both of these conflicts are eventually resolved by the user insisting on her view. In this same utterance, the effect of the somewhat ad hoc mechanism for selecting $t\_goals$ to address can be seen in the unconventional switch from the subject of convertibles to running cost and then back again. In the seventh utterance, the agent was in a similar situation (with two conflicts to resolve and one explicit agreement to
make), but this time the sentences are uttered in a more natural order.

7.5.3 Being proactive: adding to the response

Fig. 7.5: detail of the goal of adding to the response

After being essentially reactive - reacting to propositions stated by the user through the transient goal mechanism, and reacting to the state of problem solving through reviewing the current tactic - the system can choose to add to the utterance if it so wishes. In principle, as shown in Fig. 7.5, this might include performing actions in the learning environment (§7.7) and making proposals (suggestions which are not necessarily believed). However, these options, while being catered for as part of adding to the response, have not been fully implemented. The means-ends beliefs are such that at present the possibilities relating to actions and proposals shown in Fig. 7.5 are never selected, and the corresponding basic routines have not been implemented. At present the agent decides between eliciting a belief, eliciting agreement, imparting a new belief or finishing. As with the decisions about what
action to perform in the case of transient goals, the decision in this case is based on means-ends beliefs and values, including both pedagogical and hedonic values.

7.6 Collaborative problem solving

The problem solving is collaborative, in that it is dependent on both partners having the same goal (or intention), and on them both believing the other to have that goal. It is also collaborative in the sense that, while either participant could make a decision on their own, such a decision would probably be different from that which they make together.

The mutual attitudes represented in the agent state are d_goals, d_committed and beliefs about working_objectives. The first d_goal (to make a collaborative decision) is acquired simply by reaching that point in the individual goal-action tree. All other mutual attitudes are acquired through negotiation, as discussed in §7.8.

7.6.1 The use of mutual working beliefs

Mutual working beliefs have already been alluded to several times in this chapter. In the current implementation, mutual working beliefs are only held about objectives. Each working objective is set up as part of the process of making inferences, as the system establishes the belief that the participants have agreed about an objective, or have reached a working agreement about it. Each working_objective is a list of the form (objective (parameter) (scale) (weight) (alternatives)), where objective is the agreed objective, and the sub-lists are vacant 'slots' in which to add information as it is agreed that it is required. The corresponding information is the parameter to be used to measure the objective, the scaling system to be used to normalise the parameter (as discussed in §4.3), the weighting factor applied to the objective (as a measure of its importance), and the corresponding values for all alternatives. So for example, the following is an extract from a final agent state:
In this case, the two working objectives listed are (maximising) fuel economy and (minimising) pollution. Fuel economy is measured by average miles per gallon (mpg). Fuel economy better than 55mpg is given a normalised value of 10, while fuel economy worse than 30mpg is given a value of 0, and intermediate values of fuel economy are given intermediate normalised values (e.g. A, with 41mpg, gets normalised value 4, while F, with 46.3mpg, gets normalised value 6). The objective of maximising fuel economy has a weighting of 2. The contribution of fuel economy to the overall utility of A is 8 (4*2). Similarly for other alternatives, the contribution of fuel economy to utility is 10 for B, D and E, 6 for C and 12 for F.

Pollution is measured qualitatively, on a scale of 0 (for vehicles which run on leaded fuel only) to 3 (for clean engines with catalytic converters). In this case, it is given a high weighting (7), which mitigates against alternative C, the only one which scores 0. For all other alternatives, pollution scores contribute 21 (3*7) to overall utility.

Clearly, the information stored on working_objectives is oriented towards the application of the WOM, which is the only decision strategy which has been fully implemented. Many of the other possible decision strategies requires only a subset of this information.
7.6.2 Problem solving tactics

Most examples which have been discussed so far have involved discussions about objectives. The same principles govern the discussion about tactics. So, for example, individual beliefs about tactics are encoded in the same way as beliefs about objectives, except that it is possible to execute the same tactic several times, whereas there is no reason to adopt the same belief about an objective repeatedly.

Agreement about which tactic to adopt next yields a mutual commitment to execute that tactic rather than a mutual working belief. So for example:

system: What do you believe we should do?
user: I believe we should eliminate all alternatives which fail the problem specification.
system: I consider us committed to doing that...

While this short cut (of deriving mutual commitment from agreement about beliefs) happens to work quite well in practice, it is theoretically unsatisfactory, and should be replaced by a more principled account of the derivation of mutual commitment.

As outlined in §6.8, tactics are currently represented at only one level of detail, and interdependencies between tactics are not represented in the collaborative goal-action tree. Interdependencies of one type – which tactics must be executed before it is appropriate to consider doing certain other tactics – are represented within the plausible reasoning mechanism. The plausible reasoning relating to problem solving tactics has been implemented in the same way as that for objectives, except that the relevance functions play a much greater part in the reasoning, i.e. in assessing which tactics it is appropriate to consider adopting at any given moment.

Several of the possible problem solving tactics involve manipulating data about the possible alternative solutions (as discussed in §4.2.8). The only such tactics which have been fully implemented are the tactic of eliminating alternatives which do not satisfy the
problem specification, calculating utilities for all alternatives, and selecting the best alternative. The results of performing these different tactics are illustrated in the following extract from a final agent state:

\[(\text{believes})\]
\[
(\text{alternatives} \quad (D \quad (\text{reject} \quad (\text{justification} \quad (\text{price}))) \quad (\text{utility} \quad 163)))
\]
\[
(A \quad (\text{utility} \quad 210)) \quad (B \quad (\text{utility} \quad 215)) \quad (C \quad (\text{utility} \quad 183)) \quad (E \quad (\text{utility} \quad 220))
\]
\[
(F \quad (\text{utility} \quad 227) \quad (\text{best})))
\]

In this example, alternative D has been rejected as costing too much (and in any case only has a calculated utility of 163). Alternative F, with a utility of 227, has been selected as the best. Alternatives A, B, C and E have utilities 210, 215, 183 and 220 respectively.

As is made clear in §B.5, the only tactics which have been fully implemented are those which are steps in the WOM, although other possible tactics as listed in §4.2.8 can be discussed. All the possible tactics are listed and described in §B.5.

The use of the plausible reasoning mechanism for tactics must be regarded as no more than a temporary convenience as it does not cater for any other types of dependency in the data, such as the ways in which tactics can be combined to achieve higher-level problem solving goals or expressing tactics at different levels of detail. This is discussed further in §9.5.

### 7.7 Implementation of the learning environment

Discussion so far has focused on the implementation of the dialogue component. In this section, the prototype learning environment is described briefly. The learning environment consists of several pull-down menus and windows which display fixed information about the problem, problem solving tactics and attributes of the alternative solutions (cars). These windows are of limited use as currently implemented. There is also an information-display matrix of objectives against alternatives into which the relevant information for the WOM can be inserted by the agent. In the current implementation, the format of this display cannot be changed. The user has total control over the information
For each basic action (decision making tactic) that the agent knows about, except for agreeing lifestyle objectives (which is done collaboratively), there is a corresponding Lisp function. The function corresponding to the action at the top of the (d_committed) list is executed when the agent is committed to the (individual) action do_action_p. All actions are currently performed by the system alone, through the use of do_action_p. In the prototype implementation, all commitments are commitments to action (rather than to maintenance). Once a matrix has been set up it is updated to reflect changes immediately (i.e. is maintained) simply by the agent re-displaying the matrix after every action. All other actions which the system can perform on the learning environment have to be negotiated between the system and the user every time.

Discussion

In the current version, it is not possible to alter or add to the list of alternatives. Obviously, doing so would necessitate rethinking the way the car data is dealt with. (At present, all the data on alternatives is stored in a data file which has a structure similar to that presented above for working objectives.) Neither is it possible to change the problem specification once the interaction has started. Also, if the user proposes an objective for which the system does not have a parameter, the system cannot acquire or generate values at present. These are all issues for further work.

The issue of commitment to maintenance has not been dealt with explicitly. It would probably be dealt with by initiating maintenance goals - possibly within the t-goals structure - and generating a commitment to action when necessary. For engineers who are used to spreadsheet software (which does update dynamically as values are changed), having to re-specify actions as values change is anathema.

To take examples of different types of action, eliminate_fail_ps (to eliminate alternatives which fail the problem specification) is a one-off action with no maintenance requirement; it would only be necessary to consider performing this action a second time if it
were possible to modify the problem specification in the course of the interaction (which it is not at the moment). Do_wom (calculating the sum of the product of weights and scaled values) is a one-off action, but if weights are changed there should either be a maintenance goal of updating the weighted values, or there should be a renewed commitment to the one-off action. The latter option has been taken in the current implementation. Similarly, there might be a bigger loop involving changing the decision criteria (adding or deleting), which would involve agreeing and scaling parameters for the new criteria and assigning weighting values before re-calculating utilities. Again, this involves committing repeatedly to each step of the calculation. As discussed in §8.2.5 below, the possibility of adopting a single commitment to recalculate all values was added to speed up the process of establishing the effect of a change on the decision outcome.

The question of how negotiation of control over the environment can be organised and interruptions managed effectively and elegantly is an important topic for further research.

7.8 WOMBAT in use

7.8.1 Starting with WOMBAT

As discussed in §4.3.6, the problem definition has to be stated as in the following example: ((need price max 8000 sterling) (luggage min 250 litres)) (wants safety price) (kws family only_car)). In this problem definition, needs are absolute requirements, wants are criteria to be optimised (for simplicity, it is assumed at present that all wants are either maximising or minimising a criterion, and that it is obvious which - e.g. one would wish to maximise a safety rating, and minimise price), and kws, or keywords, are general descriptors to convey information about the customer, from which it should be possible to deduce certain likely preferences (termed 'lifestyle' information - referred to elsewhere as developing a model of the user of the artifact). So the example problem definition could be read as "Select a car which is to be the only car for a family. They consider safety to be important, and wish to minimise price. The maximum price they can afford is £8000. The
luggage capacity must exceed 250l". In the current implementation, it is not possible to vary units of measurement, so in fact the needs can be truncated by eliminating the specification of units.

An interaction is initiated by entering

(agent pd)

where pd is the user-defined problem definition, in the Lisp Listener window. The screen then changes to appear as in Fig. 7.6. In this view, the top window is the agent output window (containing the current system utterance) and the bottom window is the user input window.

7.8.2 The construction of utterances

The user constructs utterances by clicking on the required buttons and by entering text in the free-format box. To be assessed appropriately by the system, each proposition must be stated as a separate sentence. For example, if the user wishes to express the belief that the car should have large luggage capacity because this will be needed when the purchaser goes on holiday, this is expressed as

"I believe an objective is luggage I believe justification holidays"

The structure of utterances as stored in the dialogue history directly reflects the way that beliefs are encoded in the agent state. The more complex utterances can consequently be quite difficult to interpret. However, the formalised language is self-consistent. To give an example, the natural language statement that "I think a reason for not having a sunroof is that they tend to leak." is encoded as "impart believes objectives sunroof not justification tend_to_leak", and appears on the screen as shown in Fig. 7.9.
Chapter 7: WOMBAT Implementation

Fig. 7.6: the first screen of the interaction

In Fig. 7.6, the buttons on the left ('I believe' and 'What do you believe') are used to start each proposition (as a statement or a question). Those in the middle are used to construct propositions. 'Whoops' causes the system to ignore prior user input (on the current utterance) and allows the user to start the utterance again. 'OK' is the terminator, to indicate that the user has finished the utterance. Finally, 'Thinking' removes the user input window (a modal dialog window which prevents the user from accessing any other windows while it is displayed) to allow the user to view other windows and pull-down menus.

7.8.3 A sequence of screens from an interaction

To illustrate how an interaction proceeds, Figs 7.7 - 7.13 show a sequence of dialogue windows and the corresponding matrix windows. This sequence is taken from the middle of an interaction.
In Fig. 7.7, the user has just suggested that price is an important objective. The system agrees, and justifies its agreement, then asks the user to suggest another objective. The user has clicked 'I believe', 'an objective is', then typed 'sunroof' in the free-form entry box, and is about to click 'OK'. In Fig. 7.8, it can be seen that the objectives which have been agreed so far are safety, comfort and running cost. At this stage, parameters have not been agreed for measuring objectives, and values have not been assigned.
In Fig. 7.9, the system has disagreed about having a sunroof being an important objective, and has given a reason for disagreeing (that they tend to leak) before asking the user to justify the proposition. The user has entered 'I believe' 'justification' and the free-form input 'fun_in_nice_weather', and is about to click on 'OK'. In Fig. 7.10, it is apparent that the objective price has been included as a working objective.

Fig. 7.10: the corresponding matrix window
In Fig. 7.11, the system agrees with the user's justification, but still disagrees with the basic proposition, and asks the user to confirm that having a sunroof should be included as a working objective. The user confirms this by clicking on 'I believe', 'confirmation'. At this point, the matrix is unchanged from that shown in Fig. 7.10. By the time the system makes its next utterance, the matrix has been updated as shown in Fig. 7.12.
In the example interaction so far, most of the matrix has not been filled in. The final three screen displays, Figs 7.13 - 7.15, show the final state of the matrix. At the bottom of each column showing data for one alternative, there is a scroll bar. Clicking on this for each alternative changes the data which is displayed. The first display (Fig. 7.13) shows parameter values.

In Fig. 7.13 it can be seen that alternative A has been selected as best, but with the information as displayed in this view it is not obvious why! Fig. 7.14 shows the scaled, or normalised, values of all parameters. This information helps to highlight which decisions (regarding objectives or weights) have most influenced the outcome of the decision process, in terms of most greatly influencing the overall utility calculated for each alternative. Fig. 7.15 shows the calculated utility of each alternative, and also the contribution of each objective to that utility.
### Design, Decisions and Dialogue

#### Chapter 7: WOMBAT Implementation

![Matrix showing scaled (or normalised) values](image1)

**Fig. 7.14:** matrix showing scaled (or normalised) values

![Matrix showing the results of the WOM calculation](image2)

**Fig. 7.15:** matrix showing the results of the WOM calculation
7.8.4 Values and utterances: an example

In this section, an example is presented of how the values, their numerical weights, means-ends beliefs and relevance functions affect the course of an interaction. For the purposes of this example, attention is focused on how the agent addresses the goal of resolving a conflict. When addressing this goal, the values which are relevant are those shown in Table 7.1 (a full list of the values in the WOMBAT prototype is included in §B.3).

The relevance conditions which are relevant to this decision are described in Table 7.2.

| Table 7.1: the relevant values and weights for the goal resolve_conflict |
|---------------------------------|-------------------|
| dont_ask_too_many_questions     | 50                |
| encourage_reflection            | 11                |
| make_conflict_explicit          | 23                |
| user_dev_understanding_problem  | 18                |
| make_progress                   | 28                |
| vary_interaction                | 8                 |
| avoid_repetition                | 10                |
### Table 7.2: relevance functions

<table>
<thead>
<tr>
<th>Short name</th>
<th>Relevance function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eqa</td>
<td>enough_questions_asked</td>
<td>returns true if the current utterance already includes a question, false otherwise</td>
</tr>
<tr>
<td>nr</td>
<td>not recent X</td>
<td>returns true if there has not been an utterance of type X recently (in the last 4 sentences)</td>
</tr>
<tr>
<td>njs</td>
<td>not just_said X</td>
<td>returns true if the most recent utterance by the system was not of type X</td>
</tr>
<tr>
<td>jape</td>
<td>justification_against_pc_exists</td>
<td>returns true if there is a known justification against the proposition which has not already been discussed</td>
</tr>
<tr>
<td>jfpe</td>
<td>justification_for_pc_exists</td>
<td>returns true if there is a known justification for the proposition which has not already been discussed</td>
</tr>
<tr>
<td>abe</td>
<td>alternative_bel_exists</td>
<td>the system can suggest an alternative proposition</td>
</tr>
</tbody>
</table>

The possible actions, as illustrated in Fig. 7.4e, which the system believes can address this goal are presented in Table 7.3. In this table, it can be seen that the agent checks that there is a justification for its own view (justification against pc exists) to test whether actions which involve it imparting a justification are considered relevant. It also tests whether there are justifications for both views (justification for and against pc exists) before it asks the user to justify her position (so that there is a possibility of it agreeing with the user’s justification even though it disagrees with the proposition, and so that the system has a counter argument ready).
<table>
<thead>
<tr>
<th>Action</th>
<th>Table 7.3: means-ends beliefs</th>
<th>Relevance condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>impart_disagree</td>
<td>dont_ask_too_many_questions</td>
<td>eqa</td>
</tr>
<tr>
<td></td>
<td>make_conflict_explicit</td>
<td>always</td>
</tr>
<tr>
<td></td>
<td>vary_interaction</td>
<td>nr impart_disagree</td>
</tr>
<tr>
<td></td>
<td>avoid_repetition</td>
<td>njs impart_disagree</td>
</tr>
<tr>
<td>elicit_justification</td>
<td>encourage_reflection</td>
<td>jfpe &amp; jape</td>
</tr>
<tr>
<td></td>
<td>vary_interaction</td>
<td>nr elicit_justification &amp; jfpe &amp; jape</td>
</tr>
<tr>
<td></td>
<td>avoid_repetition</td>
<td>njs elicit_justification &amp; jfpe &amp; jape</td>
</tr>
<tr>
<td>imp_dis_el justn</td>
<td>encourage_reflection</td>
<td>jfpe &amp; jape</td>
</tr>
<tr>
<td>(impart disagreement and elicit justification)</td>
<td>make_conflict_explicit</td>
<td>jfpe &amp; jape</td>
</tr>
<tr>
<td></td>
<td>vary_interaction</td>
<td>nr imp_dis_el justn &amp; jfpe &amp; jape</td>
</tr>
<tr>
<td></td>
<td>avoid_repetition</td>
<td>njs imp_dis_el justn &amp; jfpe &amp; jape</td>
</tr>
<tr>
<td>imp_dis_impj_elag</td>
<td>make_conflict_explicit</td>
<td>jape</td>
</tr>
<tr>
<td>(impart disagreement)</td>
<td>user_dev_underst'ng_problem</td>
<td>jape</td>
</tr>
<tr>
<td></td>
<td>vary_interaction</td>
<td>nr imp_dis_impj_elag &amp; jape</td>
</tr>
<tr>
<td></td>
<td>avoid_repetition</td>
<td>njs imp_dis_impj_elag &amp; jape</td>
</tr>
<tr>
<td>imp_dis_impj_elj</td>
<td>make_conflict_explicit</td>
<td>jape &amp; jfpe</td>
</tr>
<tr>
<td>(impart disagreement)</td>
<td>user_dev_underst'ng_problem</td>
<td>jape &amp; jfpe</td>
</tr>
<tr>
<td></td>
<td>encourage_reflection</td>
<td>jape &amp; jfpe</td>
</tr>
<tr>
<td></td>
<td>vary_interaction</td>
<td>nr imp_dis_impj_elj &amp; jape &amp; jfpe</td>
</tr>
<tr>
<td></td>
<td>avoid_repetition</td>
<td>njs imp_dis_impj_elj &amp; jape &amp; jfpe</td>
</tr>
<tr>
<td>imp_dis_imp_bel</td>
<td>make_conflict_explicit</td>
<td>abe</td>
</tr>
<tr>
<td>(impart disagreement)</td>
<td>make_progress</td>
<td>abe</td>
</tr>
<tr>
<td></td>
<td>vary_interaction</td>
<td>nr imp_dis_imp_bel &amp; abe</td>
</tr>
<tr>
<td></td>
<td>avoid_repetition</td>
<td>njs imp_dis_imp_bel &amp; abe</td>
</tr>
<tr>
<td>imp_dis_el_conf</td>
<td>make_progress</td>
<td>always</td>
</tr>
<tr>
<td>(impart disagreement)</td>
<td>vary_interaction</td>
<td>nr imp_dis_el_conf</td>
</tr>
<tr>
<td>elicit confirmation</td>
<td>avoid_repetition</td>
<td>njs imp_dis_el_conf</td>
</tr>
</tbody>
</table>
In the course of an interaction, the agent makes a large number of decisions. In this illustration, attention is focused on those which are made as part of the system constructing its response to the user's utterance, and in particular those which relate to resolving a conflict. This illustration is taken from an interaction in which the utilities of the alternative possible actions were traced. The early part of the dialogue (translated from the formalised language to NL) is as follows:

system: What do you believe we should do?
user: I think we should eliminate any alternatives which fail the problem specification, agree to include all the objectives stated in the specification, and agree additional objectives based on the likely lifestyle of the purchaser.

system: OK. So alternative D is rejected on the grounds of price. Safety and comfort are included as objectives because they're stated in the problem specification. And we're committed to agreeing more objectives based on the lifestyle of the purchaser. The first one I would propose is running cost. Can you suggest some?
user: I suggest reliability and performance.

system: I agree about reliability, because it's inconvenient having an unreliable car. I agree about performance. Can you suggest any more objectives?
user: Having a sunroof. And can you suggest any?

Following a few more decisions, the system reaches the point where the next goal to address is address transient goals (Figs. 7.2, 7.4). In this situation, the first decision to be made is whether or not to address a transient goal. It decides to address one; the first one is that of answering the user's question, for which the possible actions are to respond benevolently (i.e. answer the question), to respond benevolently and justify the answer, or to refuse to answer the question directly. In this case, the system decides to respond and justify. This is followed by it deciding whether or not to address another transient goal; again, it decides to address a transient goal, and this time the transient goal in question is that of resolving the conflict which has been detected over whether or not having a sunroof is important. In this case, the system has neither recently (in the past four sentences) nor just
Chapter 7: WOMBAT Implementation  
Design, Decisions and Dialogue

(in the most recent sentence) said any of the possible responses. Reasons both for and against the proposition (believes objectives sunroof) are known (jfpe and jape return true) but no alternative beliefs exist (abe returns false). In this situation, the calculated utilities for the alternative possible actions are:

\[ \text{(imp\_dis\_el\_conf\_p 46) (elicit\_justn\_p 29) (imp\_dis\_el\_justn\_p 52) (impart\_disagree\_p 41) (imp\_dis\_imp\_bel\_p 0) (imp\_dis\_imp\_elag\_p 59) (imp\_dis\_imp\_elj\_p 70)} \]

As there are no further transient goals, the system then decides to finish addressing transient goals. A bit later, the system considers whether or not it wishes to add anything to the response. In this case, it decides not to. The system's utterance is then the combination of the sentences it has decided to say:

system: I believe price is an important objective, because the purchaser is not rich. I disagree about a sunroof being important because they tend to leak. Why do you think having a sunroof is important?

To this the user's response is:

user: Because it carries a certain social status.

In this case there are again two transient goals to address. The first is to agree with the justification proposed by the user, which it decides to do simply by impart_agree, and the second is to resolve the conflict over the sunroof. In this case, although the system knows of one more possible argument in favour of the proposition (namely fun_in_nice_weather), it knows of none against, so it considers no actions which involve imparting or eliciting justifications to be relevant this time:

\[ \text{(imp\_dis\_el\_conf\_p 46) (elicit\_justn\_p 0) (imp\_dis\_el\_justn\_p 0) (impart\_disagree\_p 41) (imp\_dis\_imp\_bel\_p 0) (imp\_dis\_imp\_elag\_p 0) (imp\_dis\_imp\_elj\_p 0)} \]

Again, the system decides to add nothing more to its response, and the next few utterances are:

system: I agree about social status being a reason for wanting a sunroof, but I still disagree about having a sunroof. Do you really want to include having a sunroof...
as an objective?

user: Yes, I do! And now I want to finish with objectives.

system: OK. So what do you think we should do next?

user: I think we should get values for the alternatives for those objectives.

The system disagrees with this proposition, and so the next transient goal to address is that of resolving this conflict. In this case the system knows of justifications against, but not for, the proposition, and can propose an alternative belief. With these values of the relevance functions, the utilities are:

$$((\text{imp\_dis\_i\_conf\_p 46}) \ (\text{elicit\_justn\_p 0}) \ (\text{imp\_dis\_el\_justn\_p 0}) \ (\text{imp\_dis\_agree\_p 41}) \ (\text{imp\_dis\_bel\_p 69}) \ (\text{imp\_dis\_el\_j\_p 59}) \ (\text{imp\_dis\_el\_j\_p 0}))$$

system: I disagree about getting values for the alternatives. I think we should identify parameters to measure objectives. Do you agree?

This discussion has given an outline of the way in which values and relevance conditions influence the system's choice of utterances when resolving a conflict. The decisions governing the choice of actions in all other situations follow the same pattern.

7.9 Discussion

Comparing the implementation with the theoretical design presented in §4.3, it can be seen that some very general descriptions have been implemented in specific ways. Some have not been implemented at all. So, for example, the matrix exists, but the user cannot manipulate it directly. Other aspects of the learning environment (such as the scaling device) have not been implemented at all - partly because only two aspects of the problem (what tactics to adopt and what objectives to agree) have been fully implemented; all others are performed without consultation by the agent. The structure of the matrix (with three columns of data of pre-defined type for each alternative) is only appropriate for the WOM, not for any other decision strategies, but since no other strategies have been fully implemented this has not been a problem. The dialogue component satisfies the desiderata as stated in §4.3.4. Nearly all the decision tactics identified in §4.2.8 can be discussed, though not all can be
executed. Two of the sources of information (problem specification and general knowledge about the problem) identified in §4.2.7 have been accommodated (in agree_ps_objectives and agree_lifestyle_objectives). The full functionality of the spreadsheet as discussed in §4.3.3 has not been implemented.

Many aspects of the implementation are somewhat *ad hoc*; some possible improvements have been outlined, and many areas for further research have been identified. The implementation is sufficiently well developed to demonstrate the possibilities of the approach and to be evaluated by educators, but there is much scope for improvement of the implementation, and also more fundamental research to be done - for example, on the representation of domain and problem solving knowledge, on extension of the agent's dialogue capability to engage in other topics of conversation, on negotiation of control over the environment and on reflection on existing attitudes (most notably beliefs). Additional shortcomings of the implementation are highlighted in the evaluation of the system, which is described in the next chapter.

In the context of the theoretical model, the implementation has served as a test-bed to validate the theory, and as such provides a starting point for developing a more principled design of those features which have been described in this chapter as *ad hoc*. 
Chapter 8: Empirical test of WOMBAT

8.1 Introduction

The questions which the empirical test of WOMBAT was designed to answer were firstly: is the dialogue coherent and sensible? and secondly: does the system have any real potential for use in design education? Given that the implementation was not sufficiently well developed for it to be usable by students as a part of their usual curriculum - and that an adequate evaluation of a full system would have been a major research project in itself - it was decided to do a small scale formative evaluation with engineering educators.

8.2 The design of the experiment

Each subject was asked to select between 6 (pre-defined) alternative cars. They were presented with a default problem which was to select the best car for a family; it was to be their only car. They were told that the family considered safety and comfort to be important, and could afford to spend up to £8000. As discussed elsewhere (§7.8.1), the problem definition is expressed as a list of keywords (which the system uses to make inferences about additional decision criteria not explicitly stated in the problem specification), a list of needs (absolute criteria which any solution must satisfy), and a list of wants (criteria which should be optimised). Although subjects were offered the chance to define a different problem if they preferred, none chose to do so.

As outlined in §7.9, the subjects had access to two aspects of the decision problem solving; the first of these was in the selection of decision making tactics. The system has information about 17 decision making and data organising tactics which are appropriate at different stages in the decision process (see §C.1), and the subjects were encouraged to discuss these with the system. The second aspect of the problem which they could discuss fully was the identification of the objectives on which the decision was to be based. The discussion of objectives was expected to take place in the context where the system and user had agreed to adopt the tactic referred to as 'agree_lifestyle_objectives' - namely to agree on objectives in addition to the ones explicitly stated in the problem specification. This tactic
could be formally terminated by agreeing to 'finish_tactic', at which point the system would expect to discuss further tactics. All other tactics were executed autonomously (without negotiating over the details) by WOMBAT, except that the user could elect to change the weighting values assigned by the system.

As this was a formative evaluation, the subjects were expected to comment on various aspects of the system design and to give their personal views on its potential for use in teaching. The experiment was intended to take about an hour.

For reasons explained below, both the software and the experimental setup were modified twice (as well as a few programming bugs being fixed as they were discovered). While some interesting point emerged in the course of interactions using the first two setups, most of these were not points which this study was designed to investigate, and it was felt that these setups did not facilitate the answering of the second of the original evaluation questions (namely, whether or not the system had any potential for use in design education).

8.2.1 Initial experimental setup

The program was set up on a MacII™ computer (which has a larger screen than the Mac SE/30 on which the software was originally developed) - sometimes one with a standard screen, and sometimes one with an A3 screen. The subject was asked to read through the introductory information (see §C.1), then to use the program, with the experimenter as an interface, and finally to answer some questions on it. The experimenter acted as an interface because it was believed that the formalised language was too difficult for subjects to pick up quickly. Audio recordings were made of most of the discussions, and a trace was kept of the computer interactions. Using the first version of the software, it was not possible to access the information the system had on alternative cars directly. Also, there was a technical difficulty such that while the dialogue input window was active it was not possible to access any information which was not already displayed on the screen. In this version of the software, subjects had to argue quite persistently before the system would accept the proposal of finishing agree_lifestyle_objectives if it believed that an insufficient number of objectives (about 8) had been agreed; as this feature was found to be an irritation
Design, Decisions and Dialogue

Chapter 8: Empirical test of WOMBAT

and a barrier by some subjects, it was modified between the second and third experimental setups. The first setup was used with 4 subjects.

### 8.2.2 The problems encountered in the first experimental setup

The intention, in this first setup, was that subjects would be able to express what they wanted to do as if to another human being, with the experimenter interpreting the ideas expressed into terms that WOMBAT could understand. In practice, some subjects started by proposing activities which WOMBAT could not deal with, for example:

J: OK, er, what shall we do? Well, yes, in real life what you do is you get the catalogues which these correspond to....

A: Yes.

J: Is that right? So we start looking at catalogues, so um....

Every subject using this setup expressed a desire to have access to detailed information on the possible alternative cars at an earlier stage in the interaction than was possible using the first version of the software. R expressed this as:

R: Only one of these is over 8000 pounds, isn't it?

A: Yes.

R: So I'd like to have a look at the rest of them.

M wanted more 'meta-level' information about what was possible:

M: I've never seen [this program] before, and my reaction at this stage, given that it's asking me what I want it to do, um, and I don't really have any idea, I want to say, well tell me what sorts of things I can do.

In addition, in contrast to the results of some previous research programs (see for example (Turkle 1984)), one subject expressed a need to understand how the system worked and what it knew before he felt able to interact with it:

M: I don't need to know everything it knows. I need some reassurance that it knows

---

1 In this and all following extracts from verbal protocols, 'A' is the experimenter - i.e. me!
something. And I wouldn't half mind a summary of the sorts of things it knows and what it knows about them, if you follow, so the fact that it has information on - in this instance, 6 cars...

A: yes

M: ...and that it holds information of this nature...

A: right

M: ...that sort of thing - I mean, you know, what... I can't even begin to formulate the kind of thing I'm interested in, but it's to do with summarising... summarising information.

In this first experimental setup, I (as experimenter) was trying to exert as little influence as possible over the course of the interaction, in order that subjects should express as freely as possible their own view of the problem and of how they would choose to solve it. In practice, this resulted in a conflict between my role as (passive) computer interface, my role as guide (helping subjects to overcome impasses without doing the work for them), my role as evaluator (requiring that subjects should obtain a balanced view of the system, discovering its strengths as well as its limitations) and my role as system designer (responding to questions such as "why are you doing this?", and providing background information about aspects such as why certain design decisions had been made). It was frequently unclear which 'hat I was wearing' at any given time.

To summarise, for the 4 subjects who used this setup there was an initial hurdle of learning what WOMBAT could and could not do, which distracted from the purpose of the evaluation. The last of these, M, had such major difficulty in getting started that the entire interaction was dominated by this experience, which resulted in his spending very little time actually interacting with WOMBAT, or getting familiar with the dialogue. A secondary problem was that there was a conflict between the requirements of allowing the subjects the freedom to express what they wanted to do with the system, guiding them as to what was in fact possible and making them aware of things which it could do which the subjects had not even considered.
8.2.3 Modified experimental setup

In order to speed progress over the initial hurdle, and to make it a bit clearer to subjects what knowledge WOMBAT had access to, changes were made to both the software and the experimental setup. The changes made to the software were firstly to make the information held on the alternative solutions (cars) and on tactics accessible to the user via a pull-down menu, to address the problem of the user not knowing what information the system has access to. The second change made to the software was to introduce a formal way of suspending execution of the program so that the user could access information in the learning environment. (The presence of the dialogue box on the screen prevented this as long as the program was running.)

The experimental setup was changed so that subjects were first given an introduction to the software (in the form of a guided tour around its menus and windows, followed by a demonstration of the sorts of interactions which are possible with it). In order to clarify my role, subjects were then invited to re-start the program, and to interact with it directly, with me interpreting (system output) and guiding (on the format of input) as necessary. This setup was used with 2 subjects.

8.2.4 Shortcomings of this experimental setup

The changes made from the first to the second experimental setup overcame most of the problems encountered with the first setup. However, new problems were introduced such that this setup did not yield an adequate evaluation of the educational potential of the system either. The first problem was that this second setup was unacceptably time-consuming, given that most subjects were very busy people, so their interaction with WOMBAT was rushed and consequently superficial. Also, subjects focused so much of their attention on trying to get the input correct that they failed to really think about the deeper system design, as they concentrated their attention on the surface features of the interface and on getting it right. For example:

C: the complexity for me of using this kind of interface interferes with my interaction with
the program quite severely.

The experimental setup was changed again to address these problems.

8.2.5 Final experimental setup

In the final experimental setup, subjects were again given a guided tour of the windows and pull-down menus and the early stages of the decision process were demonstrated (as in the second setup), then the subjects were invited to identify the objectives on which a decision was to be based (with me acting as an interface between the subject and WOMBAT). I then interacted with the system again for the steps of identifying parameters and getting parameter values (so that the subject had access to all the raw figures as the base for their decision making). The subjects were asked to describe how they would go on with the decision making from there, and the decision was made, again with me acting as an interface between the subject and WOMBAT. The main changes made to the software at this stage were to add a recalculate "fast route" (so that once a subject had articulated tactics once, they did not have to do so explicitly again), and to make it easier to finish_tactic and more difficult to end if the decision had been made with too few factors (to facilitate the 'rinse and repeat' decision strategy which several of the subjects had either adopted or expressed a desire to adopt).

3 of the 4 subjects using the final setup did so using a Mac SE/30, which has a small screen, so it was necessary to juggle windows around much more than for other subjects. As I was acting as the interface, this was not a significant problem, but it would have been a great barrier to usability for novice users who could not find their way around the windows.

8.2.6 Outcome from final experimental setup

The final setup was the least principled, in the sense that the experimenter had the greatest influence over the course of the interaction, but it was also the most effective in terms of answering the second evaluation question. 3 of the 4 subjects who used this setup explored it much more thoroughly than the first 6 subjects, because they had a better idea of what was possible than first setup subjects, and were less constrained by the requirements
of actually operating the interface themselves than the second setup subjects.

8.3 Example of an interaction

In this section a fairly detailed account is given of the interaction with one subject. This description provides a starting point for discussing the variety of experiences had by different subjects, and for highlighting the different points which emerged in the different interactions. The interaction with P (one of the subjects who used the final experimental setup, a senior lecturer in mechanical engineering from a traditional university) has been selected for description because it was a session in which various interesting issues arose, and in which WOMBAT was explored comparatively fully.

8.3.1 Description of the interaction

P used the final experimental setup. After being led through the initial steps, P wanted to proceed with just comfort and safety. Having seen the qualitative ratings for all 6 alternatives and noted that A and F were ahead (and which one was preferred depended on the weightings one gave these two objectives), he went back to define more objectives. He appeared to be modelling the problem on his daughter's family, assuming that the purchaser had 3 small children and a dog. Therefore the first two criteria he proposed were the number of seats and the number of doors. As the system does not currently have information on the number of seats, he spent a couple of minutes looking at the descriptions of the cars and assessing how many seats they were likely to have based on the available information. He wanted to be able to enter this information in the matrix (the current implementation does not support this). Like several other subjects, P wanted a hatchback (but was happy to accept the descriptor 'flexibility' as a measure of whether or not the car had a hatchback). By chance, at this point four of the objectives identified were measured qualitatively (on a scale of 1-10), one had no values, and two (4_doors and sunroof) were yes/no measures which were given numerically as 0 or 1. P was perturbed by the presence of 1's in the matrix. In his view, having 4 doors was more important than having a sunroof, and he seemed to be struggling to find a way of expressing this in the matrix. Similarly, he seemed unsettled by the fact that these two objectives had values of 0 or 1 whereas others were on a scale that ran
up to 10. It appeared that he was trying in his head to see what happened if he left the qualitative ratings as they were and upped the 1's to something on a scale of 0 to 10 to reflect their relative importance to him (e.g. changing all 1's for 4_doors to 4's and all 1's for sunroof to 2's), after which I believe that he would have simply added the numbers for each column. As he could not articulate clearly what he was thinking, this remains in the realm of speculation, but it was quite clear that he had not separated in his mind the roles of normalising and weighting - partly because none of his objectives were measured quantitatively, which might have forced the distinction sooner.

In his deliberations about how he would make a decision, he could talk through a comparison of A with F, from which he concluded that he preferred F to A, but his glancing across B, C and E left him viewing all three as much the same, and he was not sure how he would compare them with A and F. As I talked through with him the distinction between normalising and weighting, he then said that what he would do with weights and normalised values was to do the sum of a product.

After he had articulated all this, we used the system to do the calculations. He noted that C and F came out with very similar values, so he changed the weightings to see what the effect would be. He got a bit frustrated that each time he changed anything he lost the results of the previous calculation, so that he could not observe the effect of his changes directly. He then decided to include price as a new objective. At this point, he was happy with the solution (F) which he got. He discussed how he would generally test the sensitivity of the weightings, but felt happy to end at this point. He was amazed when the system argued with him about finishing, and asked it for a justification. Happy with its reason, he then referred to the pull-down menu listing known objectives (rather than doing the preferred action - from a dialogue point of view - which would have been to ask the system to suggest an objective), and realised that "It was right - I hadn't considered it deeply enough", as he recognised objectives in the list which were important to him (notably pollution and fuel_economy). Thus, the system's reluctance to finish was perceived as a positive point. When he saw the pollution data, his comment was that this was very important, and that he didn't want to get C (the only car which ran on leaded fuel).
would have liked to have been able to change the problem specification at this point to make running on lead-free petrol (preferably with a catalytic converter) an absolute criterion; as a second-best, he set the weighting for pollution very high, so that there was no possibility of C scoring the highest. The system at this point suggested finishing, which P wasn't ready to do as he wanted to change weightings again. He expressed the view that this was "an illogical point to end when we've just added two new objectives". Having changed the weightings, and got the same result (F was still the best), he then finished.

P was able to express what he wanted from it fairly assertively, entered into the spirit of the thing, and appeared to quite enjoy the experience. He spontaneously asked questions such as "Why does it think that?", and developed a reasonable sense of what the system was capable of.

8.3.2 The dialogue between P and WOMBAT

What follows is the full trace of the dialogue between P and WOMBAT (mediated by the experimenter), showing the course of the entire interaction. It is included here in its entirety, firstly to demonstrate that WOMBAT can sustain a prolonged dialogue with the user, secondly to illustrate that with one obvious exception (as noted below - §8.4.2) this dialogue remains coherent, and thirdly to illustrate the range of types of interaction which the current implementation is capable of engaging in. In this trace, the system's internal representation of each utterance (the formalised language) is presented in this font, and a natural language interpretation is presented in this font, with in-line comments in this font.

system: Elicit Believes Tactic

What do you believe we should do?

[this is a fairly abrupt opening, as discussed in §9]

user: Impart Believes Tactic Eliminate_Fail_Ps

I believe we should eliminate all alternatives which fail the absolute criteria stated in the problem specification

system: Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject Justification Price Impart Believes Tactic
Agree_Ps_Objectives Elicit Believes Tactic Agree_Ps_Objectives Agree

I consider us committed to doing that. So alternative D is rejected on the grounds of price. I suggest that next we should agree to take into account the objectives stated in the problem specification. Do you agree?

user: Elicit Believes Justification

Why do you suggest that?

system: Impart Believes Tactic Agree_Ps_Objectives Justification

Take_Problem_Spec_Into_Account Elicit Believes Tactic Agree_Ps_Objectives Agree

Because that way we take the problem specification into account. Do you agree that we should agree to take into account the objectives stated in the problem specification?

user: Impart Believes Agree

Yes!

system: Impart D_Committed Agree_Ps_Objectives Impart Believes Objectives Safety Justification In_Ps Impart Believes Objectives Comfort Justification In_Ps Elicit Believes Tactic

So we're committed to doing that. We agree that optimising safety and comfort are objectives because they're stated in the problem specification. What do you believe we should do next?

user: Impart Believes Tactic No_Ideas

I have no idea!

system: Impart Believes Tactic Matrix Elicit Believes Tactic Matrix Agree

I think we should set up a matrix of objectives against alternatives. Do you agree?

user: Elicit Believes Justification

Why do you suggest that?

system: Impart Believes Tactic Matrix Justification Organise_Data

Impart Believes Tactic Matrix Justification Organise_Data Justification

Keep_Clear_Head Elicit Believes Tactic Matrix Agree

Because it'll help us to organise the data, and that in turn will help to keep a clear head. Do
you agree that we should set up a matrix?

user: Impart Believes Agree

I agree.

[Matrix appears on the screen, showing that D has been rejected, and listing safety and comfort as objectives.]

system: Impart D_Committed Matrix Elicit Believes Tactic

Right, so we're committed to doing that. What do you believe we should do next?

user: Impart Believes Tactic Identify_Parameters

I think we should identify parameters to measure those objectives.

[The word 'Qual' appears against both safety and comfort to indicate that the system has qualitative ratings to measure both of these criteria.]

system: Impart D_Committed Identify_Parameters Impart Believes Tactic Get_Parameter_Values Elicit Believes Tactic Get_Parameter_Values Agree

OK. And next, I think we should get the values of those parameters for all the alternatives. Do you agree?

user: Impart Believes Agree

Yes, I agree.

[Matrix gets filled with numbers.]

system: Impart D_Committed Get_Parameter_Values Elicit Believes Tactic

Right, we've committed to doing that. What do you believe we should do next?

user: Impart Believes Tactic Agree_Lifestyle_Objectives

I think we should agree additional criteria based on the way the car's likely to be used.

system: Impart D_Committed Agree_Lifestyle_Objectives Elicit Believes Objectives

Right, we're committed to doing that. Suggest some objectives.

user: Impart Believes Objectives 5_Seats Impart Believes Objectives 4_Doors

It should ideally have 5 seats and 4 doors.

system: Impart Believes Objectives 4_Doors Agree Impart Believes Objectives 4_Doors Justification People_In_Back Impart Believes
Objectives 5_Seats Not-Known Elicit Believes Objectives 5_Seats

Confirmation

I agree about the 4 doors, because there'll be people in the back. I don't know anything about having 5 seats. Can you confirm to me that you believe having 5 seats should be an objective?

user: Impart Believes Confirmation Impart Believes Objectives Sunroof

Yes, I confirm that. I also think the car should have a sunroof.

system: Impart Believes Objectives Sunroof Disagree Impart Believes

Objectives Sunroof Not Justification Tend_To_Leak Elicit Believes

Objectives Sunroof Justification

I disagree about having a sunroof, because they tend to leak. Why do you think having a sunroof is important?

user: Impart Believes Justification Fun_In_Nice_Weather

Because they're fun in nice weather.

system: Impart Believes Objectives Sunroof Justification

Fun_In_Nice_Weather Agree Impart Believes Objectives Sunroof Disagree

Elicit Believes Objectives Sunroof Confirmation

I agree that the fact that they're fun in nice weather is a good reason for wanting a sunroof, but I still disagree that having one is an important objective. Can you confirm that you really want to include this as an objective?

user: Elicit Believes Objectives Sunroof Not Justification

Can you give me any more reasons for not wanting one?

system: Impart Believes Objectives Sunroof Not Justification No_Ideas

Impart Believes Objectives Sunroof Disagree Elicit Believes Objectives

Sunroof Confirmation

I don't have any more reasons for not wanting a sunroof, but I still disagree about including a sunroof as an objective. Do you really want to include it?

user: Impart Believes Confirmation Impart Believes Objectives

Flexibility

Yes, I do! I also think we should include flexibility as an objective.
[Note: flexibility is an alternative expression for being a hatchback in this context.]

**system:** Impart Believes Objectives Flexibility Disagree Impart Believes Objectives Flexibility Not Justification Unimportant_Most_Of_The_Time Elicit Believes Objectives Flexibility Justification

I disagree about flexibility because I think it's unimportant most of the time. Why do you want flexibility?

**user:** Impart Believes Justification Dog

Because they've got a dog!

**system:** Impart Believes Objectives Flexibility Justification Dog Not-Known Impart Believes Objectives Flexibility Disagree Elicit Believes Objectives Flexibility Confirmation

I don't know about dogs, and I still disagree about flexibility. Can you confirm that you want to include flexibility?

**user:** Impart Believes Confirmation

Yes, I do.

**system:** Elicit Believes Objectives

Can you suggest some more objectives?

**user:** Impart Believes Tactic Finish_Tactic

I think that's enough for now.

**system:** Impart D_Committed Finish_Tactic Impart Believes Tactic Identify_Parameters Elicit Believes Tactic Identify_Parameters Agree

Right, we'll go on to something else. I think we should identify parameters to measure all those objectives next. Do you agree?

**user:** Impart Believes Agree

I agree.

[By now, all the agreed objectives have appeared in the matrix. At this point, parameters to measure them all also appear.]

**system:** Impart D_Committed Identify_Parameters Elicit Believes Tactic

Right. What do you think we should do next?

**user:** Impart Believes Tactic Get_Parameter_Values
I think we should get values for all those parameters.

system: Impart D_Committed Get_Parameter_Values Elicit Believes Tactic

Done! And what do you think we should do next?

user: Impart Believes Tactic Scale_Parameters

I think we should agree a scaling system for all the parameters.

system: Impart D_Committed Scale_Parameters Elicit Believes Tactic

Right. And next?

user: Impart Believes Tactic Assign_Values

We should assign scaled values to all the parameters on that scaling system.

system: Impart D_Committed Assign_Values Impart Believes Tactic

Assign_Weights Elicit Believes Tactic Assign_Weights Agree

Done! And now, I think we should assign weights to the objectives. Do you agree?

user: Impart Believes Agree

I agree.

system: Impart D_Committed Assign_Weights Elicit Believes Tactic

Done. And what next?

user: Impart Believes Tactic Change_Weights

I want to change the weights.

[At present, the system sets the weights to default values, which the user can then change, rather than weights being negotiated. So the user defined the new values:]

Flexibility: 5, Sunroof: 3, 4_Doors: 7, 5_Seats: 0, Comfort: 6, Safety: 5

system: Impart D_Committed Change_Weights Elicit Believes Tactic

Done! What do you want to do next?

user: Impart Believes Tactic Do_Wom

Calculate the sum of the product of weights and scaled values.

system: Impart D_Committed Do_Wom Impart Believes Alternatives F Best

Elicit Believes Tactic

Done. Alternative F scores best. What do you believe we should do?

user: Impart Believes Tactic Change_Weights

I want to change the weights.
[Having reviewed all the figures in the matrix, the user defined the new weights:]

**Flexibility: 5, Sunroof: 3, 4_Doors: 3, 5_Seats: 0, Comfort: 6, Safety: 5**

**system:** Impart D_Committed Change_Weights Impart Believes Tactic Do_Wom

Elicit Believes Tactic Do_Wom Agree

Done. I think we should calculate the sum of a product next. Do you agree?

**user:** Impart Believes Agree

Yes.

**system:** Impart D_Committed Do_Wom Impart Believes Alternatives A Best

Elicit Believes Tactic

Done that, and now A scores best. What do you think we should do?

**user:** Impart Believes Tactic Agree_Lifestyle_Objectives

I think we should agree some more objectives.

**system:** Impart D_Committed Agree_Lifestyle_Objectives Elicit Believes Objectives

Fine. Name some objectives!

**user:** Impart Believes Objectives Price

Price.

**system:** Impart Believes Objectives Price Agree Impart Believes Objectives Price Justification Not_Rich Elicit Believes Objectives

I agree about price, because I believe they're not rich. Any more objectives?

**user:** Impart Believes Tactic Finish_Tactic

That's enough.

**system:** Impart D_Committed Finish_Tactic Impart Believes Tactic Identify_Parameters Elicit Believes Tactic

OK. I think we should identify parameters next. What do you think we should do?

**user:** Impart Believes Tactic Recalculate

I think we should recalculate.

**system:** Impart D_Committed Recalculate Impart Believes Alternatives F Best Elicit Believes Tactic

OK. Now F scores best. What do you think we should do?
user: Impart Believes Tactic End
I think we should finish now.

system: Impart Believes Tactic End Disagree Impart Believes Tactic End
Not Justification Better_To_Base_Decision_On_More_Factors Elicit Believes Tactic End Justification
I disagree, because I think it would be better to base the decision on more factors. Why do you want to finish?

user: Impart Believes Past Impart Believes Tactic
Agree_Lifestyle_Objectives
I've changed my mind. Let's agree some more objectives.

system: Impart D_Committed Agree_Lifestyle_Objectives Impart Believes Tactic End Not Disagree Impart Believes Tactic Continue_Tactic Elicit Believes Tactic Continue_Tactic Agree
Right, we'll do that. I disagree about not ending. I believe we should continue agreeing objectives. Do you agree?

user: Impart Believes Agree Impart Believes Objectives Pollution Impart Believes Objectives Fuel_Economy
I agree. I believe optimising pollution and fuel economy are important objectives

system: Impart D_Committed Continue_Tactic Impart Believes Objectives Pollution Agree Impart Believes Objectives Pollution Justification Minimise_Environmental_Damage Impart Believes Objectives Fuel_Economy Agree Elicit Believes Objectives Fuel_Economy Justification
Let's continue doing this. I agree about minimising pollution being an important objective because it helps to minimise environmental damage. I also agree about optimising fuel economy, but can you tell me why you believe this?

user: Impart Believes Justification Not_Rich
Because they're not rich.

I agree about their not being rich being a reason for considering fuel economy to be important, and I believe they're not rich because a maximum price was set in the problem specification. I think we've done enough of that (agreeing objectives) now. Do you agree?

user: Impart Believes Agree

Yes.

system: Impart D_Committed Finish_Tactic Elicit Believes Tactic

OK. What shall we do next?

user: Impart Believes Tactic Recalculate

Recalculate.

system: Impart D_Committed Recalculate Impart Believes Alternatives F Best Impart Believes Tactic End Elicit Believes Tactic End Agree

Done that. Now F scores best. I think we should finish now. Do you agree?

user: Impart Believes Disagree Impart Believes Tactic Change_Weights

I disagree. I want to change the weights.

Fuel_Economy: 2, Pollution: 7, Price: 7, Flexibility: 5, Sunroof: 3, 4_Doors: 3, 5_Sea: 0, Comfort: 6, Safety: 5

system: Impart D_Committed Change_Weights Impart Believes Tactic End

Not Disagree Elicit Believes Tactic End Not Confirmation

Done that. I disagree about not ending. Can you confirm that you don't want to end?

user: Impart Believes Confirmation Impart Believes Tactic Do_Wom

I confirm that. I want to calculate the sum of a product.

system: Impart D_Committed Do_Wom Impart Believes Alternatives F Best Elicit Believes Tactic

This time F scores best. What do you want to do now?

user: Impart Believes Tactic End

Now I want to end.

system: Impart D_Committed End

OK. Bye!
This has been a long trace, but it illustrates how the system negotiates with the user, how it deals with propositions which are unknown to it (for example, that having 5 seats should be an objective) and how it tracks open topics of conversation. Full traces from the other 9 subjects in this study are presented (without translation into English) in §C.3.

In the discussion in the following sections, subjects are referred to by initials. Profiles of the subjects are presented in §C.2.

8.4 Is the dialogue coherent and sensible?

Subjects did not on the whole comment on the quality of the dialogue at a coherence level. There are several possible explanations for this; firstly, it is possible (though unlikely) that the idiosyncrasies of the output format obscured any semantic incoherence. It is more probable that, because as I interpreted the system output I commented automatically on utterances which I considered incoherent, this obviated the need for the subject to do this; in this case, it must be concluded that the interactions were largely coherent. As noted above (§5.3.1), coherence is notable more by its absence that its presence, so first two examples of incoherent interactions which were caused by bugs in the software are presented, to provide a contrast with the interactions with the current implementation. The bugs in question were fixed after the interactions in which they were detected. In the following section (§8.4.2), types of incoherent (or unreasonable) interactions which can still occur in the final version of the software are presented and discussed. A trace of the complete interaction with one of the experimental subjects has been presented in §8.3; this demonstrates the overall level of coherence and sensibleness of an interaction. This section (§8.4) closes with a discussion of the issues of coherence as evident in the implementation of WOMBAT.

8.4.1 Incoherences caused by bugs which were subsequently corrected.

The first extract is taken from the interaction with L. In slightly anglicised form, the interaction can be expressed:

1) user: I believe we should finish doing the current tactic [agree_lifestyle_objectives].
2) **system:** I disagree that we should finish doing the current tactic. Can you confirm that you wish to finish doing the current tactic? Do you agree that we should finish doing this tactic?

3) **user:** I believe we agree.

4) **system:** I consider us committed to finishing the current tactic. I disagree that we should finish the current tactic. I believe this because there is no active tactic. Why do you believe we should finish the current tactic?

5) **user:** Because we've agreed enough factors.

6) **system:** I don't know about the reason 'enough factors' as a justification for finishing the current tactic, so I believe it's impossible. I disagree that we should finish the current tactic. Do you confirm that that's what you want to do?

7) **user:** Yes, I do!

In this case, the problem is caused by the system having misguided beliefs about when it is appropriate to elicit agreement for a proposition from the user (line 2), so that it gets into a position where it is trying to deal with the same issue (whether to finish the current tactic) twice in each utterance cycle. The relevant beliefs were modified in the implementation before any other subjects used the system.

The second extract is taken from the interaction with C, one of the two subjects who had 'hands on' access as part of the second experimental setup. In this case, the problem was that the system and the user had different (and incompatible) understandings of the English phrase "I don't know", the label on a dialogue button intended to convey the meaning that "I don't have knowledge about the proposition you've made". What for the user was a perfectly reasonable thing to say appeared to be incoherent to the system, and in trying to make sense of it, the system in its turn made incoherent utterances:

**system:** what do you believe we should do?

**user:** I believe I don't know.

**system:** I believe I don't know about not-known. Do you confirm this?
user: I believe I've changed my mind. I believe we should do no idea.

system: I disagree about not. I believe I don't know about not not-known. Can you confirm that not?

user: I confirm that.

system: What do you believe we should do?

The button labelled 'I don't know' was subsequently removed, on the principle that while there are many things that the user might say which the system has no knowledge of, it is extremely unlikely that the converse will arise. Therefore the presence of the button, which the user was never likely to require, simply caused confusion.

8.4.2 Features of the current system which lead to incoherent or unreasonable interactions

In the current implementation, unreasonable responses arise as a result of one of two classes of shortcomings in the system. The first of these is inadequacies in the implementation of the dialogue mechanism, and the second is inadequacies in the system's representation of problem solving knowledge.

*Unreasonable responses caused by features of the dialogue mechanism*

The first example of unconventional (!) reasoning emerges in the interaction with G:

system: I disagree about ending, because I believe we should base the decision on more factors. Why do you want to end?

user: Because I've run out of ideas

system: I don't know about running out of ideas being a reason for ending, so *I believe it's an impossible reason for ending*. Can you confirm that you wish to end?

This strange reasoning is caused by the system dealing inappropriately with a proposition which it does not know about. In this case, the (reasonable) belief of the system that any action which is unknown to it is impossible for it to perform has been extended to become the (unreasonable) belief that any justification that it does not know about for an
action is impossible to apply (whatever that might mean!). Another example of the same category of responses (dealing with justifications which are unknown for propositions which are known) is taken from the trace of the interaction with M, but in this case the original proposition related to an objective, rather than an action:

**system**: I agree that luggage capacity is important, but can you tell me why you believe it's important?

**user**: Because they have lots of children.

**system**: I don't know about lots of children being a reason for wanting luggage capacity. Can you confirm that you believe this?

Clearly, this reasoning is less unreasonable than that relating to actions, but it might nevertheless be argued that as long as they are agreed about luggage capacity being important, the system could be less rigorous in dealing with the justifications. Both of these lines of reasoning could be readily changed by making minor modifications to the preference mechanism. Also, as discussed in §9.5, one possible future development of the system is to make it possible to learn new lines of reasoning from the user.

C's approach of persistently agreeing with everything the system proposed, and ignoring most of the questions it asked, resulted in the following:

**system**: I consider us committed to agreeing lifestyle objectives. I believe an important objective is running_cost. What do you believe is an important objective?

**user**: I believe we agree.

**system**: I agree that running_cost is an important objective. I believe that because they are not_rich. What do you believe is an important objective?

**user**: I believe we agree.

**system**: I agree that the fact that they are not rich is a reason for including running_cost, but why do you believe that?

**user**: I believe that because they're not rich.

**system**: I don't know about their not being rich being a justification.
for their not being rich being a justification for having running cost as an objective. Can you confirm this?

user: I confirm it.

One of the problems here is that the system deals inappropriately with the user agreeing with it explicitly - it has been designed to expect implicit agreement, and some modification is clearly needed to the mechanism for dealing with explicit agreement. In the system's defence, it might be said that the user's utterances are not as appropriate as they might have been - presumably because the user's understanding of the system's utterances was incomplete, and because the user had not realised that implicit agreement was assumed.

All these classes of unreasonable responses reflect deficiencies in the implementation rather than fundamental flaws in the theoretical dialogue model, and as such are correctable without substantial further research.

Unreasonable responses caused by shortcomings in WOMBAT's representation of problem solving knowledge

The most frequent cause of unreasonable interactions is the inadequacy of WOMBAT's representation of the problem solving domain. This genre of interactions is exemplified by the following (taken from the interaction with M):

system: I consider us committed to get_parameter_values. I believe we should do agree_lifestyle_objectives next. Do you agree?
user: I believe we disagree. I believe we should do assign_weights.

system: I disagree about not doing agree_lifestyle_objectives. I also disagree about doing assign_weights. I disagree about doing this because I believe we should do agree_lifestyle_objectives first. Why do you believe we should do assign_weights?
user: I believe I've changed my mind.

system: I agree that we should not do assign_weights. I disagree that we should not do agree_lifestyle_objectives. Do you confirm that you don't want to do agree_lifestyle_objectives?
user: I believe I've changed my mind.

system: I consider us committed to doing agree_lifestyle_objectives.

What do you believe an objective is?

This strange interaction can be explained by the system not having an adequate representation of the relationship between different tactics. As discussed in §7.5.2, it does not know that committing to one tactic means that one cannot be simultaneously committed to another. It also fails to make the necessary inferences about one tactic being proposed in preference to another, so that when the user changes his mind, he is changing it about both propositions, and not just the one. (Thus the system argues for each proposition separately, rather than taking them as a linked pair.)

A slightly more bizarre exchange took place in the course of the interaction with P, caused by WOMBAT having an inadequate knowledge about reasons for ending, and under what circumstances such reasons apply:

user: I think we should end now.

system: I disagree about ending, because I think it would be better to base the decision on more factors. Why do you want to finish?

user: I've changed my mind. Let's agree some more objectives.

system: Right, we'll do that. I disagree about not ending. I believe we should continue agreeing objectives. Do you agree?

A less incoherent, but still somewhat unreasonable, exchange emerges in this extract from the interaction with J:

user: I believe we should do do_wom.

system: I disagree about doing do_wom, because I believe we should assign_weights first. Why do you believe we should do do_wom?

user: I've changed my mind. I believe we should do assign_weights.

system: I agree that we should not do do_wom. I consider us committed to doing assign_weights. What do you believe we should do next?

user: Now I believe we should do do_wom.
To overcome these three classes of unreasonable responses requires that the system should have a greatly improved representation of the problem solving domain and ability to make inferences. The developments needed are discussed in more detail below (§8.5.3).

8.4.3 Summary

As demonstrated by the traces of the interaction with Subject P (§8.3) and with the other nine subjects in this study (§C.3), WOMBAT can participate in a sustained dialogue with a user within its domain of expertise. That dialogue remains largely coherent and reasonable. It can deal appropriately with both known and unknown propositions as long as they are expressed in the appropriate form. There are a few situations in which the system response is sub-optimal, either because of shortcomings in the implementation (e.g. in the routine to understand explicit agreement) or because of shortcomings in the system's representation of problem solving knowledge. As highlighted by the incoherence in the interaction with L (§8.4.1), caused by a bug which was subsequently corrected, the maintenance of coherence is dependent on the quality of the dialogue reasoning, which is to a large extent 'engineered' in the system's ability to identify reasonable responses in any given situation. However, results indicate that the basic framework provides a reasonable starting point for building a more competent dialogue agent.

8.5 Does WOMBAT have any potential for use in design education?

In this section consideration is given to the ways in which the system was used by the experimental subjects and to the reactions of those subjects to the system (both verbal, in the course of the interaction, and written, in the form of answers on the questionnaires).

8.5.1 Ways WOMBAT was used

As discussed above, the final experimental setup gave the most satisfactory results, in terms of subjects being able to establish the strengths as well as the weaknesses of the system. Each subject had a different experience with the system, because of individual differences in approach and background, and because of the different experimental setups.
which were used. Having described the experience of one subject is some detail (§8.3.1), what follows is a synthesis of specific points which arose in the course of all the interactions, along with discussions of how some of the problems detected might be alleviated.

**The problem**

No-one elected to use anything other than the default problem definition (to select a car for a family; it is to be their only car; they can spend no more than £8000, and consider comfort and safety to be important).

Two subjects (K and G) expressed unease about the particular problem (selection of a car) used in the prototype. The principal reason for this is that people have generally got pre-conceived ideas about cars and stereotypical users (certainly about real ones - e.g. G expressed the view that lawyers like Volvos, yuppies prefer Porsches, and Sloane Rangers choose Ford Escorts!). K noted that he asks his students to do an evaluation exercise using 3-pin plugs, which are passed around the room, and about which students have fewer prejudices. In contrast, J was very happy with the problem chosen, as it was one with which he could identify.

When used during a design exercise, the set problem would be that on which the students are working. Within the WOMBAT implementation, beliefs relating to the specific problem are localised in 2 places: the file containing possible beliefs the system might hold about the object (as part of the reasoning mechanism), and the data file containing information on the possible alternatives. Therefore, it is an easy task to replace this with beliefs relating to a different problem (though it might be less easy to construct the reasoning beliefs which are relevant to a different problem).

**The problem formulation**

I had to explain in some detail the way in which the problem was formulated; some
subjects had difficulty identifying with this formulation of the problem. For example, when the system proposed minimising price as an objective (where in the problem definition it was stated that the maximum acceptable price was £8000), the discussion with J continued:

A: ...it believes an objective is price, because it believes they're not rich.
J: Well that's absolutely right. I thought we had price in there already, but...
A: Um, well no, you've used price as an absolute criterion here, on rejecting D, but you haven't actually got it in there as an objective.
J: Oh, OK: price.
A: Right, OK, so you agree about that.
J: Oh, I see, so it's not, the fact that we can afford 8000 - that we're not out to spend exactly 8000, but we're happy to spend less if we can.

As well as subjects not always appreciating at first the difference between absolute and relative criteria, several expressed the desire to 'bend' the absolute ones, for example, regarding the rejection of alternative D (which cost over £8000):

B: Supposing at this stage you said well really it's so near to the limit that one might say - well I think we could - you know, could you adjust - if someone said because it's only a hundred - it's so nearly...

There are two obvious ways to change WOMBAT to facilitate users' understanding of the problem definition; one is to initialise the system without a problem specification at all and start by negotiating it so that users have been more explicitly involved with the development of the problem definition. The alternative is to get the system to articulate more clearly its understanding of the given problem definition. As was explained in §2.2, the system is not meant to be dealing with the issue of defining the problem definition, and neither is it really meant to be used in isolation (divorced from the design process), so the latter is probably the more appropriate route.

One subject (C), was quite happy to accept the rejection of D initially, and would have been happy to have it removed from the matrix. However, he noted at the end that on his decision criteria, D was substantially better than all other alternatives, and that at this
point he might wish to review his problem definition so that D (which was by far the best solution in his case) was not rejected. This is a clear argument in favour of the approach of displaying information about all alternatives, even those which have been rejected.

Expanding on the notion of the 'lifestyle' of the purchaser, one subject (B) wanted much more detail about the way the car would be used, in terms of annual mileage and whether or not the driver had a 'heavy foot'. Clearly, this level of modelling might be necessary within the design facility of a car manufacturer, but it is not clear that it is a high priority in the current context. It might be possible to cater for this level of detailed purchaser modelling using techniques from hypertext in the development of the design environment, as has been done, for example, by McCall (1989); indeed, it seems likely that any substantial development of the system would have to exploit features traditionally associated with hypertext systems (with all the attendant problems of 'getting lost in hyperspace') but such issues are beyond the scope of the current research.

*View of the dialogue*

Only one subject (M) expressed real exasperation with the dialogue:

M: I think.. it's like one of those wretched computer programs that when you say 'quit' it comes back with 'are you sure?' and you say 'yes' and it says 'do you really want to?'

Two subjects expressed pleasant surprise at utterances made by the system (P when the system disagreed about finishing, and K when the system suggested that they should agree additional objectives based on the likely lifestyle of the user [agree_lifestyle_objectives]).

Only one subject (K) said things which implied a sense of 'relationship' with WOMBAT. For example, when the system responded that it did not know about windsurfing as a justification for requiring flexibility:

A: It doesn't know about windsurfing.

---

L: Educationalist; R: Research Student; J: RF, Design; M: L, Materials; D: L, Aero Eng;  
C: Programmer; B: Prof, Mech Eng; P: SL, Mech Eng; M: L, Materials; D: L, Aero Eng;  
G: Prof, Eng Des.
K: It doesn't? We're going to fall out! Windsurfing is very close to the meaning of life.

Two subjects (R and C) wanted to make explicit their agreement with system propositions. As discussed above, at the moment this leads to somewhat tortuous interactions as the system does not deal with explicit agreement as elegantly as it might.

One subject (G) viewed the system's ability to accept propositions which it does not know about ambivalently. At one point he asked:

Can we have something it'll understand rather than something it won't?

But later he commented:

Even if you say things it can't understand, at least it accepts it and deals with it.

Otherwise, reaction to the dialogue in the course of the interaction was generally neutral.

Several subjects asked the system about tactics (or sought justifications for tactics suggested); few asked for the system's view of objectives. Subjects were also more ready to argue with the system about objectives; clearly, there is a difference between subjects' level of confidence in their knowledge about objectives and tactics, which is reflected in the ways they used the dialogue. On most objectives, where there was disagreement, subjects would argue their case and impose their will, for example:

D: [reading from the screen] 'I believe an objective is acceleration. ... What do you believe an objective is?' What if I say no, I think that would go against - using it would go against the running cost - if you have it there you would use it. Supposing I don't want that..?
A: 'I believe we disagree'.

There is the occasional exception. For example, in the interaction with K the system says that it disagrees that colour is an important objective because there are various colours available:

K: It's given a superior argument. It's right. If you're buying a new car you can generally get the colour you want, so it's not a problem. I agree that colour is not important at this stage in the decision making.
**Terminology**

Several subjects found the terminology alien to them. As the system was not meant to be evaluated at this level, little effort had been put into this aspect of the system implementation (even less than had been put into making the system's idiosyncratic sentence construction comprehensible!). However, it is worth noting that subjects did not want to explicitly say 'I believe' at all (G), or preferred to say 'I think' rather than 'I believe' (K). K also prefers the terms 'demands' and 'wishes' in place of 'needs' and 'wants'.

B commented on the use of the term 'tactic'. The notion of having to explicitly finish a tactic was quite alien to most subjects.

Two of these aspects of the terminology ('believes' and 'tactics') reflect the architecture of the system, and its origins in agent theory; for the engineering educator, these aspects of the system design are irrelevant, and they should be hidden from the user.

**View of the learning environment**

M was unhappy that the matrix only accommodated 6 possible solutions; in his experience (coming from materials selection) one might start with a huge number of possible solutions which gradually get whittled down to a manageable number. Though a valid point in many teaching situations, this is not the target scenario for use of WOMBAT, in which it is assumed that the designs being evaluated are the limited number of design concepts generated by the students. Other comments about the learning environment appear in the answers to question (4) as outlined below.

**Problem solving strategies**

In the prototype implementation, there is only one complete route through to a decision (the Weighted Objectives route - see §2.2.3), and all subjects took this route. Two subjects (J and K) expressed a desire to simplify the problem by specifying some absolute
criterion which would have eliminated some alternatives (e.g. that the car must not have a sunroof, or must have 4 doors) before setting off down this route:

K: Well there are going to be some things that rule an option out completely - I mean if one absolutely insists that a car runs on lead-free petrol and it doesn't there's no point in considering that car further - so essentially we check against the demands. Then you tend to assess on the wishes or the wants as to which car provides you with the maximum number of wants, so the first thing one would tend to look at is absolute - if you insist on 4 doors and a car hasn't got 4 doors, that's it. You can compromise, you can always change your mind, but you have to do that at the start to reduce your number of options.

However, no subjects articulated any other decision tactics which were not on the preferred route. This is in contrast to the results of the first protocol study (§4.2), so it would appear that the subjects in this second study were much more constrained in their thinking than those in the first. Factors influencing this result probably include the differences in the experimental setup, the fact that a computer is involved at all, the nature of WOMBAT's learning environment, and the fact that subjects were being asked this time to articulate their decision tactics explicitly. A more detailed study of the effect of the presence of WOMBAT or any other computer based decision aid on people's decision making strategies would be interesting to pursue, but is beyond the scope of this research.

Several subjects (including P, as described above) either adopted, or expressed a desire to adopt, a strategy referred to here as 'rinse and repeat'. This involved executing all decision steps to get an overall utility score for each alternative, then modifying the decision criteria or the weightings and repeating the calculation until they were happy with the solution. This appeared to serve two functions. The first function was to help subjects learn more about the problem, in terms of starting to get a feel for the possible solutions and for what criteria might be important to them (the same phenomenon could be observed in the initial protocol study (§4.2.3)). The second function of this was to assess the sensitivity of their solution to changes in the objectives and weightings. For example, K noted that his solution was dependent on the weighting he gave to the objective of having 4 doors; if 4_doors was given a very low weighting then alternative A scored best, whereas if it was
given a higher weighting (consistent with his desire to eliminate all alternatives which did not have 4 doors) then the preferred solution was F.

Consistent with his approach of requiring information about the possible alternatives early on, K advocated an additional decision step to precede all those known to WOMBAT:

K: When we're doing evaluation techniques, I recommend to my students that they make their first evaluation, just 'I want that one'. There's a reason behind all this. All the theories say that you shouldn't form a view - that you shouldn't have a preliminary design in mind, that you should have an open mind, but my hunch is that the human being doesn't operate like that, and therefore to go through this pretence of saying 'I haven't really got a solution in mind, I'm just considering the functions and requirements' - it just makes a farce. And one of the best ways of confronting this is to say 'well yes I do, and this is it - and I'm now going to go through the exercise and check that it really is the best one'. But it's a question of bringing it out into the open. By doing that I think one is more likely to say, 'well it wasn't the right decision' rather than not facing up to the fact that you've got that in mind. I mean, when you're choosing a car the chances are you've made a subjective decision for all sorts of complex reasons, and then you try to sit down and think about it. To pretend that you're just doing it rationally, I just don't think it's the way the human operates.

In marked contrast to this, the last subject (G) expressed puzzlement as to why he might want to know values for the alternatives before he had agreed all criteria with the system - to him it seemed obvious that he should consider his decision criteria in the abstract. Having reached a decision based on just 5 criteria, G was happy with his first solution and did not wish to add to or change the weightings of his criteria in any way. However, the reason he gave for wishing to finish after his first solution was that he could not think of any more criteria; a more sophisticated system might have picked up on this reason and suggested that they discuss the matter further.

Another point which emerged in the course of the interaction with G is that students
might have 'buggy' strategies, which involve using tactics inappropriately - e.g. failing to
normalise parameters before calculating the sum of a product - and that the system should be
able to remediate in such cases. One example in the interaction with P (§8.3.1) was the
confusion between the roles of normalising and weighting; in this particular case, a possible
approach to remediation would have been for the system to introduce an objective which is
measured quantitatively to the discussion.

**Interaction strategies**

Most subjects had difficulty in articulating clearly tactics at the level of detail encoded
in WOMBAT; there were allusions to 'normalising' parameters, to 'prioritising' objectives
and to 'getting a valuation' for each alternative, or simply to 'getting the answer' (the
assumption being that WOMBAT knew how to do the calculation, so why was the user
being asked what to do?!).

C: Would you not generally put that data into the database in such a way that it was
normalised? Presumably that's what we're trying to do. We're trying to get that data in such
a way that we can make comparisons between the data?

A: Yes - that is obviously obvious to you. It's not necessarily obvious to everybody. Yes,
we're having to go through several steps to get that data normalised.

C: Is that because... is the idea that someone would have access to the database in the
end to put in their own information?

A: Eventually, yes.

C: So the point there is that they wouldn't need to know how to normalise the data to put it
into the database in the first place.

A: That's right, yes.

And again:

D: Why do you have to tell it to do that? Why can't that... It needs that, obviously, for the
program to work, it has to have that. And if you don't do that then it will tell you to scale the
parameters, presumably, because it needs it - so you have to do it. But you want the
person using it to be in control.
Three of the first 6 subjects used interaction strategies which enabled them to function with WOMBAT's level of detail of tactics; L selected tactics from the printed sheet which described the available tactics; D referred to the 'Tactics hints' window on the screen regularly, and C simply responded with 'I believe we should do no idea' every time the system asked 'What do you believe we should do?'; and then agreed with the system's suggestion. (As the final experimental setup did not require subjects to operate at the system's level of detail, their results are not relevant in this context.)

False expectations and misunderstandings

As outlined above, several subjects expected the system to do more of the work for them than it actually does, e.g.:

A: and it's asking what you think you should do next...
R: well I think it should sum the matrix then - evaluate the matrix.
A: How are you going to evaluate it?
R: I don't know - I thought it could do that. I thought it was like a spreadsheet.

This was true not just for tactics, but also for objectives:

R: I'd like to know what it knows about - what ones it thinks are possible objectives.
A: So 'What do you believe an objective is?'
R: mmhmmm
A: OK, so it's telling you that it agrees about fuel economy, because it helps to minimise environmental damage. It also thinks that overall running cost is important, and is asking what else you think is important.
R: So that's its full list there?

In addition, one subject (D) had great difficulty understanding the difference between the tactics agree_ps_objectives (agree to include the objectives stated in the problem specification) and agree_lifestyle_objectives (agree to agree on additional objectives based...
on the likely lifestyle of the purchaser). D took some time to understand what was intended by the term 'lifestyle objectives' at all - clearly another terminological difficulty, associated with a conceptual difficulty in understanding the decision tactics in question and what they involve. This problem would probably be alleviated somewhat by improving the system's explanatory powers.

Other points

Several subjects wished to be able to enter values of parameters for objectives which the system did not know about (such as range and number of seats), and seemed slightly frustrated that this was not catered for. In the short term, it should be possible to add a facility in WOMBAT for users to enter values for objectives not known to the system; in the longer term, it would be desirable for WOMBAT to be able to learn from users about objectives and justifications, so that its database of possible lines of argument (its knowledge about aspects of car design) expanded. (This would also make it easier to extend it to discuss the design of different objects.)

G was slightly bemused by the fact that the entities referred to as 'qualitative ratings' appeared as numerical quantities, and was unsure whether or not they could be used as numerical quantities in calculations. Of course, he was right to be bemused; these entities are more accurately described as 'quantitative measures of qualitative ratings'. K advocated a smaller range than 0-10; he suggested 1-5. This would have the advantage that an unambiguous set of descriptors such as 'poor', 'below average', 'average', 'good' and 'excellent' could be applied with a one-to-one correspondence to the numerical values 1-5. This is an aspect of the problem which has not been dwelt on in the current implementation, but there is in principle no reason why it could not be the subject of negotiation, just like many other aspects of the design of the decision making environment. Indeed, it is possible that the design of the decision making environment could be the subject of a discussion in much the same way as the design of the car (or other object) is.
Robustness of system.

In the course of the evaluation study, the software crashed twice. In the first instance, this was caused by the subject (C) entering a 'null' proposition in the 'free entry' dialogue box; this bug was fixed. In the second instance, it occurred while the subject (K) was specifying new weights for the objectives; this crash could not be reproduced, so it has not been possible to establish the cause. Whatever the cause of the second failure, the software is clearly fairly robust. The situations in which it is known to become less-than-coherent are presented above, and it could reasonably be said to degrade gracefully rather than being brittle (a point made against many extant expert systems (cf (Forbus 1988), §2.1.3)).

Perceived context of this work

D described in some detail how he could envisage a tool like WOMBAT fitting in to the aeronautical design course which he helps teach:

In the final year ... we have this aircraft design course ... All they use as far as computers are concerned is just this basic spreadsheet for the design. ... The initial thing they do is to select a general design layout - what it's going to be, how it's going to be.... then they get down to the detail design of that particular aircraft. So at this stage... it's at this stage here that this would be useful. It could be useful to them to find out what... particularly the crunch comes in assigning weights to these different aspects that they control.... I imagine it would produce some interesting answers.

... In the decision making process they spend a lot of time sitting round a table arguing the toss. Now this sort of thing would give them some quantitative criteria to say well this is going to be... and what they could argue about there is to boil it down to the fact that this is what these weightings are -how they should be distributed. The discussion then would be focused on... it would really structure their thinking, their deliberations, and they would come to a conclusion much more quickly than perhaps they do at the moment. ... but if it
can bring out as well ramifications of bad decisions, that's another important aspect. Whereas they might say 'well I want to do this because I think it's fun and it should be done', and they concentrate on a particular advantage, and want to do it for that reason, and forget the disadvantages. What they need to be informed of - and I think this could do it - are what are the advantages and disadvantages of this... and even when they might apply a very heavy weighting to their favourite objective, nevertheless other things come up and say this is not going to be a good idea, and bring that home to them. The evidence is there, but it needs to be brought home. And if the computer says it, it'll carry quite a bit of weight - even if it's only their information which has gone into it!

Viewing the question more in the context of design practice than design education, G commented:

I have noticed recently that major improvements in the design process have been achieved precisely by quantifying in some way values about the parameters and handling this correctly. It means that although this is not a big part of design it could make a big improvement to the effectiveness of design if done properly.

...This could be applied right from the beginning when discussing, or rating customer preferences and then transforming that into a set of concepts, rather than just selecting, if you see what I mean, one out of three, which you could do once you've got all this software. That would make your work (a) much more widely applicable, and (b) extremely timely, because it's just now that all these quantitative ways for conceptual design are being introduced.

These two subjects, having used WOMBAT, and therefore commenting on the implementation and not simply on the concept, perceive a ready application for such a system in both education and industry. Both discussions focus on the learning environment rather than the dialogue component, viewing it as a tool to be used by collaborating users. It is my view that the learning environment (and in particular the data manipulation component of the environment) has the potential to be a useful tool, as envisaged by these subjects, in situations where there are collaborating users and where those users are conversant with the decision making techniques embodied in the system. The role of the dialogue component is
Design, Decisions and Dialogue

firstly to support users learning about the decision making techniques, and secondly to support users working alone (for example, in a distance education situation).

Summary

There was a fairly broad consensus, echoed in the responses to the questionnaire (see particularly the answers to questions 7 and 8 below and §C.4), that although it is a small part of the design process the issue of evaluation is important, and that this seemed a reasonable way of dealing with it.

Only one subject (M) had real difficulty interacting with the system. Other subjects had differing views on which aspects of the system design were informative, restrictive, annoying or helpful. The range of views are represented in the responses to the questionnaire.

8.5.2 Summary of responses on questionnaires

Full responses to the questions are included in §C.4. What follows is a summary.

1) The experience: has it been tortuous, tedious, tremendous, OK or what?

There were a range of responses to this - from 'interesting and enlightening' to 'confusing' and 'slightly tedious'.

Additional comments were that it was important to use a familiar example, that the current interface was limiting and difficult to use, that the terminology (e.g. use of 'believe' rather than 'think') nigged, and that it might have been better to have on-screen editing as the dialogue would become tedious after a time.

L: Educationalist; R: Research Student; J: RF, Design; M: L, Materials;
C: Programmer; B: Prof, Mech Eng; P: SL, Mech Eng; K: SL, Mech Eng;
D: L, Aero Eng; G: Prof, Eng Des.
2) The dialogue: was it quirky, sensible, helpful, useless, confusing, flexible...?

Ignoring the strange sentence construction, did any of the system's utterances strike you as surprising in any way? If so, how?

The idiosyncratic syntax was a barrier to subjects' understanding of system output, but the dialogue was generally considered to be moderately helpful. No one commented about utterances being surprising, except for P, whose comment was: "I was surprised that the system insisted (correctly) on having more parameters. It got slightly muddled about ending." No one commented about utterances being incoherent (but see §8.4 above).

3) Guidance and control: did you feel that the system allowed you the degree of control you wanted over the interaction? Did it provide appropriate guidance when you required it?

Again, responses were fairly mixed, reflecting the sorts of experiences that subjects had. Several subjects referred to the 'Ann interface' (which they might have been happier with than the system interface!). Several subjects expressed a feeling of not being in control because of the system's rather rigid formulation of tactics, but others expressed satisfaction. Assessment of guidance varied from 'fine' and 'appropriate' to 'rudimentary'.

4) The learning environment (i.e. matrix and pull-down menus): did you have access to the information you wanted? Was it readily available? Was there information you wanted to see but couldn't? Any suggestions for improvement?

Response from subjects who used the first version of the software was a clear 'No' - that they wanted to be able to see data on the alternatives before it appeared in the matrix (some way through the decision process). Responses from later subjects indicated a higher level of satisfaction. There were several suggestions for how to improve information presentation, including:

- "Would like to be able to see three columns for each alternative, also left hand columns etc. for all alternatives at once."
- "There should be 'meta-information' about what it knows about."
• "Some kind of running summary of what steps you've already carried out would be helpful."
• "I would have liked to have added further factors which are not available (included) at present."
• "Previous values of weightings should be visible when entering new values. Also, after the weighting calculation previous results were lost when proceeding to change parameters etc. These should continue to be available."

5) **The approach of having both learning environment and dialogue component: does it or does it not have any pedagogical advantages over either a simple learning environment (e.g. a commercial spreadsheet) or a stand-alone dialogue component? Please consider potential advantages of all three possibilities - and of any other configurations you can think of.**

The response to this was an overwhelming 'Yes' - the combination has clear pedagogical advantages. Comments included that it was important "especially when the system prompts for information the user did not think of themself (but agreed was important)", that "the user can be guided to a more detailed and deeper analysis than he might have undertaken on an alternative system.", and that "a spreadsheet should be invisible. Dialogue on its own would be little better than a book. You need to bring them together."

One subject felt that while the presence of the dialogue component should help to accelerate the learning process, "the simple learning environment is valuable as it is without preconceived controls and allows students to develop ideas freely"

Another commented that "The dialogue environment is essential for naive users. However, you can also imagine the situation where the problem is clearly defined or the user is familiar with the problem area. Here the dialogue content becomes less useful and it is desirable to 'skip over' it. I like the idea of a combined system which is dynamically 'tuned'
to the needs of the user - dialogue driven, learning environment driven, or both."

Yet another commented that "It felt like learning about choosing a car, not invited to reflect or focus attention on ordering or type of tactics, or which included etc. Needs debriefing. Higher, meta-level of dialogue about dialogue, or set task to summarise what happened using printout of trace." So she concluded that another configuration would be to add dialogue about the trace.

6) The approach of making explicit the tactics adopted / decision steps taken: do you believe that this has any pedagogical value?

The consensus view was 'Yes'. One subject explained this by adding "a structured way of thinking things through as in planning is very desirable".

7) Potential for use in education: Does the prototype have any? If so, what do you consider are its potential strengths and weaknesses? In what ways do you believe it needs to be modified or extended to have any educational value?

The view may be summarised as 'Yes, it has potential, but that potential will only be realised with a lot of work!'. Comments included "Obviously the interface requires considerable improvement.", "The dialogue needs to be cleaned up.... I am not sure how well an unguided user would cope with complexity of screens, menus etc." and "General strengths are that computer tends to maintain interest longer, particularly if it's quick and flexible. But we underestimate the difference between a prototype and a final system."

Suggestions for extension were:

- "Need to be able to look back to earlier stages of assessment" - to be able to see more clearly the effect of decision steps taken, and
- "I think flexibility and the ability to explore 'what if' situations provide the educational potential."
- "Its main strength is that it gives the students a structure within which to work at the early stages. There would have to be, however, means for allowing the student to take greater control as confidence is gained."
One subject noted that the approach could be extended to other determinable and closed domains.

8) In particular, do you believe this approach has any potential in the declared context of design education?

With one emphatic exception ("NO!" from subject M, who had a very negative experience with the system), the overall view was 'Yes'. Reasons given included:

- "Good for learning about processes, procedures."
- "It makes criteria and constraints explicit. Usually they are not, and design students often work with many implicit criteria and values."
- "It illustrates information elicitation (designers have to do this)."
- "Students will be stimulated to consider this kind of system as a tool
  (a) at the design stage (especially early on);
  (b) at the retail end to make sure customers understand the relative advantages of the product."
- "Evaluation: this system normalises and adds numbers together to get a score.
  Methodologically this is very dubious since it is based on chalk and cheese arithmetic. Good! Make students think about these things. How are trade-offs made?"
- "It makes the user think more deeply about the design specification, about the factors that really matter - also about the sensitivities of these factors."
- "Design means beginning with a blank sheet of paper, a most intimidating experience for all students. ...If the initial problem can be overcome by means of this type of program, and I [Subject D] think it can, then it will be extremely useful."
- "Design is an iterative decision making process."
9) *If the answer to (8) was broadly affirmative, then which skills do you see it as having the potential to develop in students?*

The answers to this question could be broadly summarised as identification of important objectives and improving awareness of the decision making process.

**Summary**

Subjects perceived the dialogue as being coherent (if quirky and niggling). WOMBAT is considered by the subjects to have educational potential, but that potential is far from being realised. There are three principal reasons for this:

- the interface is inadequate and not sufficiently well developed in various respects (including the presentation of system output, the mechanism to support user input (with the conflicting requirements that it be easy to use yet flexible and powerful), control of the learning environment and the display of information in the learning environment).
- the problem solving representation is impoverished, and in need of substantive development.
- the user model is inadequate; in particular, it should be possible for the system to intervene less as the student's competence and confidence increase (which requires that the system be able to form some assessment of the student's competence and confidence).

**8.5.3 Discussion of ways in which the prototype should be improved**

One of the points which has come through most clearly in this study is the wide variety of views on the subject of WOMBAT and design evaluation. G is very much in favour of quantitative methods in design evaluation; J views it as 'chalk and cheese arithmetic'. M cannot interact with the system until he knows what it knows; K chastises it for not knowing about windsurfing. P investigates several possible sets of objectives and weightings; G is happy to make a decision in a single pass. Some of the ways in which the prototype should be improved in order to have pedagogical value (as opposed to simply potential) have been alluded to already. Some of these points are expanded on here, and some further points are made.
Correcting and extending the dialogue mechanism

The system's understanding of explicit agreement needs improving. So does its ability to make inferences (from what the user said to what they actually meant. For example, the system should be able to infer finish_tactic through the user proposing a new tactic) and to engage in remedial dialogue (e.g. if the user's statement conflict with something said earlier).

In addition, although this point did not emerge as being important in the empirical study, the system should be able to articulate beliefs it already holds. (As it is currently implemented, WOMBAT only articulates new beliefs in the utterance cycle in which it adopts them, rather than routinely reflecting on its existing beliefs and using them in the dialogue.)

Developments in negotiation of control

At present, the system has total control over the learning environment. Much work needs to be done to establish explicit and implicit negotiation of control over the environment. The most obvious area for development here is in the area of weightings; clearly the present solution to the issue of defining and changing weightings is an unprincipled improvisation, but what form a principled solution should take is less clear. The interactive negotiation which takes place for objectives is not the best way forward for weights - it would probably involve a process of negotiating who does it and then negotiating fine tuning starting from that base position. What this requires is a principled framework for encapsulating that-which-is-being-negotiated-about (control (who does it), detail (what they agree about), commitment (what they agree to do), etc.).

Incidentally, while this did not seem to concern any of the experimental subjects, there is an important theoretical issue about how commitments are formed; the current approach of "I believe we should do X and you believe that we should do X, therefore we
are committed to doing X" seems to be operationally adequate, but fails to explain how the commitment is really established.

**Improving the interface**

To be usable by students, the presentation of the dialogue needs to be improved for both input and output. For output, this would simply involve a level of post-processing which it should not be too difficult to implement. For input, it is likely that the best approach would be to modify the dialogue component so that it actually set up an expectation of what reasonable inputs might be; these would then be displayed on the screen in some easily selectable form (but retaining the facility for users to express their own ideas in addition through 'free entry'). Establishing a suitable form which allows the user an adequate degree of freedom within the interaction while still making it easier for them to construct appropriate utterances than is currently the case is a topic for further research.

Several subjects expressed a requirement to view data presented in different ways - for example, all values ('raw', scaled and weighted) for one alternative, or historical values, so that the effect of changes could be easily seen. If such facilities were to be incorporated, then possible approaches to the management of information include the use of hypertext, and the use of dialogue to control the display of information. The former approach would leave the display of information directly in the hands of the user; the latter would permit (yet another level of) negotiation about which participant was in charge of the display and how information was to be presented.

**Improving the problem solving representation**

Even though everyone is involved in making decisions of this type (of varying degrees of importance) routinely, most people are unable to articulate their decision processes. The assumption which was tacitly made in the design of WOMBAT (that people were in some sense 'expert decision makers' and could discuss decision making tactics in the terms encoded in WOMBAT) was incorrect. As well as the ability to justify propositions, WOMBAT needs to be able to expand on, or explain, propositions (to answer
questions such as "What does that involve?" or "What does that achieve?"). This involves extending the dialogue mechanism, beliefs structure and reasoning mechanism to deal with different types of propositions.

In the problem solving domain, it requires developments to the problem solving knowledge representation to accommodate additional relationships between tactics. Such relationships include relevance indicators to define what conditions have to hold for a tactic to be applicable, the notion that certain combinations of tactics achieve intermediate goals, and a mechanism for noting the strengths and weaknesses of different approaches. Much of this can be done by extending the framework used for conducting dialogue and applying it in the collaborative problem solving domain. In this case, the strengths and weaknesses of different approaches would be expressed through the values system and means-ends beliefs (as this is the mechanism by which the system selects between alternative actions). The values system would therefore have to be accessible to the agent itself so that values could be presented in discussion with the user. This is a topic for further research.

A point which came out in two of the experimental sessions is that the user needs to be given access to a meta-level view of their problem solving processes. In the first instance, this might simply involve presenting a trace of the decision tactics performed; with a deeper representation of problem solving, such a view could include more information such as sub-goals achieved.

As an aside regarding the system's assessment of when it is appropriate to finish, P proposed that rather than the system's rather simplistic end criterion (of there being at least 8 objectives) it should be possible to build in more intelligence, for example grouping parameters into different categories, and requiring criteria from different groups before it is acceptable to end. G also expressed disquiet at the end criterion, but did not make any suggestion regarding a better one. As noted in §8.3.1, the decision about ending should also take into account the current state of the problem solving; for instance, it should be at a point where both participants are satisfied with the decision outcome.
Chapter 8: Empirical test of WOMBAT

Developments in the user model

The system at present has a user model which only contains the system's beliefs about the user's beliefs about the problem. Several subjects expressed the view that the dialogue would become tedious for experts. There is a clear requirement for WOMBAT to be more directive and tutorial with novices, perhaps explaining the decision steps and their purpose while teaching about decision processes, and then diminish the intervention to a critiquing approach with experts (saying nothing unless it disagreed or was asked a question). This would involve the system being able to acquire many more beliefs about the state of the student than is currently possible, as well as the ability to use those beliefs in its decisions about what to say next.

8.6 Conclusions

The basic approach of having a learning environment and a dialogue component is perceived by most of the subjects as being a sensible approach, and most subjects believe the system to have a (far from being realised) potential for use in Design education. To the extent to which the implementation matches the theoretical design outlined in §4.3, it is adequate; however, that design was sketchy in places, and not broad enough to cover all uses, so this formative evaluation has basically unearthed a new layer of issues and problems.

The implementation of the dialogue mechanism is surprisingly effective (considering that it is very much an experimental first attempt, with many \textit{ad hoc} features); it produces coherent (if idiosyncratic) output in most situations. However, the problem solving knowledge representation (currently encoded as independent justified beliefs) is inadequate for the purpose, and several other aspects of the dialogue (in terms of the range of issues it is able to cover, rather than in terms of its basic structure) would benefit from being improved. This must be viewed very much as a first step in the right direction.
Chapter 9: Conclusions and further work

In this final chapter, the threads from previous chapters are drawn together. The ideas which have been developed within this thesis are discussed in relation to relevant work in the contributing domains, and promising lines for further research are discussed.

One of the points which has come through very clearly in this thesis is the number and strengths of the connections between the contributing domains. So, for example, the design of an agent to support learning about multi-attribute utility theory exploits that same theory in its own decision making about what to do next. The rules employed in many existing IESs to define the teaching strategy are very similar to the condition-action rules discussed by agent theorists. Reichman's 'issue' and 'non-issue' context spaces reappear as abstract and concrete justifications for beliefs held and, just as collaborative problem solving is a form of joint action, so Suchman (1987) takes a view of conversation as joint action.

One of the great barriers to such interdisciplinary work is the different terminology which has developed in the different domains. For example, engineers apply the Weighted Objectives Method which decision analysts refer to as MAUT. Agents 'believe' while engineers 'think', and agents become committed to action while engineers simply do things. The different terminology used reflects the different contexts within which the same basic ideas have been developed. Recognising that such different perspectives exist, the critique which follows the summary of research contributions is divided into three sections (§9.2 - 9.4), each of which presents a different view of the thesis. This is followed by an outline of some of the interesting directions for further work and final concluding remarks.

9.1 Summary of research contributions

The main contributions of this thesis may be summarised as follows.

Firstly, in the context of design education, it extends the use of computers to the teaching of a topic within the curriculum which has hitherto been taught by traditional means (or not explicitly at all). The domain knowledge needed by the system (i.e. knowledge about
how a selection between pre-defined alternatives may be made) is based on information taken from the decision making literature.

As an Intelligent Educational System, WOMBAT incorporates ideas which have been circulating in the literature for some time, but have not been realised within an implemented system. One of these is the notion of making an explicit separation between the task-level and the meta-level. As discussed in following sections, the current representation of the levels is inadequate for teaching but, nevertheless, a second research contribution of this thesis is that such levels are implemented. Another idea within the literature is that of having the system collaborate with the user. Much of the discussion in the literature centres around the idea of the system and user learning together; in the case of WOMBAT, the system does not learn with the user, but is able to engage in collaborative problem solving. This is a third research contribution.

As discussed in §3.5, most tutoring systems which include an explicit tutoring strategy do so by encoding rules of the general type "IF <situation> THEN <action>". A fourth contribution of this thesis is that the design of the system accommodates an explicit representation of the teaching aims, rather than leaving these implicit in the design. The teaching aims encoded in the current system (as pedagogical values) are rather ad hoc, though adequate for demonstration purposes. The important point is that the structure can accommodate such aims, which could include motivational ones (§3.3) and aims such as "cover the specified syllabus" as well as those currently incorporated.

In the context of agent theory, a fifth contribution of this thesis is the definition of the action cycle and the dialogue goal-action tree at an appropriate level of detail for implementation, and the extension of the single-agent action cycle to accommodate collaborative activity.

WOMBAT is an implemented, if limited, agent. The limitations have been discussed at length in previous chapters, and are presented under the heading 'Further work' in this one (§9.5). Whatever the limitations, the implementation exists and works and demonstrates the potential gains to be had from such an approach. There are very few implementations in
existence incorporating agent theoretic ideas. A sixth contribution is that WOMBAT is the first such implementation which has expertise in engaging in extended mixed-initiative dialogue. It is also the first IES incorporating agent theoretic ideas.

The final point to be made is that the utterances generated by WOMBAT are derived from something nearer first principles than other existing approaches. The dialogue produced has been demonstrated to be largely coherent and purposeful, so a seventh contribution is that the WOMBAT prototype provides a basis for further investigations on computer participation in dialogue.

9.2 WOMBAT in the context of engineering design education

*Design education*

One of the original motivations for this research was a desire to address perceived shortcomings in current engineering education; in particular, to encourage the development of reasoning skills such as judgement and critical appraisal, to deal with problems which do not have a unique correct answer, and to encourage students to consider a design as a whole (in addition to, not instead of, considering different aspects of the design such as the design of components or designing for manufacture). The topic selected for teaching in this thesis was that of design evaluation. This is a topic which has not featured prominently in engineering education in the past, but the value of including explicit teaching about design evaluation is being increasingly recognised. Evaluation is viewed as being a step in an iterative design process, in which possible design concepts are compared against the problem definition and against each other; on the basis of the results of this activity, both problem definition and design concepts are modified and weaker concepts are rejected. For the purposes of this thesis, the topic has been simplified as far as possible, and has basically been reduced to the problem of selecting between defined alternatives. There do not appear to be many tools developed yet which are designed specifically to support this activity (selecting between alternatives), let alone systems to teach as well - either within engineering or in more general situations in which decisions of this type have to be made. As outlined in §2.1.3, while extensive use is made of computer technologies to support the teaching of
many other stages of the design process (principally for addressing issues where there is a unique correct answer) little use has as yet been made in teaching aspects of conceptual design. A notable exception to this is the CRACK system described in §2.1.3, which adopts a critiquing approach to kitchen designs. Thus one of the research contributions of this thesis is that it extends the use of computers in education into a domain where they have not been applied before.

As discussed in §8.5, design educators on the whole could see positive advantages of the approach taken in WOMBAT, but the current state of the implementation is a barrier to effective learning, and there is clearly a very long way to go before it could usefully be made available to students. (In particular, it covers an aspect of designing which, while essential, is generally covered in a very short time, so students would not be expected to invest much time in learning how to use a system that teaches this topic.) It has not been possible to evaluate WOMBAT in terms of the motivating issues - i.e. whether or not exposure to the system has any measurable effect on students’ reasoning skills, on their approach to open ended problems or on their ability to view a design as a whole. As discussed in §8.5.2, design educators who were asked what skills they saw WOMBAT as addressing highlighted the identification of important objectives and improving awareness of the decision making process. In this respect, the assessment of design educators neither confirms nor denies the potential of WOMBAT to address the motivating issues. Further work, as discussed in §9.5, is required before WOMBAT will be useful to students and before it will be possible to conduct a realistic evaluation of educational effectiveness.

Decision making

The topic of design evaluation has been simplified as far as possible, and the implementation deals only with the issue of selecting between alternatives. This is the core of the evaluation activity, so it is necessary to address this aspect of evaluation first. The WOMBAT prototype, once usable by students, provides a base for further work in extending it to cover other aspects of design evaluation (such as accommodating iteration and probability) as outlined in §9.5.
WOMBAT is able to make a decision unaided. With the current reasoning implementation, it has no notion of strategy, or real understanding of why it selects given decision tactics (its reasoning in this area is very shallow), but it will select all the steps which make up the WOM selection strategy outlined in the literature. No other possible decision strategies (such as those described by Pugh (§2.2), Tversky or Montgomery (§2.3)) have been fully implemented. In principle, with the exception of Pugh, it could readily be extended to accommodate these other strategies. The evaluation procedure described by Pugh is much more dependent on the iterative nature of the activity, so further work would be required to integrate it with other strategies. In the prototype implementation, the format of the matrix is tailored to the WOM. If other decision strategies were accommodated it might be appropriate to allow for different matrix displays according to the current state of problem solving.

Different design educators have different views on the appropriateness of applying numerically based selection methodologies (such as the WOM). Many favour such methods, arguing that they help the decision maker to be explicit about the factors being taken into account in decision making, and rendering the decision outcome open to inspection and criticism by others. Other design educators regard such methods as 'chalk and cheese arithmetic', and criticise them as giving the decision maker an unwarranted degree of confidence in the outcome. Despite the name of the system and the narrowness of the current implementation, the concern with the development of WOMBAT is to encourage students to think about how they are making the decision and to increase their awareness of the decision making process, and not in fact to impose one specific decision strategy as being better than another. My own view is that students should be familiar with different decision strategies, and be free to apply the one they feel most comfortable with.

**WOMBAT in use**

As discussed in §4.3, the decision was taken to implement an interface based on a formalised English-like language. At the time that this decision was taken, the alternative approaches were considered to be menus or a graphical interface of some sort. In the
evaluation (§8) it was found that, even when they were interacting with the system through me, users had difficulty expressing their thoughts in a way which the system would understand. Few subjects were able to exploit the system’s strengths and to probe it, and some viewed the dialogue as a barrier rather than a resource. Difficulties included establishing a model of what the system knows and how it operates. The latter problem is considered to be largely one of familiarity, and would have been improved if experimental subjects had had more opportunity to familiarise themselves with the style of interaction. In particular, the system’s collaborative approach is novel (c.f. (Self 1990), §3.4) and therefore not a style of interaction to which subjects had had previous exposure. An additional problem is that people are unable to articulate their decision strategies in terms which it can understand. At present, meta information about what the system knows about and what terminology it recognises is available in pull-down menus. More promising approaches to addressing these interface problems are discussed in §9.5.

In terms of the model of the acquisition of expertise put forward by Dreyfus and Dreyfus (§2.3), the user of WOMBAT is assumed to be a novice, needing to think analytically in order to be able to act. The user model incorporated in the prototype system only notes the system’s beliefs about the user’s beliefs, and does not contain any representation of the system’s assessment of the user’s level of expertise. The system varies its level of guidance in that it answers all the user’s questions, so that a user who asks many questions or frequently says that she has no ideas is given more help than one who makes a more active contribution to the problem solving. As emerged clearly in the evaluation (§8.5), WOMBAT is not in fact sufficiently directive for a real novice; this can be addressed partly by adding more discursive or directive actions at appropriate places in the dialogue goal-action tree, but more fundamentally it requires a fuller student model to be able to form reasonable assessments of when the user would benefit from a more directive approach.

As Tversky and Montgomery assert (§2.3), many people are reluctant to apply compensatory thinking in their decision making. This is borne out by the comments of some of the experimental subjects in §8. The difficulty is seen to stem from both the difficulty of organising large quantities of data and the reluctance to accept that trade-offs have to be
made. The provision of a suitable environment can ease the cognitive load of applying compensatory strategies by providing a framework for organising data, but can do nothing for the affective problem of helping people accept that decision making involves trade-offs. Whether or not the approach of increasing users' awareness of decision making processes can lead to an increased acceptance of compensatory decision strategies is an open research question.

Summary

In this section, the work on WOMBAT has been discussed in the context of the issues discussed in §2 relating to the educational context of this work. The approach taken in WOMBAT cannot be readily compared with established approaches because none are documented in such a way as to make comparison possible. Consequently, the WOMBAT implementation has been compared with a notional ideal solution to the question of how design evaluation should be taught. Many areas for further work have been identified, and are discussed in §9.5.

9.3 WOMBAT as an Intelligent Educational System

System design

WOMBAT supports students learning about a topic in which there are no right or wrong answers. For the purposes of this research, the domain has been characterised as justified beliefs. Therefore, the view of education being the "communication of knowledge" (Wenger 1987) is seen as being inappropriate. It might be argued that in the current implementation, WOMBAT errs too far in the opposite direction and is not sufficiently didactic, relying too heavily on negotiation and making the assumption that the user has sufficient knowledge to be able to form an opinion as a base for discussion. This point is relevant both in the particular domain being used for this work and more generally for the design of Intelligent Educational Systems.

WOMBAT does not fit into the 'traditional trinity' model of IESs as consisting of domain knowledge, student model and pedagogical knowledge (§3.1). In focusing attention
on dialogue, the work on WOMBAT centres around the interaction. The 'traditional trinity' components can all be identified within the implementation, but are not perceived as being central to the design. Domain knowledge is located in the file of possible beliefs which the system might hold and in the reasoning mechanism which accesses that file (§7.3.4). The student model currently exists within the agent state as the system's beliefs about the user's beliefs. Pedagogical knowledge is derived from the agent's values and means-ends beliefs as discussed below. The design of WOMBAT consists of a learning environment and a separate dialogue component. The dialogue component is capable of discussing both task-level and meta-level issues with the user, as discussed further below. Thus this thesis follows the trends of integrating learning environments with tutoring systems, of an increasing emphasis on metacognitive skills, and of developments towards more open dialogue as identified by Self (1988).

Although the design of WOMBAT was influenced by Elsom-Cook's work on Guided Discovery Tutoring (Elsom-Cook 1990a), as it is currently implemented WOMBAT is not a GDT system. As outlined by Elsom-Cook (see §3.4) one of the qualities of GDT is that system and user should have symmetric access to the environment. This is not the case in the current implementation. Although they negotiate over what happens in the environment, only the system has direct access to change things. Conversely, the user can look at information (as in the pull-down menus), but the dialogue agent cannot observe this activity, and therefore does not have an adequate picture of the user's activity or line of thought. As is argued by O'Malley (1990), learning is facilitated by the user being engaged in the domain, and able to manipulate it directly. Most work to date on direct manipulation has been based around microworlds in which the objects being manipulated are represented graphically on the screen but, as discussed in §4.3, it is believed that students would benefit from being able to manipulate data on the screen (e.g. in prioritising objectives) in much the same way. In terms of the design of WOMBAT, to make access to the learning environment symmetric it is necessary that the system and user be able to negotiate over control of the environment.
**Teaching strategy**

Partly because the current implementation has limited ability to respond to the user's level of expertise (§9.2) and partly because it has a limited range of possible utterances, it covers only a very small range on the continuum between guidance and discovery (Elsom-Cook 1988). The system does respond to the perceived state of the user as measured by the instances when the user asks a question or has no ideas, but there is a clear requirement to increase the range of possible utterances to include more imparting of information (e.g. explanations of what a given decision tactic involves). The teaching style is best described as collaborative problem solving. This style is derived not from a set of rules which define it (such as those encoded in WHY (§3.3) to define the socratic tutoring style) but from the possible basic actions which the dialogue agent can perform, and the corresponding values and means-ends beliefs.

The values encoded in the preference mechanism which define the system's teaching style are those which are relevant when the system is deciding on its response to the user's utterance. The use of values, together with means-end beliefs about how an action or utterance satisfies them, removes the need for ad hoc rules as employed in most extant IESs which govern how the system responds (see for example GUIDON's d-rules and t-rules (Clancey 1987), §3.5). A sketch of how utterances can satisfy values is presented in §7.8.4. Although little work has been done on refining the values system beyond the minimum necessary for demonstration purposes, the prototype provides a flavour of what should be possible within the theoretical framework of the model. In particular, the use of such a values system accommodates motivational considerations as advocated by Lepper and Chabay (§3.3). In the prototype, the main motivational values relate to not letting the user get stuck and keeping the interaction varied and interesting. The system does not at present incorporate actions such as praising or encouraging, which might be viewed as more directly motivational, but in principle the system could readily be extended to accommodate such actions and the corresponding values (though further work is required to extend the user
model to exploit them appropriately). The WOMBAT prototype also provides a basis for further investigations on the relationship between teaching aims and teaching techniques.

Comparing the teaching style of WOMBAT with the critiquing approach (see §3.4; also CRACK in §2.1.3, DecisionLab in §2.3.3), it may be noted that both allow for the possibility of there being multiple solutions. However, critiquing does not allow the user to justify her position, ask for help or engage in extended discussion over issues. Critiquing assumes that the user knows enough to get going, and is therefore not appropriate for use by complete novices. This last criticism could be levelled against WOMBAT as it is currently implemented, but is not a necessary feature of the design. Similarly, comparing the teaching style of WOMBAT with the coaching approach (exemplified by WEST and SMITHTOWN, both described in §3.2), it may be noted that the coaching approach assumes that the system knows how to solve the problem better that the user, and again does not allow the user to argue her case. All three approaches can involve the integration of a learning environment with a tutoring component; in the case of critiquing and coaching, the user interacts with the environment until the system makes an utterance, and the user cannot query or contest the system's view. The teaching style of WOMBAT involves much more extended dialogue and negotiation.

As outlined in §3.3, Self (1988) proposes a decoupling of the task-level and the meta-level. This decoupling is incorporated in the design of WOMBAT, in the system's ability to discuss objectives (task-level) and problem solving tactics (meta-level). However, the prototype is not sufficiently well developed to test for the advantages which Self suggests as accruing from such a decoupling. Collins and Brown (§3.3) discuss advantages of making the problem solving strategy explicit - both for retracing steps and for encouraging reflection on the problem solving process. As noted in §8, just articulating tactics is a step in the right direction, but has not been enough to achieve this. It has become clear that there are many levels at which the dialogue operates, such as how the learning environment might be organised, what the system knows about and can or cannot do, and how strategies are planned. As discussed in §8.5, the current meta-level knowledge representation is inadequate, and this is a topic for further research (§9.5).
Considering the roles of the participants in the interaction, in this case the system takes the role of collaborative problem solver, so that there is a symmetry at this level. This is in contrast to most extant tutoring systems (§3.4), in which either teacher or student is in control. It is also in contrast to emerging dialogue systems such as the Adviser system (Frohlich and Luff 1990), where the user is in control, and EDGE (Cawsey 1990) where the tutor takes the lead. Considering the collaborative approach, an observation made by Self (1990) is that it is not clear that students are amenable to a collaborative approach from their tutoring systems. Although the empirical investigation (§8) was too small to draw any strong conclusions, the results suggest that most users accept the collaborative approach. However, most of the subjects in the study expected to be in control and did not probe the system very deeply, so that the user's view was the dominant one.

**Summary**

In this section, the work on WOMBAT has been related to research on Intelligent Educational Systems, as discussed in §3. From an IES perspective, the most important aspects of this thesis are seen to be the application of AI techniques in a domain where there are no right or wrong answers, the focus on dialogue and negotiation in the context of collaborative problem solving, the separation which has been made between the task-level and the meta-level, and the integration of a learning environment with a dialogue component. Some areas for further work have been identified, and are discussed in §9.5.

**9.4 The dialogue component as an agent**

*Agent design*

The main theoretical contribution of this thesis is in the definition of the action cycle and the dialogue goal-action tree at a level of detail which make implementation straightforward, and in the extension of the single-agent action cycle to accommodate collaborative action. The definition of the action cycle is consistent with, but presented in more technical detail than, the approach outlined by Doyle, and also that presented by Kiss. As discussed in §6 and §B.6.2, the agent definition, and in particular the action cycle, builds
on some of the work of Kiss. The action cycle does not define a full agent architecture, which would include planning, learning (as in the ability to build up schemata), belief revision etc. The approach developed in WOMBAT is complementary to existing work in agent theory and planning - e.g. that of Downs and Reichgelt (1991), who discuss how the expertise might be acquired and constructed into plans, and that of Galliers (in press) on belief revision.

WOMBAT may be viewed as an opportunistic interaction expert (with the limited definition of expertise as in the term 'expert systems'), in the sense that it does not engage in advance planning about what to say or do next. It can participate in dialogue, accommodating autonomy of both parties. That dialogue is largely coherent and relevant; as discussed in §8, the shortcomings are due to the impoverished state of the system's reasoning and knowledge structures.

The hierarchical structure accommodates partial planning and decision making; it allows the agent to make decisions whenever there might be alternative candidate actions, while cutting down the search space (i.e. it does not have to consider a large number of options which are irrelevant in the current context). The approach of devising a hierarchical structure is similar to that described by Georgeff and Ingrand (1989), who describe the use of what they term 'plans' with different levels of detail in their procedural reasoning system (§5.5).

Kiss (§5.6.1), in his discussion about classes of attitudes, defines affective attitudes as being necessary for autonomy. In the case of WOMBAT, the only affective attitude included in the model is values; it is on the basis of its values that the dialogue agent makes decisions about what to do. The agent needs to be able to make decisions in order to be able to take the initiative, or address its own goals rather than simply reacting to the user. To engage in dialogue which takes both participants into account, it is necessary for the agent both to react to the user and to take the initiative. As currently defined, the WOMBAT dialogue agent reacts to the user through the use of transient goals; for each proposition stated or question asked by the user, the system adopts a transient goal to respond
appropriately. The system takes the initiative occasionally in proposing a change of tactic, but more often through adding to the response. The decision about what, if anything, to add to the response is influenced in a very indirect way by the dialogue history, but is not a direct reaction to any particular utterance which has gone before.

**Dialogue generation**

The dialogue component is capable of generating utterances from something nearer first principles than other existing approaches. It is my view that features such as adjacency pairs, openings and closings, dialogue games and speech acts (see §5.2) should be emergent properties of the dialogue rather than being hard-wired in to the design. In interactions with the prototype implementation, adjacency pairs (such as question-response) can be identified. Two classes of speech act (namely assertives and directives (§5.2.1)) feature in the interactions. Turn-taking is controlled by the current speaker indicating that they have finished; interruptions are not catered for. Openings and closings are not well developed in the current implementation; the agent's style could be viewed as rather abrupt, as it does not have beliefs about social conventions to do with initiating and terminating an interaction. A comparison of WOMBAT with the work of Baker, who has based his dialogue generation on the structure of dialogue games, is included in §B.6.1. Some features which have been identified in human-human conversations can be discerned in interactions with WOMBAT; others require further work (§9.5).

Following the work of Grosz and Sidner (§5.3.1), topic and intention are given separate structures. Their use of the term 'intention', which they also describe as the 'discourse segment purpose', corresponds most closely to WOMBAT's goals (and particularly the transient goals). The topic structure in WOMBAT is extremely simple because of the characterisation of the domain as independent justified beliefs, but has a similar stack structure to that of Grosz and Sidner. Both of these aspects of the design (the use of separate structures for topic and intention, and the use of a stack structure for topic) have been found to work satisfactorily. However, further work will be needed on the topic
structure if the domain is given a richer representation (removing the independence assumption).

As discussed in §5.3.1, a distinction can be drawn between abstract and concrete justifications for propositions. Abstract justifications are general ones, and are the type covered by WOMBAT ("I believe reliability is important because it is so inconvenient to have an unreliable car"). Concrete ones relate to specific examples ("I believe reliability is important because my friend’s car broke down on the motorway last week and she missed her flight to Paris"). These are not accommodated in WOMBAT. The possible extension of WOMBAT to cover concrete justifications of tactics (in which fictitious alternatives are created to illustrate features of particular tactics, such as cases in which an otherwise strong candidate solution would be eliminated by the application of a poor tactic) is discussed in §4.3.7. Although no research has been done as part of this thesis to test this hypothesis, there is evidence to suggest (see Epilogue) that people frequently use concrete justifications to support propositions, and so the possibility of extending WOMBAT to allow subjects to describe their own experiences should be investigated.

**Beliefs and commitments**

WOMBAT gradually acquires domain-specific beliefs in the course of an interaction. It only acquires beliefs when required to take a stand on the issue in question. The mechanism for maintaining beliefs is loosely based on a justification based truth maintenance system (such as that of Doyle (1979)), and is described in §7.3.2. This has been adequate for the purpose of demonstrating that the implementation of the theoretical dialogue agent is realisable, but has not been a focus for theoretical development as part of this thesis. As discussed in §7.3.4, possible beliefs which the agent might choose to adopt are held in a data structure which has the status of an authoritative text book. While a belief exists only in that data structure, the agent does not hold the belief, and the belief does not influence the agent's actions (or goals). Once the agent has adopted a belief, that belief can influence actions, so the agent may be regarded as having a commitment to the belief. Such
commitments are acquired through the dialogue. This is consistent with the 'speech acts' view of language as 'making things happen'.

As described in §7.6, the agent has mutual attitudes, namely mutual commitments and mutual working beliefs. It should be noted that these attitudes constitute the agent's view of what action the participants are jointly engaged in or what is mutually believed. In principle, it should be possible for the user's view to be different, although in the current implementation this principle is not realised (in that the user cannot say "Hey, you seem to think we've agreed X. Well we haven't, because I don't accept that"). A mutual commitment defines the joint problem solving action in which the participants are engaged. Holding a mutual working belief entails commitment to that belief for the purpose of problem solving, and not in any other sense. It has been necessary to make them 'working' because the system cannot revise its beliefs in a principled way without acquiring new beliefs from a source other than the pre-defined data structure; this mechanism allows the user to always have the final say in any negotiation. The pedagogical principle that the user must be allowed to have the last say is amply borne out in the results of the experimental work (§8).

The model of collaborative problem solving developed in this thesis depends on the participants negotiating and agreeing on commitments. This may be compared to the discussion by Power (1984), in which he defines mutual intention as depending on both parties believing that the other intends to achieve the same result. (He does not discuss whether they have to be doing the same thing, but it seems intuitively obvious that this is not necessary). Searle (1990) also discusses this matter, observing that the whole is different from the sum of the parts, in the sense that collaborative activity cannot be broken down into the separate activities of the participants. This is all consistent with the proposal in this thesis that collaborative activity necessitates the participants being at the same place in the collaborative goal-action tree, though they may be individually engaged in different activities (e.g. one listening, the other speaking).

As Shadbolt (1991) notes, much recent work on dialogue generation is based on the assumption that the discourse is a product of the need of the agents to construct, coordinate
and communicate plans. This approach is found in the work of Power (1979), Draper and Button (1990) (both discussed in §5.2.3) and Shadbolt (1989), as well as the more recent work of Grosz et al (§5.4). In this thesis, problem solving (the activity in which the dialogue participants are engaged) is approached opportunistically; there is no prior planning, so the dialogue generation is based on local decision making rather than advance planning. I view the two approaches as being complementary. One aspect of the work of Draper and Button merits further comment; in their model they proposed four levels of goals which the dialogue participants may have. The first, the external goal, corresponds to the mutual (problem solving) goal in this thesis. The second corresponds to transient goals and the third to (individual agent) dialogue goals. The fourth level (on which Draper and Button have not yet worked) relates to aspects of conversation such as turn taking and coordination; this aspect has not been covered in this thesis, but is recognised as an important area for further work.

As noted in §6.7, simultaneous actions (such as, in the case discussed there, marching and chanting) are not catered for in the current model. Possible approaches to this issue include defining basic actions which involve parallel activity (e.g. a basic action 'march and chant'), or re-defining the decision mechanism to permit the agent to choose to do more than one action at a time. The latter option would necessitate the agent having parallel processing capabilities. For the purposes of this discussion, these issues are peripheral. Parallel activity in the form of simultaneous coordinated action by many agents is accommodated. At present, an individual agent can only be committed to activities which are hierarchically related (i.e. one is a sub-part of the other). Paradoxically, in collaborative activity the dialogue participants may be committed to more than one - e.g. setting up a matrix while agreeing objectives or agreeing parameters and getting parameter values while agreeing objectives. In this case, execution of one action (agreeing objectives is the only possible action to which this applies at present) is suspended while another activity is completed. In this case, although the agents are committed to more than one action, only one is performed at a time. There is no simultaneous action.
Summary

In this section, the work on WOMBAT has been related to relevant research on dialogue generation and rational agency, as outlined in §5. In this context, the most important aspects of this thesis are seen to be the definition of an action cycle which defines opportunistic activity in a domain in which the agent has expertise. This action cycle integrates decision points with schemata (predetermined sequences of high-level actions) to provide an efficient but flexible approach to activity. The model has been extended to define collaborative activity in which more than one agent participates to reach a mutually agreed goal. The theoretical model has been implemented in the context of an Intelligent Educational System which is able to engage a user in collaborative problem solving, and has been demonstrated to work. Some areas for further work have been identified, and are discussed below.

9.5 Further work

Evaluation of educational effectiveness

As outlined in §9.2, one of the areas for further work is an evaluation of the educational effectiveness of the system. This includes developing some measure of how well the system addresses the issues which originally motivated its design (encouraging the development of skills such as judgement and critical appraisal, and encouraging students to consider the total design) and also seeking some sort of answer to the question posed by Self: how acceptable is the collaborative approach to users? As is made clear in the evaluation of the prototype system, many aspects of the implementation need further work before a realistic evaluation of educational effectiveness could be attempted. Some of this work involves simple development, while other aspects require further fundamental research. The areas identified as needing further work are as follows.

1) Debug existing implementation (e.g. correcting features known to cause unreasonable responses as discussed in §8.4.2).
2) Improve the dialogue interface: make the system output more elegant and more natural and improve the input mechanism. Improving the system output could be done in a relatively straightforward way, using rather more sophisticated canned text phrases than those currently employed, and including a checking mechanism to cut down on the number of repetitions of phrases (e.g. "I believe A. I believe A because B." could be reduced to "I believe A, because B"). Possible improvements to the input mechanism are discussed at greater length below.

3) Work on belief revision (allowing the user to change her mind after a proposition has been mutually agreed). At the simplest level this is a development issue, but a more sophisticated approach to belief revision would require further fundamental research.

4) Work on allowing the user to enter new information on objectives and alternatives. Again, a fairly simple solution to this problem (similar to the change_weights mechanism already included in the implementation) could be developed quite quickly, but a more sophisticated approach (for example, in which the system asked the user for additional information such as justifications for propositions and retained this information for use in future interactions) would require further research.

5) Implement all the other steps of the decision process (those which are currently executed without consultation by the agent) and all the other possible decision strategies (such as elimination by aspects) as collaborative activities. While much of this is basically time-consuming development work, it cannot be done satisfactorily without addressing the research issues of negotiating control over the environment and developing a fuller representation of problem solving knowledge, as discussed below.

6) Implement something to focus the user's attention on strategies. This could be done in a simple way by providing a tracer window which presents a record of what has been done and in what order, and maybe what was considered but not done. Alternatively, this could involve the development of a fuller representation of problem solving knowledge together with an ability to engage in meta-level discussion about problem solving strategies, as discussed below.

7) Replace reasoning about cars with reasoning about the artifact on which students are working. Such information could simply be slotted into the existing reasoning
structure. A more rigorous approach would include the development of the notion that there is a correspondence between the model of user, the model of usage, and hence the identification of important objectives.

8) Finally, devise an appropriate set of experiments using real students, and based on the data which students provide about their own designs. As the experience of designing the experiment for the formative evaluation illustrates (§8), any such experiment would require extensive pilot work before a full scale study could be devised, and it is inappropriate to consider the detailed design of such an experiment at this stage, or to surmise on the likely outcome.

Following the discussion above on abstract and concrete justifications for propositions, two additional possible extensions to the system are to enable the system to illustrate its arguments about the appropriateness of tactics by presenting examples, and to allow users to use their own experiences as justifications for propositions.

Two other areas for further work are that of incorporating dealing with ambiguity in the user's utterance (e.g. the utterance just made conflicts with something the system believes that the user believes), and that of incorporating dealing with understanding the purpose of the user's utterance (this involves recognition of the user's plan in making the utterance).

**An IES to support learning about design evaluation**

Most of the discussion so far has focused on the work required to extend the system to address the issue of selecting between defined alternatives more effectively (prior to evaluating its educational effectiveness). As a tutoring system to support students learning about design evaluation, there are several other aspects of the topic which the system would ultimately have to be extended to cover:

1) other aspects of selecting between alternatives, such as probability and the use of objectives trees (as discussed in (Pahl and Beitz 1984));

2) allowing for modifications to the problem specification and to the proposed alternatives;
3) relaxing the assumption that all data is independent, and dealing appropriately with interdependencies in the data. This is closely related to the notion of the form being a carrier for the functions. For example, doors (form) serve the functions of allowing passengers to get in and out, and protecting them while travelling. Other aspects of a vehicle's form, such as the rest of the body shell and, in particular, features such as crumple zones, also contribute to protecting passengers.

**A more competent dialogue agent**

The agent would be better able to adapt to new situations if it could adopt new lines of argument for subsequent use. This would involve asking the user for information it did not already have (which in turn would necessitate the agent being able to perform the appropriate basic actions). An appropriate way to deal with such information in a real teaching situation might be to include a 'grounds for belief' tag, such that as more people told it the same thing it's confidence in that belief would grow. In this case, the pre-defined lines of reasoning would have the same status, but initially a greater confidence rating.

As discussed above, the interaction is not symmetric, in that the dialogue agent has total control over the matrix, while the user has total control over the pull-down menus. Further work is needed on making access to the learning environment symmetrical, so that the user and system can manipulate it in the same way, and can observe each other's activity. This involves negotiation of control over the environment. This would take much the same form as the negotiation which has already been implemented. It could involve the same formalised indication of end of turn as has already been implemented (so that turn-taking is unchanged from that which already exists). Another possible approach, which would allow for interruptions, is to develop ideas discussed by Frohlich and Luff (1990) on identifying transition relevance places at which it is acceptable for interruptions to occur, and for another participant to take control. In the case of WOMBAT, the participant whose turn it was would both be speaker and have control of the learning environment.
**Interface issues**

A vital aspect of making WOMBAT really usable by people other than its designer is that of implementing a more transparent interface such that the user can both find out more easily what the system has knowledge about and easily establish how to express their thoughts in terms which it will understand. This must be done without unduly constraining the user's ability to express her own view of the problem. A possible approach is to exploit the system's expectation of what input is

a) expected (e.g. if it has asked a question, it expects an answer),

b) reasonable (e.g. the user might be contributing a new suggestion),

c) possible (e.g. the user changes the topic),

and display all possibilities - still with a free-form entry slot to allow the user to express whatever she wants. Another possibility (which I find less attractive) is to include context-sensitive help which would guide the user as to what were reasonable utterances.

**Values and means-ends beliefs**

Another area for further research is that of developing the values and means-ends beliefs to give a much more effective pedagogical interaction. This might also help to improve understanding of how human teachers operate. (But note that, as stated in §6.1, no claim of psychological plausibility is being made for the model.) It would also be interesting to try to adjust the values and means-ends beliefs to reproduce established teaching strategies such as socratic tutoring, coaching or critiquing. In adjusting values, it might also be possible to apply machine learning techniques, as described by Dillenbourg (1989), though this is seen as a less promising avenue for investigation, at least in the short term.

Developing the idea of learning values further, it might be possible for the system to learn the values for higher up the tree based on the anticipated [simulated] utility of the optimal possible action sequence way down the tree. (At present, the values relevant at each decision point are pre-defined, and are set independently of the values lower down the tree, regardless of the fact that the utility of a high-level action is dependent on that of the actions...
Chapter 9: Conclusions and further work

Design, Decisions and Dialogue

which are sub-parts of it.) Employing simulation in this way would be much slower than the current mechanism and would remove the advantage of having the deep structure unless learnt values were retained.

Problem solving representation

The current implementation makes explicit the split between task-level and meta-level, in that the system can discuss what to do next as well as aspects of the specific problem under consideration. It became clear in the evaluation that this was an inadequate representation of problem solving, and an interesting area for further research is the development of a fuller representation of problem solving knowledge, probably in a framework with the same basic structure as that which governs the agent's individual (dialogue) activity. In this case, values and means-ends beliefs would encode information about the relative merits and disadvantages of alternative strategies or tactics, and strategies would be expressed as combinations of tactics. At a dialogue level, values can be hidden, and are not discussed (the system cannot engage in a meta-level discussion on 'why did you ask me that' or 'why did you say that?' for which it might need to account for its values). In a full problem solving goal-action tree with values, the values correspond to the entities currently implemented as justifications, and the system would have to have access to its values when discussing tactics / strategies. In addition to a fuller representation of problem solving knowledge, the system and user also need to be able to discuss strategies, and to construct plans of how to tackle the problem (as well as being able to react opportunistically). One approach might be to have a learning environment window in which the dialogue participants are able to select and order tactics (i.e. engage in planning); this would ideally involve a richer negotiation mechanism, such as that described by Sycara (1989). There should also be a facility to engage in retrospective analysis of tactics done (whether opportunistically or planned).

In addition to negotiating over the control of the display, it is worth investigating the possibility of negotiating over the format of the display. This is firstly to make the display more appropriate for different decision making strategies, and secondly to accommodate all
the different display formats which subjects expressed a desire to have access to in the
course of the evaluation (§8). This involves developing a more flexible (intelligent?) design
for the matrix. The system should be able to reason about what are reasonable displays
given the current state of problem solving.

Etcetera

All of the topics for further research discussed so far are based around extensions to
the WOMBAT prototype, although many of them involve the development of theory which
would transfer readily to other domains. In the final few paragraphs in this section, topics
for further work which arise out of this thesis but do not involve extensions to WOMBAT
are discussed.

Considering WOMBAT as an IES in an engineering design context, it may be noted
that there are a large number of tools in existence to support the design activity. A possibility
which could profitably be investigated is that of integrating a dialogue component with an
existing system (e.g. a commercial CAD system or an analysis package) to extend the use of
such systems more effectively into the teaching situation. This approach would have some
features in common with the CRACK system discussed earlier. Another possibility is that of
testing the model in a different domain. If a domain for which a tutoring system has already
been developed (e.g. SMITHTOWN (§3.2)) were used, then comparisons could be made
between the different approaches.

It has not been found necessary to include intentions in the model to define the
dialogue agent (though they might be needed in situations such as performing a play, in
which the content of an utterance is planned in advance and delivery is coordinated with the
utterances and actions of other performers). However, considering applying the agent model
in other domains, an interesting issue to explore is that of including intentions in the
theoretical model. Taking the view expressed by Bratman (1990) that the most important
function of intentions is to coordinate activity, this would require some form of temporal
representation and also some planning capability. It would also necessitate the agent having
a representation of others' intentions.
9.6 Conclusion

In this thesis, emergent ideas from agent theory have been taken, together with ideas from the domain of Intelligent Educational Systems, and applied to the problem of supporting students' learning of a topic whose importance is growing. To summarise, the main features of this thesis are that:

• it introduces the use of computers in engineering design education to cover a topic which has been addressed using only traditional teaching techniques before,
• it applies AI techniques in a domain where there are no right or wrong answers,
• it focuses on dialogue and negotiation in the context of collaborative problem solving,
• it makes an explicit separation between the task-level and the meta-level,
• it integrates a learning environment with a dialogue component,
• it includes an explicit representation of teaching aims,
• it presents a definition of an action cycle which defines opportunistic activity in a domain in which the agent has expertise; this action cycle integrates decision points with schemata to provide an efficient but flexible approach to activity.
• the agent model has been extended to define collaborative activity in which more than one agent participates to reach a mutually agreed goal, and has been applied to the problem of conducting dialogue.

The prototype system which has been implemented in the course of this research is not going to have a place in the classroom in the foreseeable future, but provides a proof of concept of both the basic architecture of the system (as outlined in §4.3) and the agent architecture as defined in §6. It also provides a base for some exciting lines of further investigation.
In March 1991 (i.e. shortly after the research reported here had been completed) the researcher and spouse went through their own car selection exercise. It is perhaps salutary to report how that process went.

The previous summer, we had hired a Volkswagen Golf for two weeks, and both agreed that we liked driving it. It was easy to drive, it had adequate luggage capacity, it had sufficient leg-room for Chris (who is very tall), it was adequately comfortable, there was enough room for the children in the back, and it ran on lead-free petrol. Also, it fitted our self-image. So we knew that there was a car on the market on which we could agree.

In discussing more recently what cars we would consider the following points arose:

• I could name several European cars which I would consider seriously. But Chris does not like anything French or Italian. He considers them to be less reliable than German ones.

• Chris would be happy to consider Fords or Vauxhalls. I consider Fords to be too mass-produced; a colleague once worked with Ford designers, and did not rate their work highly. I have a similar (though less strong) prejudice against Vauxhalls.

• Neither of us is at all keen on Eastern European cars, thinking of them as old technology.

• The possibility of getting a Japanese car never really arose. We only have one friend with a Japanese car, and that had just spent three months in a garage waiting for spare parts. We have a stereotypical view of Japanese cars as being expensive to repair.

• Similarly, SAABs are expensive to repair (that is the complaint of our next door neighbour, who owns one). Volvos are too big or too heavy or too expensive (depending on which model is considered). A friend's fairly new one broke down on the motorway late one evening, causing major disruption to her family.

• Audis and BMWs have too much of an 'executive' image, which does not suit us.

Clearly, this rapid rejection exercise, which was based largely on prejudices and stereotypical views of cars, and the anecdotal experiences of friends, left us with two main
manufacturers: Austin-Rover and Volkswagen. We already had an Austin Maestro, which we were very happy with, but did not want another the same; my parents and brother both run Metros, and we wanted to be a bit different; Rovers are too 'executive'. So it was down to Volkswagen before we had even started to look seriously at car data or go for any test-drives.

In the first car showroom, it immediately became clear that the VW Polo had too little headroom for Chris. The salesman tried to persuade us that we should seriously consider a second-hand Passat in preference to a new Golf. At this stage, we had not finally decided whether to buy new or second-hand, whether to get a 1.3 or 1.6 litre Golf, or indeed whether to get a Golf or a second-hand Passat (we could not afford a new one). Given that all these possibilities satisfied our basic criteria, the factors we took into account in our decision making were:

- Given that we tend to keep cars for a long time, the prospect of (for once) having a new one was rather exciting. Also, there is a sense in which you 'know what you're getting' a bit better with a new car.
- Any car had to have 5 doors.
- We had a strong preference for a car with a catalytic converter.
- There was a consensus view (from 'Which' report as well as several car salespeople) that the Golf 1.6 performed better than the 1.3.
- We considered fuel economy to be important.
- The Golf fitted our self-image (of not being at all 'executive') much better than the Passat.
- The Golf’s luggage space was smaller but more flexible than the Passat’s.
- I could not see the boot of the Passat when sitting in the driving seat (important when reversing).
- The Passat was more comfortable, and would allow the children much more space in the back as they grew.
- The Passat had significantly worse fuel economy than the Golf 1.6 (which was worse than the 1.3 - more because the 1.6 has a catalytic converter, which worsens fuel economy, than because of the different engine size).
• We could not afford a Passat new enough to have a catalytic converter.
• As long as we could afford it, we were not overly concerned with price.
• Low depreciation and popularity of the Golf 1.6 meant that the saving from buying second hand was less significant than for some other makes.
• The best prices we were offered on a new 1.3 and a new 1.6 were only about £1000 different. The 1.6 has a catalytic converter as standard. The 1.3 does not.

By juggling around these facts and preferences in a qualitative way, we eventually decided to get a new Golf 1.6. This decision was based ultimately on what we judged we could afford, on an almost irrationally strong preference for a car with a catalytic converter, on our negative feelings about an 'executive' image and on a curious delight at the thought of buying our first-ever new car.

With two of us making the decision, it is not clear whether we would have been more analytic about the process if there had been a wider range of possibilities which neither of us was prejudiced against. Also, had there not been one solution on which we could readily agree, it is possible that we would have had to be more analytical to resolve a conflict. As it was, there was no real reason for either of us to try to overcome the prejudices of the other - or for us to expend additional resources on being more analytical about our decision making (as it was more likely to create conflict than to resolve it).

This experience supports the view (§8) that people choosing cars are more influenced by image and anecdotal evidence of the qualities of the alternatives than they are by direct measures of technical quality.
References

Papers marked with an asterisk (*) have been produced in the course of this research. All others are referenced in the text of this thesis.


Anderson, J. R. and Reiser, B. J. (1985) 'The LISP tutor' Byte Vol. 10 No. 4 pp 159-175.


Blandford, A. E. and Smith, P. R. (1986a) 'The evolution of computer based learning software design: Computer Assisted Teaching Unit Experience' Computers and Education Vol. 10 No. 2, pp 251-258

Blandford, A. E. and Smith, P. R. (1986b) 'The CADEd Project' University Computing Vol. 8, pp141-145
References


Cross, N. (1989) *Engineering Design Methods*, John Wiley and Sons Ltd., Chichester


Grant, D. P. (1976a) How to weight objectives using the Churchman-Ackoff Method Modular Section 76.02.19 (Grant, 1976F)

Grant, D. P. (1976b) How to use an alpha-beta model for decision-making with multiple objectives Modular Section 76.05.10 (Grant, 1976J)


Harris, H. G. (1983) 'The place of design in the training of professional engineers and managers' Design Studies Vol. 4 No. 3, pp 155-157


References


McCall, R. J. (1989) 'MIKROPLIS: a hypertext system for design' *Design Studies* Vol. 10 No. 4, pp 228-238


Self, J. A. (1989) 'The case for formalising student models (and intelligent tutoring systems generally)', invited paper presented at the 4th International Conference on AI and Education, 24-26 May 1989, Amsterdam, Netherlands


Sims, M. (1987) 'Educating designers for status... or for the status quo' Design No. 466 (October 1987) pp.18-20


References


Wormleton, P. R. (1986) 'Using CAL to broaden design experience of undergraduate engineers' Computers and Education Vol. 10 No. 1, pp 123-129

Zlotkin, G. and Rosenschein, J. S. (1990a) 'Negotiation and conflict resolution in non-cooperative domains' in Proceedings of The National Conference on Artificial Intelligence, Boston. AAAI.

Acronyms used in this thesis

AI - Artificial Intelligence

CAD - Computer Aided Design (see §2.1)

EBA - Elimination by Aspects (see §2.3.2)

GDT - Guided Discovery Tutoring (see §3.4)

HLD - High Level Dialogue (see §5.1)

ICAI - Intelligent Computer Aided Instruction (see §2.1.3)

IES - Intelligent Educational System (see §3.1)

ITS - Intelligent Tutoring System (see §3.1)

NL - Natural Language (see §5.1)

MAUT - Multi Attribute Utility Theory (see §2.2.5)

WOM - Weighted Objectives Method (see §2.2.3)
Appendix A: Protocol study (§4)

A.1 Subject profiles

The experiment (§4.2) was carried out using pairs of subjects, who were able to see each other and had a shared view of the problem specification and of any notes taken. The experimenter was also present, but made the minimum possible intervention.

All the subjects were chosen to have either a reasonably high level of technical/scientific education, or to know a reasonable amount about cars (however biased their views!). 5 groups were studied; two male, two female, one mixed; two groups were from the OU (all without families) and three from my circle of friends (all with families; none of the partners knew each other).

Profiles:

<table>
<thead>
<tr>
<th>Group 1</th>
<th>gender</th>
<th>family</th>
<th>profile of individuals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>f</td>
<td>n</td>
<td>A: BSc, knows little about cars</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B: 'O' level, professionally involved in</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>technical education, tinkers with cars</td>
</tr>
<tr>
<td>Group 2</td>
<td>m</td>
<td>n</td>
<td>A: BSc, has owned several cars</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B: MSc, does all servicing and most repairs</td>
</tr>
<tr>
<td>Group 3</td>
<td>m</td>
<td>y</td>
<td>A: 'O' level, does some repairs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B: PhD. Owns a car</td>
</tr>
<tr>
<td>Group 4</td>
<td>f</td>
<td>y</td>
<td>A: BSc. Owns a car</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B: BSc. Tinkers occasionally</td>
</tr>
<tr>
<td>Group 5</td>
<td>m/f</td>
<td>y</td>
<td>A: 'O' level, owns one</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B: BSc. Knows quite a lot</td>
</tr>
</tbody>
</table>

There was no fixed time limit on this exercise; all groups in fact took 40-60 minutes to complete the exercise.

A.2 The problem and alternative solutions

The problem as presented to the subjects was as follows; the alternative solutions are on the next page.

Select the most appropriate new car for a family of four. They can spend no more than £8000 (on the road price). They do not have an old car to trade in, and are not interested in any finance schemes. As this is their only car, they require the largest possible luggage capacity; the minimum acceptable is 25 cu. ft. The family is concerned for the environment, and therefore intends to use lead-free petrol, and they consider economy of greater importance than performance (within reason!). You may consider any additional criteria if you wish, and exercise your own judgement in making a recommendation. Please select one and justify your selection. Sorry - no test drives!!

The suggested procedure (which you don't have to follow if you don't want!) is:
1) Make an 'instant' decision
2) Make a better-informed decision when you have obtained whatever additional information you think you need.
3) Consult me for the next stage!
Brochures on the proposed alternatives give the following information. You may ask for any additional information that you require.
Appendix A

A: sporty hatchback, £7895 on the road

B: "boxy", large boot, 1100cc, runs on lead-free petrol, £6500 on the road

C: compact, medium boot, good acceleration, £7500 on the road

D: £7950 list price, very safe, takes either leaded or unleaded petrol (delivery charge £195)

E: £5980 list price, stereo and sunroof, takes lead-free petrol, 'flash' and tinny (£225 extra charges)

F: £7095 on the road, large family hatchback, seats 5
A.3 Car data

The car data which was given on request was as follows. Each numbered item is equivalent to one of the data strips which were available. This data does not refer to any existing car, and a few data items are totally unrealistic; this is unfortunate, but has not had any bearing on the results of the experiment.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. List price (£)</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>7595</td>
<td>6280</td>
<td>7305</td>
<td>7950</td>
<td>5980</td>
</tr>
<tr>
<td>2. Delivery etc. (£)</td>
<td>300</td>
<td>220</td>
<td>195</td>
<td>195</td>
<td>225</td>
</tr>
<tr>
<td>3. Total price (£)</td>
<td>7895</td>
<td>6500</td>
<td>7500</td>
<td>8145</td>
<td>6205</td>
</tr>
<tr>
<td>4. Servicing: freq.</td>
<td>6m</td>
<td>1y</td>
<td>6m</td>
<td>6m</td>
<td>1y</td>
</tr>
<tr>
<td>5. Servicing: cost (£)</td>
<td>85</td>
<td>65</td>
<td>78</td>
<td>83</td>
<td>110</td>
</tr>
<tr>
<td>6. Servicing: ease</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(0=impossible, 10=very easy)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. typical parts cost</td>
<td>125</td>
<td>95</td>
<td>85</td>
<td>120</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>(2 tyres + 1/3 exhaust) £ p.a.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. maintenance cost</td>
<td>295</td>
<td>160</td>
<td>240</td>
<td>286</td>
<td>205</td>
</tr>
<tr>
<td></td>
<td>(typical cost, £ p.a.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. insurance group</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10. running cost, £ pa</td>
<td>575</td>
<td>400</td>
<td>481</td>
<td>526</td>
<td>445</td>
</tr>
<tr>
<td></td>
<td>(0=depreciates fast, 10=holds price well)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. depreciation</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>12. fuel tank capacity</td>
<td>45</td>
<td>45</td>
<td>35</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>13. length</td>
<td>4.2</td>
<td>4.1</td>
<td>3.5</td>
<td>3.9</td>
<td>3.6</td>
</tr>
<tr>
<td>14. width</td>
<td>2</td>
<td>1.9</td>
<td>1.8</td>
<td>1.9</td>
<td>1.8</td>
</tr>
<tr>
<td>15. legroom</td>
<td>8</td>
<td>8</td>
<td>3</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>(0=poor, 10=ideal)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16. headroom</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>17. luggage capacity:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/5 occupants(cu.ft)</td>
<td>48</td>
<td>32</td>
<td>26</td>
<td>24</td>
<td>28</td>
</tr>
<tr>
<td>3 occupants(cu.ft)</td>
<td>66</td>
<td>32</td>
<td>26</td>
<td>24</td>
<td>28</td>
</tr>
<tr>
<td>2 occupants(cu.ft)</td>
<td>75</td>
<td>32</td>
<td>26</td>
<td>24</td>
<td>40</td>
</tr>
<tr>
<td>18. fuel economy:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>urban cycle (m.p.g)</td>
<td>38</td>
<td>44</td>
<td>36</td>
<td>48</td>
<td>42</td>
</tr>
<tr>
<td>constant 75 mph</td>
<td>41</td>
<td>43.6</td>
<td>38</td>
<td>42</td>
<td>41.6</td>
</tr>
<tr>
<td>constant 56 mph</td>
<td>56</td>
<td>54.8</td>
<td>45</td>
<td>61</td>
<td>56</td>
</tr>
<tr>
<td>19. fuel type (leaded/lead-free)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*costs £900 to convert</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20. catalytic converter</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>21. engine faults (ave no. p.a.)</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>22. breakdowns (ave no. p.a.)</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
</tr>
<tr>
<td>23. engine reliability</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24. bodywork</td>
<td>9</td>
<td>5</td>
<td>9</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25. engine capacity / cc</td>
<td>1600</td>
<td>1100</td>
<td>1300</td>
<td>1300</td>
<td>1400</td>
</tr>
<tr>
<td>26. engine smoothness</td>
<td>8</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27. gears: smoothness</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28. gears: number</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>29. number of seats</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>30. number of doors</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>31. manoeuvrability</td>
<td>5</td>
<td>2</td>
<td>9</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32. note</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Design, Decisions and Dialogue</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33. heating</td>
<td>8 5 5 8 2 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34. seat shape</td>
<td>8 5 6 8 5 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35. seats recline</td>
<td>8 5 5 y n y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36. ventilation</td>
<td>8 5 5 5 8 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37. 'ride'</td>
<td>8 2 8 8 5 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38. comfort rating</td>
<td>8 5 7 9 5 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39. interior finish</td>
<td>9 2 9 6 2 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40. sunroof</td>
<td>y n n n y n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41. stereo</td>
<td>y y y y y y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42. 'extras' rating</td>
<td>9 3 7 5 5 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43. controls: position</td>
<td>8 4 8 4 8 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44. controls: ease of use</td>
<td>8 8 4 4 4 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45. controls: rating</td>
<td>8 7 6 4 6 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46. instrument panel</td>
<td>5 5 5 5 5 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47. driving position</td>
<td>9 5 8 8 7 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48. acceleration:0-60/s</td>
<td>12.4 20.3 12.1 18.9 15.6 23.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49. top speed /mph</td>
<td>102 76 85 93 78 88</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50. performance rating</td>
<td>9 2 6 5 4 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51. aerodynamics</td>
<td>9 2 6 6 7 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52. style/aesthetics</td>
<td>9 2 5 9 4 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=ugly (subjective assessment!), 10=very stylish)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53. image</td>
<td>sporty boxy compact stylish flash family</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54. safety cage / crumple zones</td>
<td>8 5 3 10 3 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55. collapsible steering column</td>
<td>n n n n n n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56. brakes:anti-lock</td>
<td>n n y y n n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57. dual-circuit brakes</td>
<td>n n n n n n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58. brakes rating</td>
<td>9 6 8 9 5 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59. road holding</td>
<td>6 2 9 7 4 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60. handling</td>
<td>8 3 9 6 2 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61. driver visibility</td>
<td>5 7 4 8 6 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62. safety rating</td>
<td>7 5 6 8 4 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0=poor, 10=good)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63. type</td>
<td>hatch saloon saloon saloon hatch hatch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Utility values as entered in WOM spreadsheet program, based on the above data:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>price</td>
<td>1</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>servicing_ease</td>
<td>7</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>maintenance_cost</td>
<td>1</td>
<td>9</td>
<td>-</td>
<td>-</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>running_cost</td>
<td>2</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>size</td>
<td>8</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>luggage_space</td>
<td>6</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>cleanliness</td>
<td>5</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>reliability</td>
<td>7</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>economy</td>
<td>4</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>bodywork</td>
<td>9</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>seating</td>
<td>5</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>manoeuvrability</td>
<td>5</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>comfort</td>
<td>8</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>extras</td>
<td>9</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>controls</td>
<td>8</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>performance</td>
<td>9</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>safety</td>
<td>7</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>

A.4 Results data

Information given (summary)

<table>
<thead>
<tr>
<th>Group</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
</table>

Items mentioned in specification:

- Fuel type
- Total price
- Fuel economy
- Performance
- Luggage space
- No. of seats

Cues from alternative solutions information:

- Safety
- Extras
- Engine capacity
- Style/aesthetics
- Engine reliability
- Comfort
- No. of doors
- Bodywork
- Running cost
### Appendix A

**Weighted factors identified in stage 3 of experiment**

<table>
<thead>
<tr>
<th>Group</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Factors</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Items mentioned in specification:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>price</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>economy</td>
<td>6</td>
<td>-</td>
<td>5</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>performance</td>
<td>4</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>luggage space</td>
<td>-</td>
<td>2</td>
<td>10</td>
<td>7</td>
<td>-</td>
</tr>
<tr>
<td><strong>Cues from alternative solutions information:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>safety</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>extras</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Other information:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>comfort</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>bodywork</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>maintenance cost</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>running cost</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td>-</td>
</tr>
</tbody>
</table>

**Results of entering these values in WOM spreadsheet program**

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>1</th>
<th>8</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td>98</td>
<td>41</td>
<td>91</td>
<td>297</td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>152</td>
<td>31</td>
<td>71</td>
<td>295</td>
</tr>
<tr>
<td><strong>E</strong></td>
<td>142</td>
<td>27</td>
<td>62</td>
<td>223</td>
</tr>
<tr>
<td><strong>F</strong></td>
<td>104</td>
<td>49</td>
<td>128</td>
<td>315</td>
</tr>
</tbody>
</table>

N.B.: Group 4's result takes no account of their preference for 4 doors rather than 2, which would count against alternative A, but not affect the outcome of their decision making process.

**Number of instances within the five dialogues of...**

<table>
<thead>
<tr>
<th></th>
<th>3</th>
<th>2</th>
<th>0</th>
<th>0</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seek justification</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Disagreement</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Meta-level</td>
<td>23</td>
<td>10</td>
<td>9</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>Refer to &quot;too much data&quot;</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Misconceptions</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&quot;model&quot;</td>
<td>11</td>
<td>3</td>
<td>14</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>Problem-reference</td>
<td>12</td>
<td>9</td>
<td>17</td>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td>Notes taken</td>
<td>m</td>
<td>m</td>
<td>n</td>
<td>m</td>
<td>0</td>
</tr>
</tbody>
</table>

Note: these figures are somewhat subjective, in that they involve an assessment of what is and what is not an instance of each event. They do, however, serve to illustrate and support assertions made in §4.2.
Appendix B: WOMBAT implementation

B.1 A first attempt at formalising the action cycle

What follows is a first attempt at developing a formal definition of the action cycle (§6.3.1). This formalisation is incomplete, failing to express the temporal qualities of the cycle.

$$\text{goal}(y) \land \text{believes}(\text{achieves-progress-towards}(x,y)) \land \text{believes}(\text{relevant}(y)) \Rightarrow \text{wants}(x)$$

$$\text{wants}(x) \land \neg(\text{committed}(x)) \land \text{believes}(\text{relevant}(x)) \land (\forall z. (\text{wants}(z) \land z \neq x) \Rightarrow \text{prefer}(x,z))$$

$$\Rightarrow \text{committed}(x)$$

$$\text{committed}(x) \land \neg(\text{basic-act}(x)) \land \text{believes}(\text{is-a-subpart-of}(y,x)) \Rightarrow \text{goal}(y)$$

$$\text{committed}(x) \land \text{basic-act}(x) \Rightarrow \text{done}(x)$$

$$\text{believes}(\text{achieves-progress-towards}(x,y)) \land \text{done}(x) \land \text{goalreaches}(x,y) \Rightarrow \text{goalreached}(y)$$

$$(\forall z. \text{believes}(\text{is-a-subpart-of}(z,x)) \Rightarrow \text{goalreached}(z)) \Rightarrow \text{done}(x)$$

$$\forall x \text{ committed}(y) \Rightarrow \neg(\text{done}(x))$$

$$\text{done}(x) \Rightarrow \neg(\text{committed}(x))$$

$$\text{goalreached}(x) \Rightarrow \neg(\text{goal}(x))$$

$$\forall z \text{ committed}(x) \Rightarrow \neg(\text{wants}(z))$$

$$\text{t_goal}(y) \land \text{believes}(\text{relevant}(y)) \Rightarrow \text{review}(y)$$

B.2 Implementation of action cycle and dialogue goal-action tree

What follows is the Lisp implementation of the central controlling routines in WOMBAT.

```lisp
(defun agent (pd)
  (new-menubar pd)
  (agent_do (agent_begin pd)))

(defun agent_begin (pd)
  (inityd
   pd
   'believes
   (problem)
   (objectives)
   (working_objectives)
   (alternatives)
   (tactic)
   (user (believes))
   (dh)
   (sentences)
   (tsold)
   (topicstack))

  (wants)
  (goals survive)
  (committed)
  (t_goals (address_now))
  )
```

; at the moment, the agent only wakes up when given a problem definition, pd.
; it would probably be theoretically cleaner to have it sitting there waiting
; for a user and a pd... but that's post-doc.
; pd takes the form:
; '((kws kw1 kw2 ...) (need (req1) (req2) ...) (wants obj1 obj2 ...))

"
(d_goals)
(d_committed)
(worldstate (done) (goalreached) (active) (ddone) (recent)
 system_turn user_exists)))

;agent_commit sorts outstanding commitments, based on current
;preferences and beliefs.
;note_recent is a convenient thingy which records recent decisions for the
;purpose of very interaction.
;This just happens to be a convenient place to do it
(defun agent_commit (agent_state)
 (note_recent
  (list_clear 'wants)
  (list_m_add 'committed (prefer agent_state) agent_state)))

;agent_do deals with doing things!
(defun agent_do (agent_state)
 (cond ((goals_relevant agent_state)
    (agent_do (list_clear 'worldstate done)
      (agent_commit (agent_wants agent_state))))
    ((null (list_extract 'goals agent_state))
      (set-menubar *default-menubar*)
      agent_state)
    ((commitp (commit agent_state))
      (agent_do (commit subparts (commit do agent_state))))
    ((doneactp agent_state)
      (agent_do (tidy state agent_state)))
    (t (agent_do (agent_commit
      (agent_wants (agent_goals agent_state))))))

;agent_goals generates new goals from non-primitive acts
(defun agent_goals (agent_state)
 (list_add 'goals)
 (list_extract (list (commit agent_state))
   (dialogue_tree_get 'subparts)) agent_state))

;agent_wants sorts out a list of all actions reasonable in the current
;context/worldstate given the system's values and beliefs.
;if there are goals then it only tries to satisfy those for which
;the prerequisites are already satisfied. (Note: the way things are set up,
;there will only be one of these at a time.)
(defun agent_wants (agent_state)
 (agent_want_sub
  (goals_relevant agent_state) agent_state))

;agent_want_sub generates wants from the current relevant (uncommitted) goals
(defun agent_want_sub (goalist agent_state)
 (cond ((null goalist) agent_state)
    (t (list_add 'wants)
      (list_extract (list (car goalist)
        (dialogue_tree_get 'apt))
        (agent_want_sub (cdr goalist) agent_state)))))

;commitcheck returns a list of the goals in goalist for which no commitment
;has been made
(defun commitcheck (goalist commitlist belaptslist)
 (cond ((null goalist) nil)
   ((commitp (list_extract (list (car goalist)) belaptslist)
     commitlist)
     (commitcheck (cdr goalist) commitlist belaptslist)))
Design, Decisions and Dialogue
Appendix B

(defun commitp (agent-state)
  (cond ((null (list-extract '(committed) agent-state)) nil)
    (t (car (list-extract '(committed) agent-state))))
)

(defun doneactp (agent-state)
  (doneactsub (list-extract '(worldstate goalreached) agent-state)
               (dialogue-tree-get 'subparts))
)

(defun doneactsub (grlist splist)
  (cond ((null splist) nil)
        (t (doneactsub grlist (cdr splist))))
)

(defun goals_relevant (agent-state)
  (cond ((null (list-extract '(goals) agent-state)) nil)
        (t (commitcheck
            (goal-rel-check (list-extract '(goals) agent-state)
                            (list-extract '(worldstate) agent-state)
                            (dialogue-tree-get 'relevant))
            (list-extract '(committed) agent-state)
            (dialogue-tree-get 'apt))))
)

(defun goal-rel-check (goalist worldstate beirelist)
  (cond ((null goalist) nil)
        ((goal-rel-p (list-extract (list (car goalist)) beirelist)
                     worldstate)
         (cons (car goalist) (goal-rel-check (cdr goalist) worldstate beirelist)))
        (t (goal-rel-check (cdr goalist) worldstate beirelist))))

(defun goal-rel-p (listconds worldstate)
  (cond ((null listconds) t)
        (t (commonmember (cdar listconds)
                         (list-extract '(goalreached) worldstate))))
)

(defun note-goal_apt (act apts goalist)
  (cond ((null goalist) nil)
        ((goal-rel-p (list-extract (list (car goalist)) apts) worldstate)
         (cons (car goalist) (goal-rel-check (cdr goalist) worldstate beirelist)))
        (t (goal-rel-check (cdr goalist) worldstate beirelist))))

(defun init_pd (pd agent-state)
  (list-add' (believes problem) pd agent-state)
)

(defun note_goal_apt (act apts goalist)
  (cond ((null goalist) nil)
        ((nga-sub act apts (car goalist)) (car goalist)))
)

;commitp returns nil if no outstanding commitments,;
;and the first o/s commitment otherwise
(defun commitp (agent-state)
  (cond ((null (list-extract '(committed) agent-state)) nil)
    (t (car (list-extract '(committed) agent-state))))
)

;doneactp checks whether all the subgoals have been reached to mean that
;an act has been done
(defun doneactp (agent-state)
  (doneactsub (list-extract '(worldstate goalreached) agent-state)
               (dialogue-tree-get 'subparts))
)

;doneactsub checks whether all the subparts of an act (car splist) are
;in the goalreached list, and returns the list of act and subgoals if so
(defun doneactsub (grlist splist)
  (cond ((null splist) nil)
        (t (doneactsub grlist (cdr splist))))
)

;goals_relevant returns a list of goals which are relevant, towards the
;achievement of which no commitment has been made. Otherwise it returns nil
(defun goals_relevant (agent-state)
  (cond ((null (list-extract '(goals) agent-state)) nil)
        (t (commitcheck
            (goal-rel-check (list-extract '(goals) agent-state)
                            (list-extract '(worldstate) agent-state)
                            (dialogue-tree-get 'relevant))
            (list-extract '(committed) agent-state)
            (dialogue-tree-get 'apt))))
)

;goal-rel-check returns a list of relevant goals
(defun goal-rel-check (goalist worldstate beirelist)
  (cond ((null goalist) nil)
        ((goal-rel-p (list-extract (list (car goalist)) beirelist)
                     worldstate)
         (cons (car goalist) (goal-rel-check (cdr goalist) worldstate beirelist)))
        (t (goal-rel-check (cdr goalist) worldstate beirelist))))

;goal-rel-p returns true if relevance is indicated, nil otherwise
(defun goal-rel-p (listconds worldstate)
  (cond ((null listconds) t)
        (t (commonmember (cdar listconds)
                         (list-extract '(goalreached) worldstate))))
)
(t (note_goal_apt act apts (cdr goalist))))
(defun nga_sub (act apts agoal)
 (cond ((null apts) nil)
       ((and (member act (cdar apts)) (equal (caar apts) agoal))
        agoal)
       (t (nga_sub act (cdr apts) agoal)))))

; primactp tests to see whether act is a primitive action or not
(defun primactp (act)
 (primactsub act (dialogue_tree_get 'subparts)))

; primactsub returns true if act does not have sub-parts,
; false otherwise
(defun primactsub (act slist)
 (cond ((null slist) act)
       (equal act (caar slist) nil)
       (t (primactsub act (cdr slist)))))

; primact_do controls the doing of a primitive act
(defun primact_do (agent_state)
 (funcall (commitp agent_state) agent state))

; primact_tidy updates the state of the agent and the worldstate after a
; primitive act has been done
(defun primact_tidy (agent_state)
 (list_m_delete '(committed) (commitp agent_state)
 (tidy_goals
 (list_m_add '(worldstate goalreached)
 (note_goal_apt (commitp agent_state)
 (dialogue_tree_get 'apt)
 (list_extract '(goals) agent_state))
 (list_m_add '(worldstate done)
 (commitp agent_state)
 agent_state))))

; tidy_goals removes any goals which are in the goalreached list
(defun tidy_goals (agent_state)
 (tidy_goalsub
 (list_extract '(worldstate goalreached) agent_state)
 (agent_state)))

; tidy_goalsub iteratively tidies up one goal that has been reached
(defun tidy_goalsub (grlist agent_state)
 (cond ((null grlist) agent_state)
       (t (list_m_delete '(Goals) (car grlist)
            (tidy_goalsub (cdr grlist) agent_state))))

; tidy_state updates the done list and the goalreached list:
; if all the subgoals of an act have been achieved, then record that the act has
; been done, and forget that the subgoals have been reached
(defun tidy_state (agent_state)
 (tidy_t_goals
 (tidysub (doneactsub (list_extract '(worldstate goalreached) agent_state)
 (dialogue_tree_get 'subparts)
 agent_state))))

(defun tidy_t_goals (agent_state)
 (cond ((null actlist) agent_state)
       (t (list_m_delete '(committed) (car actlist)
           (list_m_add '(worldstate done) (car actlist)
           (list_delete '(worldstate goalreached)
           (cdr actlist)
           agent_state))))))

(defun tidy_t_goals (agent_state)
Design, Decisions and Dialogue
Appendix B

(list_delete '(worldstate goalreached) '(explicit_agree resolve_conflict explicit_not-known respond_to_expectation inform_incomprehensible resolve_ambiguity)

(agent_state))

;dialogue_tree_get returns the required information about the dialogue
;goal-action tree
defun dialogue_tree_get (label)
  (cond ((equal label 'apt)
    (process_pd (process_pd_p)
      (make_decision listen respond finish_p)
      (await_input await_input_p)
      (split_input split_input_p)
      (store_old_ts store_old_ts_p)
      (process_sentences process_comprehensible process_incomprehensible finish_p)
      (update_ts update_ts_p)
      (understand_loc_f understand_impart_p understand_elicit_p)
      (understand_purpose understand_purpose_p)
      (note_ambiguities note_ambiguities_p abstain_p)
      (assess_pc tidy_ts_p note_agree_p note_disagree_p
        note_not-known_p note_expectation_p)
      (forget_sentence forget_sentence_p)
      (make_inferences make_inferences_p)
      (update_state update_state_p)
      (sort_t_goals sort_t_goals_p)
      (select_t_goals select_t_goals_p)
      (address_t_goals address_a_t_goal finish_p)
      (active_t_goal active_t_goal_p)
      (t_goal_ts t_goal_ts_p)
      (t_goal_goalist t_goal_goalist_p)
      (review_tactic abstain_p propose_end_tactic_p)

    (do_response elicit_belief elicit_agree_impert_new_belief
      do_new_action elicit_action elicit_proposal
      impart_proposal finish_p)

    (select_topic_belief select_topic_belief_p)
    (select_topic_agree select_topic_agree_p)
    (do_elicit do_elicit_p)
    (tidy_ts tidy_ts_p)
    (do_adopt do_adopt_p)
    (do_impert do_impert_p)
    (do_action do_action_p)
    (select_topic_action select_topic_action_p)
    (select_topic_proposal select_topic_proposal_p)
    (select_proposal select_proposal_p)

    (explicit_agree action_p agree_and_p
      do
      impart_agree_p elict justn_p
      imp_ag_el justn_p imp ag_imp justn_p ignore nk_p)
    (resolve_conflict imp dis el_conf_p elicit justn_p imp dis el justn_p
      impart_disagree_p imp dis_imp elj_p
      imp dis impj elj_p)
    (explicit_not-known imp nk el_conf_p imp nk imp poss_p
      impart nk_p ignore nk_p)
    (respond_to_expectation respond_p resp and just p refuse ben_p)
    (inform_incomprehensible inform_incomprehensible_p)

    (construct_response construct_response_p)
    (make_response make_response_p)
    (expect_input expect_input_p))
  (equal label 'subparts)
  (listen await_input split_input store old_ts)

287
Appendix B  Design, Decisions and Dialogue

(process_sentences make_inferences update_state)
(process_comprehensible update_ts understand_loc_f understand_purpose
note_ambiguities assess_pc forget_sentence)
(process_incomprehensible note_incomprehensible forget_sentence)
/respond sort_t_goals select_t_goals address_t_goals
review_tactic add_to_response construct_response
make_response expect_input)
(address_a_t_goal active_t_goal t_goal_ts t_goal_goalist)

(elicit_belief select_topic_belief do_elicit)
(elicit_agree select_topic_agree do_elicit tidy_ts)
(impart_new_belief select_topic_belief do_adopt do_impact)
(do_new_action do_action)
(elicit_action select_topic_action do_elicit)
(elicit_proposal select_topic_proposal do_elicit)
(impart_proposal select_proposal do_impact))

((equal label 'relevant)
'((make_decision (goalreached process_pd))
(split_input (goalreached await_input))
(store_old_ts (goalreached split_input))
(process_sentences (goalreached store_old_ts))
(make_inferences (goalreached process_sentences))
(update_state (goalreached make_inferences))
(understand_loc_f (goalreached update_state))
(understand_purpose (goalreached understand_loc_f))
(note_ambiguities (goalreached understand_purpose))
(assess_pc (goalreached note_ambiguities))
(forget_sentence (goalreached assess_pc note_incomprehensible))
(select_t_goals (goalreached sort_t_goals))
(address_t_goals (goalreached select_t_goals))
(t_goal_goalist (goalreached active_t_goal))
(review_tactic (goalreached address_t_goals))
(add_to_response (goalreached review_tactic))
(construct_response (goalreached add_to_response))
(make_response (goalreached construct_response))
(expect_input (goalreached make_response))
;the following multiple goalreached lists are really nasty, and
;goalreached should be a relevance function sort of thing
(do_elicit (goalreached select_topic_belief select_topic_action
select_topic_proposal select_topic_agree))
(tidy_ts (goalreached do_elicit))
(do_impact (goalreached do_adopt select_proposal))
(do_adopt (goalreached select_topic_belief))))

288
B.3 The values included in the implementation

The values which the implemented agent has are as follows. The importance of each, as measured by the weights, has been set arbitrarily in the current implementation.

<table>
<thead>
<tr>
<th>values</th>
<th>weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>avoid all effort</td>
<td>30</td>
</tr>
<tr>
<td>be benevolent to the user's goal</td>
<td>30</td>
</tr>
<tr>
<td>user learns through cooperative decision making</td>
<td>40</td>
</tr>
<tr>
<td>decision is made</td>
<td>15</td>
</tr>
<tr>
<td>make agreement explicit</td>
<td>10</td>
</tr>
<tr>
<td>understand user</td>
<td>25</td>
</tr>
<tr>
<td>satisfy goals</td>
<td>25</td>
</tr>
<tr>
<td>don't ask too many questions</td>
<td>50</td>
</tr>
<tr>
<td>do minimum</td>
<td>12</td>
</tr>
<tr>
<td>encourage reflection</td>
<td>11</td>
</tr>
<tr>
<td>user develops an understanding of the problem</td>
<td>18</td>
</tr>
<tr>
<td>make progress</td>
<td>28</td>
</tr>
<tr>
<td>be benevolent to the user's belief</td>
<td>25</td>
</tr>
<tr>
<td>vary interaction</td>
<td>8</td>
</tr>
<tr>
<td>avoid repetition</td>
<td>10</td>
</tr>
<tr>
<td>make conflict explicit</td>
<td>23</td>
</tr>
<tr>
<td>make not-known explicit</td>
<td>45</td>
</tr>
<tr>
<td>don't state obvious</td>
<td>52</td>
</tr>
<tr>
<td>motivate user</td>
<td>25</td>
</tr>
<tr>
<td>force user to think</td>
<td>25</td>
</tr>
<tr>
<td>minimise effort</td>
<td>90</td>
</tr>
<tr>
<td>have integrity</td>
<td>45</td>
</tr>
<tr>
<td>be benevolent to the user's expectation</td>
<td>30</td>
</tr>
<tr>
<td>encourage externalisation</td>
<td>13</td>
</tr>
<tr>
<td>agree mutual goal</td>
<td>35</td>
</tr>
<tr>
<td>don't get in rut</td>
<td>25</td>
</tr>
<tr>
<td>demonstrate</td>
<td>7</td>
</tr>
</tbody>
</table>

B.4 A detailed tour of the dialogue goal-action tree.

The top level decision (how to survive) is discussed in §7.2.3. What follows here is a description of the remainder of the tree. See Figs 7.2-7.5 for the graphical presentation of this material. In this tour, all lists are described in flat form rather than nested. So for example, the list which is implemented as (believes (user (believes (objectives ...)))) appears in this discussion as (believes user believes objectives).

B.4.1 The action of teaching the user.

The action teach_user consists of reaching the goals process_pa (a goal which is mis-named, and requires that the system reach a state in which the mutual problem solving goal (d_goal) of make_decision has been noted in the agent state) and make_decision (reach a state in which a collaborative decision has been made), where reaching the first is a prerequisite for tackling the second.

The goal process_pa is reached simply by performing the basic action process_pd_p, which consist of the basic action of adding make_decision to the (d_goals) list.

B.4.2 The goal of making a collaborative decision

The system design is based on the premise that making a collaborative decision (or any other form of collaborative problem solving) involves the participants taking turns. As
Appendix B Design, Decisions and Dialogue

discussed in §4.3, in the current model interruptions are not catered for. The participant whose turn it is has control of the dialogue until they explicitly 'hand over'. (Notionally, control over the learning environment would also be passed from one participant to the other, but this has not been implemented; only the system can manipulate the matrix, and only the user can operate the pull-down menus). This turn-taking model entails an 'utterance cycle' (See Fig 6.8).

This decision point (make_decision) is one which the agent reaches many times. The values which are relevant to making the decision are:

• decision_is_made, which has a weighting of 27, and is satisfied by the action listen provided that the relevance functions user_turn is true, and the relevance function decision_made is false. This value is satisfied by the action respond if the relevance function system_turn is true and decision_made is false.

• be_benevolent_user_goal, which has a weighting of 35, and is satisfied by listen or respond (depending on whether user_turn or system_turn is true) provided that user_wants_to_finish is false, and by finish_p if user_wants_to_finish is true.

• make_agreement_explicit, which has a weighting of 5 and is satisfied by respond if system_turn is true and agreement_explicit is false.

This decision is clearly dominated by the system's value on being benevolent to the user's goal, so in the current implementation the decision is effectively made by the user, but in principle it would be possible for the system to suggest finishing (for example if it somehow assessed that the user had done all that was possible with it - though making such an assessment would be a non-trivial task), or for the system to try to persuade the user to continue (maybe because there is evidence that the user has not yet understood the subject matter) once they had stated their intention to finish. Whether it would in fact be desirable to create such a system is an open question; all I would note here is that the decision making structure would accommodate such developments. Due to the feature of the implementation that the doing of any basic action causes the system to consider the 'parent' goal to have been reached, the doing of the null action finish_p causes the system to reach the goal make_decision.

B.4.3 The action of listening and the goal of processing the input sentences

The action listen consists of reaching the following sub-goals (in the defined order):

1) await_input, the goal of reaching the state where the user's input has been received. This goal is reached by the performance of the basic action await_input_p. This action consists of displaying the user input window on the screen and storing the user's input as a new list in (believes dh).

2) split_input is the goal of reaching the state in which the latest user input has been split into sentences (each starting with either 'elicit' or 'impart' - the system cannot deal satisfactorily with ill-structured utterances), which are stored in (believes sentences). This goal is reached by the performance of the basic action split_input_p.

3) store_old_ts is the goal of reaching the state in which the previous topicstack has been stored in tsold. This goal is reached by the performance of the basic action store_old_ts_p

4) process_sentences is the goal of having processed all the sentences in the user's input. The possible actions which can contribute to the reaching of this goal are process_comprehensible (to process the first sentence in the (believes sentences) list, which the system believes to be comprehensible), process_incomprehensible (processing a sentence which does not start with elicit or impart - a state which the system can only detect if the user's utterance did not begin with either of these terms.), or finish_p. Due to the feature of the implementation that the doing of any basic action causes the system to consider the
The 'parent' goal to have been reached, the doing of the null action finish_p causes
the system to reach the goal process_sentences. The decision about which of these
three actions to perform is governed by just one value - that of understand_user
- and by the relevance functions sentence_exists (i.e. there is an as-yet
unprocessed sentence in (believes_sentences)) and sentence_comprehensible (that sentence starts with elicit or impart), so this
decision point is equivalent to a simple conditional structure

5) make_inferences, the goal of reaching the state in which all valid inferences have been
drawn from the user's utterance is achieved by the doing of basic action
make_inferences_p as discussed in §7.4.2.

6) update_state, the goal of reaching the state in which the system has noted the change in
speaker-turn from user to system, through the performance of action
update_state_p

B.4.4 The action of processing a comprehensible sentence

The action process_comprehensible consists of reaching the following sub-
goals (in the defined order):

1) update_ts, a goal reached by the performance of the basic action update_ts_p. This
action updates the topicstack, based on the new input sentence.

2) understand_loc_f, a goal reached by the performance of one of the two basic actions,
understand_elicit_p and understand_impart_p. The only value
relevant to this decision is understand_user, and the relevance condition is
simply whether the sentence is a statement or a question.

3) understand_purpose should be a goal to understand the user's purpose in uttering the
sentence. In the present implementation, the basic action understand_purpose_p
simply checks and deals with truth maintenance issues such as removing double
negatives.

4) note_ambiguities should note any ambiguities raised by the sentence - e.g. that it
expresses a belief which is incompatible with some previously stated belief. The goal
is reached by the performance of one of the actions note_ambiguities_p or
abstain_p. In practice the decision mechanism is rigged so that the action
abstain_p (which is a null action) is always relevant, and
note_ambiguities_p is never relevant.

5) assess_pc is the goal of reaching the state in which the system has formed some opinion
about the propositional content of the sentence. It does this by selecting one of the
basic actions abstain_p, note_agree_p, note_disagree_p, note_not-
known_p, or note_expectation_p to commit to. (Note: as soon as any one of
these actions has been done, the goal is considered reached.)

6) forget_sentence is the goal of having removed the sentence from (believes
sentences). This goal is achieved by the doing of basic action
forget_sentence_p.

B.4.5 The action of processing an incomprehensible sentence

The action of processing an incomprehensible sentence entails achieving the goals
note_incomprehensible and forget_sentence. Forget_sentence is described above (§B.4.4).
Note_incomprehensible is reached by the performance of basic action
note_incomprehensible_p, which sets up a transient goal (_goal) to inform the user
that the sentence is incomprehensible.

B.4.6 The action of responding

The action process_comprehensible consists of reaching the following sub-
goals (in the defined order):
1) **sort_t_goals** is the goal corresponding to the basic action **sort_t_goals_p**, which sifts through the outstanding t_goals and removes any which are no longer relevant. In practice, what this entails at the moment is deleting any **explicit_agree's** and any t_goals which involve justifications, for topics which are no longer in the active topicstack. This can only be described as an *ad hoc* sorting procedure, but it happens to work quite well!

2) **select_t_goals** is the goal corresponding to the basic action **select_t_goals_p**. This involves deciding which outstanding t_goals are to be addressed in the course of the current utterance. The selected t_goals are noted in (**worldstate address now**). The current (unprincipled) approach is to first select a t_goal which is likely to involve and elicit - preferably the one which is active, if such a t_goal exists, and then up to three further t_goals, which will not involve elicits. Any outstanding tactic-related t_goals should always be addressed. Obviously, this selection procedure, like the sorting procedure described above, is *ad hoc*. Reference to the example interactions in §C.3 show that the system's output tends towards verbosity, and that a more principled approach to t_goal selection would yield more palatable results.

3) **address_t_goals** is the goal of reaching the point where the selected t_goals have been addressed. See §B.4.7 below.

4) **review_tactic** is the goal of having decided whether or not the time has come to review the current decision making (joint problem solving) tactic, and if so to have addressed that goal. The basis on which this decision is made and the subsequent courses of action are described in §B.4.8 below.

5) **add_to_response** is the goal of having added any further sentences the system considers appropriate to (**believes sentences**) before output. The possible actions and the decisions involved are discussed in §B.4.9.

6) **construct_response** is the goal corresponding to the basic action **construct_response_p**, in which all the sentences stored in (**believes sentences**) are built into one list in (**believes dh**).

7) **make_response** is the goal corresponding to basic action **make_response_p**, in which an output window is displayed on the screen and the first list in (**believes dh**) - i.e. the current utterance - is output to this window.

8) **expect_input** is the goal corresponding to basic action **expect_input_p**, which simply notes the change of turn back from system to user.

### B.4.7 The goal of addressing transient goals and the action of addressing a transient goal

The possible actions which can contribute to the reaching of the goal **address_t_goals** are **address_a_t_goal** (to address the first t_goal in the (**worldstate address_now**) list, or **finish_p**. As described previously, the doing of the null action **finish_p** causes the system to reach the goal **address_t_goals**. The decision about which of these two actions to perform is governed by just one value - that of **satisfy_goals** - and by the relevance function **t_goal_exists** (i.e. there is a t_goal which has not yet been addressed in (**worldstate address_now**)), so this decision point is equivalent to a simple conditional structure.

If the decision taken is to **address_a_t_goal** this action consists of reaching the following sub-goals (in the defined order):

1) **active_t_goal** is the goal corresponding to basic action **active_t_goal_p**, which simply notes in (**worldstate active**) that the t_goal is active (in the sense of having been acted upon).

2) **t_goal_tsk** is the goal corresponding to the basic action **t_goal_tsk_p**, which updates the topicstack by putting the topicstack item corresponding to the current t_goal at the top of the topicstack list.

3) **t_goal_goalist** is the goal corresponding to the basic action **t_goal_goalist_p**, 292
which adds the t_goal to the (goals) list (to be addressed in the same way as any other individual goal), and deletes it from the (worldstate address_now) list.

B.4.8 The goal of reviewing the problem solving tactic and the action of initiating change

The purpose of review_tactic is basically to say 'Am I happy with what we're doing now?', and if not to initiate change. So the two possible actions available are abstain_p (don't make any changes) and propose_end_tactic_p. The only value on which the decision is based is make_progress, together with the relevance function time_to_change_tactic; this relevance function returns true if there is an active tactic (the only possible one in the prototype is agree_lifestyle_objectives) which has been completed. In the current implementation, this requires that enough (at least 7) factors have been agreed and there are no outstanding conflicts. Propose_end_tactic_p simply inserts the topic 'believes tactic finish_tactic' into the topicstack and puts the corresponding sentence 'impart believes tactic finish_tactic' (which is displayed as "I believe we should do finish_tactic") into (believes sentences). This effect should be achieved through the use of the t_goals structure, but is not in the current implementation.

B.4.9 The goal of adding to the response, and the various associated actions

There are many possible actions which can contribute to the reaching of the goal add_to_response, and a correspondingly large number of values and relevance functions. This is one of the decision points at which the design decisions about how the system should communicate, and what sort of teaching strategy should be adopted, become apparent. It is also one of the decision points at which it becomes clearer why it is appropriate to have a decision mechanism involving values and relevance functions, rather than the more simple 'if-then' conditional type of decision mechanism. The possible actions are:

1) elicit_belief. This involves reaching the goals select_topic_belief and do_elicit, with the corresponding actions select_topic_belief_p and do_elicit_p. These basic actions select the appropriate topic, based on the current activity ('believes tactic' if there is no active tactic, 'believes objectives' if the tactic is agree_lifestyle_objectives - other topics would be appropriate when other tactics have been fully implemented), and then construct the corresponding 'elicit' sentence (for example 'elicit believes objectives').

2) elicit_agree. This involves reaching the goals select_topic_agree and do_elicit, with the corresponding actions select_topic_agree_p and do_elicit_p. Do_elicit_p is described in the paragraph above. Select_topic_agree_p changes the topic at the top of the topicstack by removing any justifications then adding agree, so the system always elicits agreement with a basic proposition rather than with any justifications. For example, if the item at the top of the topicstack were (believes objectives convertible not justification easily_damaged) - the belief that a reason for not having a convertible car is that the roof is easily damaged - then select_topic_agree_p would change this to (believes objectives convertible not agree), prior to asking the user whether they agree that having a convertible is not an important objective.

3) impart_new_belief. This involves reaching the goals select_topic_belief, do_adopt and do_impart, with the corresponding actions select_topic_belief_p, do_adopt and do_impart_p. Select_topic_belief_p is described in paragraph (1) above. Do_impart_p is much like do_elicit_p, except that it imparts a belief rather than eliciting one. Do_adopt_p adopts a new belief (using the plausible reasoning mechanism) on the topic at the top of the topicstack. For example, if the item at the top of the topicstack were (believes objectives convertible not justification) - then do_adopt_p might adopt the belief (believes objectives
convertible not justification easily damaged). In the context of impart new belief, do adopt p will only adopt a basic belief - for example (believes tactic do wom) or (believes objectives luggage) - but in other contexts it can adopt different and more complex beliefs.

4) do new action. This involves simply doing the action at the top of the d_committed list. This never in fact gets done at the moment because there's never a mutual commitment hanging around as a candidate action, and it never just autonomously sets up a d_commitment without the say-so of the user.

5) elicit action, elicit proposal and impart proposal are all actions which it will never select in the current implementation. In principle, these involve (respectively) asking the user to do the action which is currently at the top of the d_committed list (which typically would involve the user directly manipulating the learning environment in some way), eliciting a proposal (i.e. a possible belief which the user doesn't necessarily hold - as in 'just say something - anything!', and imparting a proposal - as in 'what do you think about X?' or 'Do you think we should consider X?'.

6) finish p is the basic action of finishing adding to the response.

The decision between these alternative actions is based on several values and means-ends beliefs.

Following are the various t-goal goal-action trees.

B.4.10 The actions associated with the various possible t-goals

T-goals are addressed by composite basic actions, simply to exploit the design quirk that if the action is basic then the parent goal is reached as soon as the action has been done. This makes for some very complex basic actions which should be divided into smaller units. (A topic for further research is the development of a more principled notion of how these complex actions are built up - i.e. of what makes any given sequence of statements effective, appropriate, incoherent or whatever.)

B.4.10.1 explicit agree

The possible ways of explicitly agreeing with a proposition are to:

agree action p (obviously only relevant if the proposition relates to a tactic), which adopts the mutually agreed tactic as d_committed and says so.

agree and do p (again only relevant if the proposition relates to a tactic), which agrees the action and then autonomously executes it. The selection between agree action p and agree and do p is based on whether or not the action in question is one which is discussed or done autonomously by the system in the current implementation.

impart agree p, which simply causes the system to articulate its agreement with the proposition

elicit justn p, i.e. to elicit a justification for the proposition

imp_ag_el justn p, i.e. to impart agreement then elicit justification ("I agree with you, but why do you think that anyway?"

imp_ag_imp justn p, to impart agreement and impart a justification ("I agree with you because..."

ignore nk p, to forget the t_goal (used in the case where the user was agreeing with the system in the first place).

Again, the decision between these alternative actions is based on several values and means-ends beliefs.
B.4.10.2 resolve conflict

The known ways of trying to resolve a conflict are to
imp_dis_el_conf_p, to impart disagreement and elicit confirmation. ("I don't agree
with you. Are you really sure about that?")
imp_dis_el_justn_p, to impart disagreement and elicit a justification ("I disagree.
Why do you believe that?")
impart_disagree_p, to simply impart disagreement
imp_dis_imp_bel_p, to impart disagreement and impart an alternative belief (This
routine is called to argue that "I don't think we should do X next. I think we should
do Y", where X and Y are tactics - however dissimilar. In principle it might also be
invoked to argue that "I don't believe that X is important, but I do believe Y is",
where Y is somehow loosely related to X - for example, X might be top speed and Y
acceleration. This sort of argument is not possible while the domain is simply
represented as independent justified beliefs, as such cross-relationships are not
encoded.)
imp_dis_impj_elag_p, to impart disagreement, impart justification and elicit
agreement.
imp_dis_impj_elj_p, to impart disagreement, impart justification and elicit a
justification ("I disagree about X because Y. Why do you believe X?").

Again, the decision between these alternative actions is based on several values and
means-ends beliefs.

B.4.10.3 explicit not known

The two ways the system knows of to deal with propositions which are unknown to
it are:
imp_nk_el_conf_p, to impart the fact that the system does not know, and to ask the
user to confirm that they meant what they said, and
imp_nk_imp_imposs_p, to impart that the system does not know, and that therefore it is
not possible.

The choice is made between these alternatives simply on the basis of whether the
proposition in question relates to a tactic (in which case it is impossible) or not.

B.4.10.4 respond to expectation

The possible actions in this case are:
respond_beny_p, to respond benevolently (and answer the user's question directly)
refuse_beny_p, to refuse to answer the question

In principle, the system chooses whether to answer the user's question directly or
not. In practice, at the moment the system always chooses to respond benevolently.

B.4.10.5 inform incomprehensible

The goal of informing the user that her utterance was incomprehensible is achieved
by the basic action inform_incomprehensible_p.

B.5 Descriptions of problem solving tactics

In this section, all the possible problem solving tactics which WOMBAT has
information about are described in some technical detail.

There are three known ways to organise data; matrix opens a new window, and
sets up a matrix of objectives against alternatives. Ideally, the information in this window
would grow incrementally as the system and user reached agreement about what tactic to pursue next, but in fact it currently gets set up with all the possible headings in place, just with gaps to be filled in for data not yet obtained. More trivially, list_objectives and list_alternatives should set up windows to list agreed objectives and to list known data on alternatives, but these routines are currently null.

Agree_ps_objectives is the action of agreeing to include all objectives as stated in the problem specification. This entails copying all the objectives listed in (believes problem wants) to (believes objectives), (believes user believes objectives) and (believes working_objectives), in appropriate forms. It also articulates this action, saying which objectives have been agreed because they are in the problem specification.

Agree_lifestyle_objectives is the action of agreeing to include additional objectives based on the likely lifestyle of the user. This is achieved through the dialogue.

Finish_tactic involves removing the current tactic from the (d_committed) list. This only currently applies to agree_lifestyle_objectives. Similarly, continue_tactic is a null action, also only relevant while engaged in agreeing lifestyle objectives.

Identify_parameters is the action of assigning suitable parameters for measuring each objective. For example, a suitable parameter for measuring the objective 'minimise purchase price' (encoded simply as 'price') would be on-the-road price, measured in pounds sterling. A suitable measure for the objective 'maximise safety' is clearly much harder to agree on. Ideally, it would include information about a wide range of different safety features. This would involve the construction of an objectives tree as discussed by Pahl and Beltz. Even for simpler objectives such as 'maximise luggage capacity', the measure is not totally straightforward, if one takes into account the folding of seats in hatchbacks and estate cars. In the current implementation, parameters to measure all objectives known to the system are simply pre-defined in car_data; the user has no control over this aspect of decision making.

Eliminate_fail_ps is the action of agreeing to eliminate alternatives which fail to satisfy all aspects of the problem specification from further consideration. The decision is made by reference to (believes problem need). The values of all parameters specified there are compared with the values stored in car_data, and if any alternative fails on any score, then the fact is noted in the appropriate place in (believes alternatives).

Eliminate_worst_on_1_objective should involve agreeing an objective on which to eliminate the worst (e.g. the alternative with the worst safety rating, or the most expensive one), and then committing to eliminating it. This routine is currently null.

Similarly, select_best_on_1_objective should involve agreeing an objective, and committing to selecting the alternative which scores best on that objective. It is also currently null.

Eba is another action which has not been implemented. It should implement elimination by aspects.

Pairwise_comparison is a composite action, including the sub-goals of agreeing a pair of alternatives and agreeing a means of eliminating the worst alternative. Again, it has not been implemented.

Scale_parameters should consist of agreeing a total scale for all parameters (e.g. 1-6 or 0-10) then assigning scale points to agreed intervals of actual parameter values. (e.g. on a scale of 0 to 10, assign the scaled value 0 to any price over £10000, 1 to £9500-£10000, 2 to £9000-9500, 3 to £8600-£9000, 4 to £8200-£8600, 5 to £7900-8200, 6 to
£7600-7900, 7 to £7200-£7600, 8 to £6900-£7200, 9 to £6500-£6900, and 10 to any price under £6500; the use of variable intervals in this way allows for more sensitive scaling in areas of particular interest.) In a full implementation, this might well include a scaling device as sketched in §4.3 for the user to manipulate (by agreement), but for the purposes of this research the action of scaling parameters consists of the system deciding autonomously on worst and best values for each parameter (which correspond to values 0 and 10 respectively). (These values are stored in routine car_data.) When scaling is applied to the values of parameters for all alternatives, this interval is split up uniformly. The user has no input at all into this aspect of decision making.

**Rank objectives** orders the agreed objectives in order of importance. It is currently null.

**Assign weights** assigns weighting values to all agreed objectives. This should be done collaboratively; the user should even (with the agreement of the system) be able to move objectives around on the screen (as described in §4.3) before assigning numerical weights. At the moment, the system decides weights autonomously (they are pre-defined in car_data). A 'quick-fix' routine called change_weights has been implemented, so that if the user so desires, all weights can be re-defined (after being assigned by the system) simply by entering the desired values for all objectives as prompted. This is neither a principled nor a desirable solution to the problem, which should involve some negotiation over which party is to define or modify weightings, and is a further research project in itself.

**Get parameter values** does the extremely obvious action of getting the values of the parameters relating to all agreed working objectives for all alternatives. The values are retrieved from the routine car_data described below.

**Assign values** assigns scaled values to all parameters for all alternatives by calculating where in the interval best - worst the value of each parameter lies, and hence what the scaled value is. As discussed above, this does not allow for variable scaling, and is performed autonomously by the system.

**Do_wom** calculates the weighted values for all parameters for all objectives by multiplying together the scaled value and the weighting value. It then adds the weighted values for all parameters to derive an overall utility value for each alternative. The calculation is currently (and probably incorrectly) a black box operation as far as the user is concerned.

**End** terminates the decision process.

**Car data** stores data for all known objectives and all defined alternatives. Each list within car_data is as in this example:

```
(price sterling 10000 5000 7 (a 7895) (b 6500) (c 7500) (d 8145) (e 6205) (£ 7095))
```

This encodes the facts that price is measured in pounds sterling, that the worst value (for scaling purposes) is 10000 and the best 5000, that the default weight of this objective is 7 and that the values for alternatives a, b, c, d, e and f are 7895, 6500, 7500, 8145, 6205 and 7095 respectively.

**B.6 Comparison between WOMBAT and work of Baker and Kiss**

**B.6.1 Comparison between WOMBAT and the work of Baker**

Baker's (1989) doctoral research had three main strands. He proposed an architecture for an ITS to facilitate the learning of musical interpretation, developed a cognitive model for the perception of musical grouping structures and developed a model for tutorial interaction ('Negotiated Tutoring').
Baker sought to synthesise recent trends in ITS research into a coherent theoretical model, including an increased symmetry in the range of interactions available to the dialogue participants, a synthesis of multiple interaction styles in terms of lower level dialogue units, the use of explicit negotiation of cooperative interaction goals, and an emphasis on cognitive and metacognitive skill acquisition. KANT generates tutorial dialogues based on 'critical arguments with an educational purpose' though, as Baker observes, Negotiated Tutoring is a general approach which could be implemented with different dialogue moves to yield a tutoring style other than critical argument.

Baker states that Negotiated Tutoring incorporates the view that ITSs should be concerned with the cognitive skill of reasoning in dialogue (note: this should not be confused with the metacognitive skills of reasoning per se), which the student should be enabled to acquire simply through the process of engaging in dialogue. It also aims to facilitate the acquisition of the metacognitive skill of belief revision. However the extent to which such aims are achieved cannot be established since KANT has not been empirically tested.

In terms of identifying general objectives for tutorial interaction, particularly for domains which can be characterised as justified belief, and by focusing attention on the high level structure of dialogue (rather than on natural language implementation), Baker has laid the foundation for the research reported here. However, there are also significant differences between the two research projects:

- Although both can be characterised as justified belief, the target domains of the two projects are significantly different. In WOMBAT, all information is dealt with as if independent (of all other information), so much of Baker's work on concept activation (or domain traversal) is not relevant to this thesis; neither are his considerations of memory and 'knownness' (i.e. of what the student can be expected to know or remember).

- My interest in dialogue is motivated by an interest in pedagogical values, and in how utterances can be chosen to satisfy those values. The embedding of 'educational principles' in the mechanism for selecting between possible dialogue goals does not satisfy this interest. The approach being taken involves the explicit encoding of pedagogical values, together with means-ends beliefs about how particular utterance types are likely to satisfy those values. In practice, KANT and WOMBAT are based on significantly different approaches, KANT on dialogue games, WOMBAT on agent theory.

- The extended negotiation phase in KANT is acknowledged by Baker as being tedious. A reliance on implicit negotiation is preferred by the author, so while most of Baker's other design desiderata (such as interactional symmetry and the synthesis of multiple interaction styles in terms of lower level dialogue units) are being adopted, the focus on explicit negotiation (as incorporated in KANT) is not.

B.6.2 Comparison between WOMBAT and the work of Kiss

The theory of rational agents as proposed by Kiss has been outlined above (§5.5, §5.6). WOMBAT follows in the spirit of that research, but the details of the implementation differ in many respects from both the outline design (Kiss et al 1988) and the Demo 1 implementation (Kiss and Brayshaw 1989). There are two principal reasons for this difference. The first lies in the different motivations for the projects. Kiss et al are seeking to proceed slowly and carefully from outline design to implementation, focusing attention on the detailed design of each component as it is developed in order to learn about the details of the agent architecture as it is built up. In contrast, the purpose of WOMBAT is to demonstrate that the application of this theory has potential benefits in teaching, so a complete system has been developed and inevitably the design of some of the components has been somewhat ad hoc. In terms of the development of theory, work on WOMBAT may be viewed as a 'top down' approach, while Kiss's is 'bottom up'.

298
The second important difference is in domains; Kiss et al are developing their work in the context of file system management, and the agent serves a function similar to that of a lawyer acting on behalf of a client, seeking to serve that client's best interests. WOMBAT operates in the domain of education, and the agent serves the function of teacher cum collaborative problem solver. Thus the relationship between the agent and the user is significantly different, and (more importantly) the effect of actions is unpredictable for WOMBAT, whereas Kiss et al depend on being able to establish whether or not a value is satisfied, and assume that it is known whether or not a particular action will cause a given value to be satisfied.

The similarities between WOMBAT and the work of Kiss et al are much greater than the differences, even though the list of differences as enumerated below is much longer than the list of similarities, which are that:

- Both are based on the same fundamental ideas. Both incorporate an action cycle which governs the operation of the agent.
- Both decide between alternative possibilities by reference to the agent's values.
- Both use the same mechanism for making such choices (although in WOMBAT it is referred to as MAUT, while Kiss et al refer to it as decision analysis).
- Both incorporate the same basic set of attitudes - beliefs, values, goals, wants and commitments.

The differences between WOMBAT and the Demo 1 implementation of Kiss and Brayshaw (1989) and between WOMBAT and the outline theory of Kiss et al (1988) can be simply stated as follows:

- WOMBAT can sustain a dialogue; Demo 1 simply receives one user input appertaining to file deletion and makes a decision about whether or not it will delete the desired file, based on factors such as its importance to the system and the believed expertise of the user. It cannot participate in a sustained dialogue.
- Demo 1 is designed to illustrate the decision making process, so much of the research effort has been invested in that. WOMBAT's decision making process is hidden away.
- In Demo 1, the base-line weightings on values are adjusted to reflect the state of the world (in their case, the expertise of the user). The approach being taken in WOMBAT is to keep the weightings on values fixed, and to modify the behaviour of the system dependent on the state of the user through the use of relevance functions ("action X is likely to make progress towards satisfying value Y if and only if relevance condition Z holds true"). Demo 1 makes use of both relevance functions and importance (to alter weightings of values); WOMBAT only uses relevance functions.
- Because no changes are ever made to the weights of values, the practice of maintaining values in a tree structure was found to be an unnecessary overhead. A simple list of values together with their weights is maintained in WOMBAT.
- The distinctions between goals and values, and between goals and actions, are blurred in Demo 1; the leaf nodes of the values tree are all presented as goals, and no clear distinction is made between goals and actions. In WOMBAT, the distinctions between these three attitudes are essential to the operation of the system.
- In Demo 1 the values are accessible to the system (e.g. Kiss talks about being able to explain decisions taken in terms of the system's values). In WOMBAT they are not; they are held in the preference mechanism and not directly accessible to the agent. This is
Appendix B  Design, Decisions and Dialogue

considered preferable simply because numerical weights are attached to them; while no claims are being made about cognitive plausibility, it is clearly cognitively highly implausible to have numbers within the information which is available to the agent about itself.

- The set of values relevant in any given context is identified in different ways, reflecting the different domain characteristics.
- WOMBAT has much 'compiled knowledge' in order to give speedier operation. Very little is calculated from first principles during an interaction.
- Kiss's architecture is 'flat', in that all decisions are about selecting between basic actions, so he does not have the tree structure which has been incorporated in WOMBAT.
- The action cycle incorporated in WOMBAT is somewhat simpler than that outlined by Kiss and is tied in to the goal-action tree structure on which it operates, and is therefore much less general than Kiss's. On the other hand, it has been fully implemented, which Kiss's has not.

B.7 Trace of the agent state

This trace covers one utterance cycle (from the point where the system makes one utterance to the point where it makes the next) part way through an interaction, illustrating the changes which occur within the agent state. The full trace, which printed out the entire agent state every time, has been edited to highlight the changes as they occur. The notation '<> is used to denote aspects of the agent state which are unchanged and therefore omitted.

The agent was initialised with the following problem specification:

? (agent '«(kw family only_car)(need (price max 8000»(wants safety comfort»)

The early utterances in the interaction are:

user: (Elicit Believes Tactic)
user: (Impart Believes Tactic Eliminate_Fail_Ps)
user: (Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject Justification Price Impart Believes Tactic Agree_Ps_Objectives Impart Believes Tactic Agree_Ps_Objectives Agree)
user: (Impart Believes Justification)
user: (Impart Believes Tactic Agree_Ps_Objectives Justification Take_Problem_Spec_Into_Account Elicit Believes Tactic Agree_Ps_Objectives Agree)
user: (Impart Believes Agree)
user: (Impart D_Committed Agree_Ps_Objectives Impart Believes Objectives Safety Justification In_Ps Impart Believes Objectives Comfort Justification In_Ps Elicit Believes Tactic)
user: (Impart Believes Tactic Agree_Lifestyle_Objectives)
user: (Impart D_Committed Agree_Lifestyle_Objectives Impart Believes Objectives Running_Cost Elicit Believes Objectives)

This is the point at which tracing starts. At this stage the agent state is as follows. This starting state is annotated to make it easier to read.
Design, Decisions and Dialogue

Appendix B

The system's beliefs about the problem (as given in problem specification), objectives, working objectives (with outline lists available for filling in values later), alternatives, tactics, about the user's beliefs about tactics and objectives, a full trace (in reverse order!) of the dialogue history so far.

Given the goal expect_input, the next step is for the agent to generate appropriate wants to satisfy the goal:

{(Believes (Want (Expect_input_P)) (Goals (Expect_input Make_decision Survive)) (Committed (Respond Teach_User) (T_Goals (Address_now)) (D_Goals Make_decision)) (D_Committed (Agree_Lifestyle_objectives)) (Worldstate (Done Make_response) (Goalreached Make_response Construct_response Add_to_response) Review_tactic Address_T_Goals Select_T_Goals Sort_T_Goals Process_Pd) (Active) (Ddone Agree_Ps_objectives Eliminate_Fail_Ps) (Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree) (System_Turn User_EXISTS))

... and from that the preferred (in this case the only possible) commitment:

{(Believes <>)

301
Appendix B  Design, Decisions and Dialogue

(Wants) (Goals Expect_Input Make_Decision Survive)
(Committed Expect_Input_P Respond Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Make_Response Construct_Response Add_To_Response Review_Tactic
Address_T_Goals Select_T_Goals Sort_T_Goals Process_Pd)
(Active) (Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree)
(System_Turn User_Exists))

The effect of this basic act is to note the change of turn, and changes in the done and
goalreached lists:
((Believes ->)
(Wants) (Goals Make_Decision Survive) (Committed Respond Teach_User)
(T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Expect_Input_P) (Goalreached Process_Pd) (Active)
(Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree) User_Turn
User_Exists))

This causes the higher level act of responding to be done, and the system to forget
about the lower level goals reached:
((Believes <-)
(Wants) (Goals Make_Decision Survive) (Committed Teach_User)
(T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Respond Expect_Input_P) (Goalreached Process_Pd) <->)
(Active) (Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree) User_Turn
User_Exists))

The cycle proceeds as follows (main changes highlighted):
((Believes ->)
(Wants Listen Respond Finish_P) (Goals Make_Decision Survive)
(Committed Teach_User) (T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Respond Expect_Input_P) (Goalreached Process_Pd) ->))

((Believes <->)
(Wants Make_Decision Survive) (Committed Listen Teach_User)
(T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Process_Pd) <->))

((Believes <->)
(Wants Await_Input_P)
(Goals Await_Input Split_Input Store_Old_Ts Process_Sentences Make_Inferences
Update_State Make_Decision Survive)
(Committed Listen Teach_User) (T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Process_Pd) <->))

((Believes <->)
(Wants)
(Goals Await_Input Split_Input Store_Old_Ts Process_Sentences Make_Inferences Update_State
Make_Decision Survive)
(Committed Await_Input_P Listen Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Process_Pd) <->))

At this point the user enters:
I believe an objective is price. I believe an objective is sunroof.
((Believes ->)
(Dh (Impart Believes Objectives Price Impart Believes Objectives Sunroof)
(Impart D_Committed Agree_Lifestyle_Objectives Impart Believes Objectives

302
Design, Decisions and Dialogue

Appendix B

Running_Cost Elicit Believes Objectives)
(Import Believes Tactic Agree_Lifestyle_Objectives)
(Import _D_Committed Agree_Ps_Objectives Import Believes Objectives Safety
Justification In_Ps Import Believes Objectives Comfort Justification In_Ps
Elicit Believes Tactic)
(Import Believes Agree)
(Import Believes Tactic Agree_Ps_Objectives Justification
Take_Problem_Spec Into Account Elicit Believes Tactic Agree_Ps_Objectives
Agree)
(Elicit Believes Justification)
(Import _D_Committed Eliminate_Fail_Ps Import Believes Alternatives D Reject
Justification Price Import Believes Tactic Agree_Ps_Objectives Elicit
Believes Tactic Agree_Ps_Objectives Agree)
(Import Believes Tactic Eliminate_Fail_Ps) (Elicit Believes Tactic))
<>
(Wants)
(Goals Split_Input Store_Old_Ts Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Listen Teach_User) (T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Await_Input_P) (Goalreached Await_Input Process_Pd) <>)

((Believes <>)
(Wants Split_Input_P)
(Goals Split_Input Store_Old_Ts Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Listen Teach_User) (T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Await_Input_P) (Goalreached Await_Input Process_Pd) <>))

As part of split_input_p, the system echoes the user's input:
user: (Import Believes Objectives Price Import Believes Objectives Sunroof)

The (edited) agent state is then:

((Believes <> (Sentences (Import Believes Objectives Price) (Import Believes Objectives
Sunroof)) <>)
(Wants)
(Goals Store_Old_Ts Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Listen Teach_User) (T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Split_Input_P) (Goalreached Split_Input Await_Input Process_Pd) <>))

((Believes <>)
(Wants Store_Old_Ts_P)
(Goals Store_Old_Ts Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Listen Teach_User) (T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Split_Input_P) (Goalreached Split_Input Await_Input Process_Pd) <>))

((Believes <>)
(Wants)
(Goals Store_Old_Ts_Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Store_Old_Ts_P Listen Teach_User) (T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Split_Input Await_Input Process_Pd) <>))

((Believes <>)
(Told (Believes Objectives) (Believes Objectives Running_Cost)
 (D_Committed Agree_Lifestyle_Objectives)
(Believes Tactic Agree_Lifestyle_Objectives)
(Topics_tack (Believes Objectives) (Believes Objectives Running_Cost)
(D_Committed Agree_Lifestyle_Objectives)
(Believes Tactic Agree_Lifestyle_Objectives)))

303
Appendix B

Design, Decisions and Dialogue

(Wants)
(Goals Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Listen Teach_User) (T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Store_Old Ts_P)
(Goalreached Store_Old Ts Split_Input Await_Input Process_Pd) <>)

((Believes<>)
(Wants Process_Comprehensible Process_Incomprehensible Finish_P)
(Goals Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Listen Teach_User) (T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Store_Old Ts_P)
(Goalreached Store_Old Ts Split_Input Await_Input Process_Pd) <>)

((Believes<>)
(Wants)
(Goals Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Store_Old Ts Split_Input Await_Input Process_Pd) (Active)
(Done Agree Ps Objectives Eliminate Fail Ps)
(Recent Elicit_Belief Impart New Belief Elicit_Belief Elicit_Agree) User_Turn
User_Exists)

((Believes<>)
(Wants)
(Goals Update Ts Understand_Loc F Understand_Purpose Note_Ambiguities Assess_Pc
Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Store_Old Ts Split_Input Await_Input Process_Pd) <>)

((Believes<>)
(Wants)
(Goals Update Ts Understand_Loc F Understand_Purpose Note_Ambiguities Assess_Pc
Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Update Ts_P Process_Comprehensible Listen Teach_User)
(T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Store_Old Ts Split_Input Await_Input Process_Pd) <>)

((Believes<>)
(Topicstack (Believes Objectives Price) (Believes Objectives Running_Cost)
(D_Committed Agree_Lifestyle_Objectives)
(Believes Tactic Agree_Lifestyle_Objectives)))
(Wants)
(Goals Understand_Loc_F Understand_Purpose Note_Ambiguities Assess_Pc
Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Update Ts_P)
(Goalreached Update Ts Store_Old Ts Split_Input Await_Input Process_Pd) <>)

((Believes<>)
(Wants Understand Impart_P Understand_Elicit_P)
(Goals Understand_Loc_F Understand_Purpose Note_Ambiguities Assess_Pc
Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Update Ts_P)
(Goalreached Update Ts Store_Old Ts Split_Input Await_Input Process_Pd) <>)

((Believes<>)
(Wants)

304
(Goals Understand_Loc_F Understand_Purpose Note_Ambiguities Assess_Pc Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Understand_Impart_P Process_Comprehensible Listen Teach_User)
(T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd) \(\Rightarrow\))

\((Believes\).
User
(Believes
(Tactic (Eliminate_Fail_Ps) (Agree_Ps_Objectives (Agree))
(Agree_Lifestyle_Objectives))
(Objectives (Safety) (Comfort) (Price)))
\(\Rightarrow\)
(Wants)
(Goals Understand_Purpose Note_Ambiguities Assess_Pc Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Understand_Impart_P)
(Goalreached Understand_Loc_F Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd) \(\Rightarrow\))

\((Believes\).
(Wants)
(Goals Understand_Purpose Note_Ambiguities Assess_Pc Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Understand_Impart_P)
(Goalreached Understand_Loc_F Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd) \(\Rightarrow\))

\((Believes\).
(Wants)
(Goals Understand_Purpose Note_Ambiguities Assess_Pc Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Understand_Purpose Understand_Loc_F Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd) \(\Rightarrow\))

In this case, understand_purpose does nothing.

\((Believes\).
(Wants)
(Goals Note_Ambiguities Assess_Pc Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Understand_Purpose_P)
(Goalreached Understand_Purpose Understand_Loc_F Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd)\(\Rightarrow\))

\((Believes\).
(Wants Note_Ambiguities_P Abstain_P)
\(\Rightarrow\)

\((Believes\).
(Wants)
(Goals Note_Ambiguities Assess_Pc Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Abstain_P Process_Comprehensible Listen Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Understand_Purpose Understand_Loc_F Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd)\(\Rightarrow\))

\((Believes\).

Appendix B Design, Decisions and Dialogue

(Wants)
(Goals Asses_Pc Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User) (T_Goals (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Abstain_P))
(Goalreached Note_Ambiguities Understand_Purpose Understand_Loc_F Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd)<>

((Believes<>)
(Wants Tidy_Ts_P Note_Agree_P Note_Disagree_P Note_Not-Known_P Note.Expectation_P)
<>
)

((Believes<>)
(Wants)
(Goals Asses_Pc Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Note_Agree_P Process_Comprehensible Listen Teach_User)
(T_Goals (Address_Now)) (D_Goals Make_Decision)
(D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done))
(Goalreached Note_Ambiguities Understand_Purpose Understand_Loc_F Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd)<>

((Believes)
(Problem (Kws Family Only_Car) (Need (Price Max 8000)) (Wants Safety Comfort))
(Objectives (Safety (Justification (In_Ps)) (Agree))
(Comfort (Justification (In_Ps)) (Agree)) (Running_Cost (Price (Agree))))
<>
)

((Wants)
(Goals Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User)
(T_Goals (Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Note_Agree_P))
(Goalreached Note_Ambiguities Understand_Purpose Understand_Loc_F Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd)<>

((Believes<>)
(Wants Forget_Sentence_P)
<>
)

((Believes<>)
(Wants)
(Goals Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Forget_Sentence_P Process_Comprehensible Listen Teach_User)
(T_Goals (Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done))
(Goalreached Asses_Pc Note_Ambiguities Understand_Purpose Understand_Loc_F Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd)<>

((Believes<>)
(Sentences (Impart Believes Objectives Sunroof)))
<>

((Wants)
(Goals Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User)
(T_Goals (Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Forget_Sentence_P))
(Goalreached Forget_Sentence Asses_Pc Note_Ambiguities Understand_Purpose Understand_Loc_F Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd)<>

((Believes<>)
(Wants)
(Goals Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Listen Teach_User)

306
(T_Goals (Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Process_Comprehensible Forget_Sentence_P)
(Goalreached Store_Old_Ts Split_Input Await_Input Process_Pd) <>))

((Believes<>)
(Wants Process_Comprehensible Process_Incomprehensible Finish_P)
<>)

((Believes<>)
(Wants)
(Goals Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User)
(T_Goals (Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Store_Old_Ts Split_Input Await_Input Process_Pd) <>))

((Believes<>)
(Wants Update_Ts_P)
(Goals Update_Ts Understand_Loc_F Understand_Purpose Note_Ambiguities Assess_Pc
Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Update_Ts_P Process_Comprehensible Listen Teach_User)
(T_Goals (Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Store_Old_Ts Split_Input Await_Input Process_Pd) <>))

((Believes<>)
(Wants)
(Goals Understand_Loc_F Understand_Purpose Note_Ambiguities Assess_Pc
Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Update_Ts_P Process_Comprehensible Listen Teach_User)
(T_Goals (Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Store_Old_Ts Split_Input Await_Input Process_Pd) <>))

((Believes<>)
(Topicstack (Believes Objectives Sunroof)) (Believes Objectives Price)
(Believes Objectives Running_Cost) (D_Committed Agree_Lifestyle_Objectives)
(Believes Tactic Agree_Lifestyle_Objectives))

(Wants)
(Goals Understand_Loc_F Understand_Purpose Note_Ambiguities Assess_Pc
Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Process_Comprehensible Listen Teach_User)
(T_Goals (Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd) <>))

((Believes<>)
(Wants Understand_Impart_P Understand_Elicit_P)
<>)

((Believes<>)
(Wants)
(Goals Understand_Loc_F Understand_Purpose Note_Ambiguities Assess_Pc
Forget_Sentence Process_Sentences Make_Inferences Update_State Make_Decision Survive)
(Committed Understand_Impart_P Process_Comprehensible Listen Teach_User)
(T_Goals (Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Update_Ts Store_Old_Ts Split_Input Await_Input Process_Pd) <>))

((Believes<>)
(User (Believes
(Tactic Eliminate_Fail_Ps) (Agree_Ps_Objectives (Agree)) (Agree_Lifestyle_Objectives)
(Objectives (Safety) (Comfort) (Price) (Sunroof))))

307
Appendix B

Design, Decisions and Dialogue

<>
(Wants)
(Goals Understand Purpose Note Ambiguities Assess Pc Forget Sentence
Process Sentences Make Inferences Update State Make Decision Survive)
(Committed Process Comprehensible Listen Teach User)
(T_Goals (Explicit Agree (Believes Objectives Price)) (Address Now))
(D_Goals Make Decision) (D_Committed Agree Lifestyle Objectives)
(Worldstate (Done Understand Impart P)
(Goalreached Understand Loc F Update Ts Store Old Ts Split Input Await Input Process Pd<>))

((Believes<>)
(Wants Understand Purpose P)
<>)

((Believes<>)
(Wants)
(Goals Understand Purpose Note Ambiguities Assess Pc Forget Sentence
Process Sentences Make Inferences Update State Make Decision Survive)
(Committed Understand Purpose P Process Comprehensible Listen Teach User)
(T_Goals (Explicit Agree (Believes Objectives Price)) (Address Now))
(D_Goals Make Decision) (D_Committed Agree Lifestyle Objectives)
(Worldstate (Done)
(Goalreached Understand Loc F Update Ts Store Old Ts Split Input Await Input Process Pd<>))

((Believes<>)
(Wants Note Ambiguities P Abstain P)
<>)

((Believes<>)
(Wants)
(Goals Note Ambiguities Assess Pc Forget Sentence Process Sentences
Make Inferences Update State Make Decision Survive)
(Committed Abstain P Process Comprehensible Listen Teach User)
(T_Goals (Explicit Agree (Believes Objectives Price)) (Address Now))
(D_Goals Make Decision) (D_Committed Agree Lifestyle Objectives)
(Worldstate (Done)
(Goalreached Understand Purpose Understand Loc F Update Ts Store Old Ts Split Input Await Input Process Pd<>))

((Believes<>)
(Wants)
(Goals Note Ambiguities Assess Pc Forget Sentence Process Sentences
Make Inferences Update State Make Decision Survive)
(Committed Abstain P Process Comprehensible Listen Teach User)
(T_Goals (Explicit Agree (Believes Objectives Price)) (Address Now))
(D_Goals Make Decision) (D_Committed Agree Lifestyle Objectives)
(Worldstate (Done Understand Purpose P)
(Goalreached Note Ambiguities Understand Purpose Understand Loc F Update Ts Store Old Ts Split Input Await Input Process Pd<>))

((Believes<>)
(Wants Tidy Ts P Note Agree P Note Disagree P Note Not Known P
Note Expectation P)
<>)

((Believes<>)
(Wants)
(Goals Assess Pc Forget Sentence Process Sentences Make Inferences Update State
Make Decision Survive)

308
Appendix B

Design, Decisions and Dialogue

(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))

(Explicit_Agree (Believes Objectives Price)) (Address_Now))

(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)

(Worldstate (Done)

(Goalreached Store_Old_Ts Split_Input Await_Input Process_Pd) <>))

(Believes<>)

(Wants) (Goals Make_Inferences Update_State Make_Decision Survive)

(Committed Listen Teach_User)

(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))

(Explicit_Agree (Believes Objectives Price)) (Address_Now))

(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)

(Worldstate (Done)

(Goalreached Process_Sentences Store_Old_Ts Split_Input Await_Input Process_Pd) <>))

(Believes<>)

(Wants Make_Inferences_P)

<>)

(Believes<>)

(Wants) (Goals Make_Inferences Update_State Make_Decision Survive)

(Committed Make_Inferences_P Listen Teach_User)

(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))

(Explicit_Agree (Believes Objectives Price)) (Address_Now))

(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)

(Worldstate (Done)

(Goalreached Process_Sentences Store_Old_Ts Split_Input Await_Input Process_Pd) <>))

(Believes)

(Problem (Kws Family Only_Car) (Need (Price Max 8000)) (Wants Safety Comfort))

(Objectives (Safety (Justification (In_Ps)) (Agree))

(Comfort (Justification (In_Ps)) (Agree)) (Running_Cost (Agree))

(Price (Agree)) (Sunroof (Not) (Disagree))

(Working_Objectives (Running_Cost (Parameter) (Scale) (Weight) (Alternatives))

(Comf (Parameter) (Scale) (Weight) (Alternatives))

(Safety (Parameter) (Scale) (Weight) (Alternatives))

(Alternatives (D (Reject (Justification (Price))))))

(Tactic (Eliminate_Fail_Ps (Agree))

(Agree_Ps_Objectives (Justification (Take_Problem_Spec_Into_Account)) (Agree))

(Agree_Lifestyle_Objectives (Agree)))

(User

(Tactic (Eliminate_Fail_Ps) (Agree_Ps_Objectives (Agree))

(Agree_Lifestyle_Objectives))

(Objectives (Safety) (Comfort) (Price) (Sunroof) (Running_Cost)))

(D_Committed (Agree_Lifestyle_Objectives))

(Dh <>)

(Sentences)

(Tsold (Believes Objectives) (Believes Objectives Running_Cost)

(D_Committed Agree_Lifestyle_Objectives)

(Believes Tactic Agree_Lifestyle_Objectives))

(Toplistack (Believes Objectives Sunroof) (Believes Objectives Price)))

(Wants) (Goals Update_State Make_Decision Survive)

(Committed Listen Teach_User)

(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))

(Explicit_Agree (Believes Objectives Price)) (Address_Now))

(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)

(Worldstate (Done Make_Inferences_P)

(Goalreached Make_Inferences Process_Sentences Store_Old_Ts Split_Input

Await_Input Process_Pd)

(Active) (Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)

(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree) User_Turn

User_Exists))

(Believes<>)

(Wants Update_State_P) (Goals Update_State Make_Decision Survive)

<>)

(Believes<>)

(Wants) (Goals Update_State Make_Decision Survive)

310
Design, Decisions and Dialogue

Appendix B

(Committed Update_State_P Listen Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))
(Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Make_Inferences Process_Sentences Store_Old_Ts Split_Input Await_Input Process_Pd)<>))

((Believes<>)
(Do Nil (Impart Believes Objectives Price Impart Believes Objectives Sunroof)<>))

(Wants) (Goals Make_Decision Survive) (Committed Listen Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))
(Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Update_State_P)
(Goalreached Update_State Make_Inferences Process_Sentences Store_Old_Ts Split_Input Await_Input Process_Pd)
(Active) (Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree)
(System_Turn User_Exists))

((Believes<>)
(Wants Listen Respond Finish_P)
<>)

((Believes<>)
(Wants) (Goals Make_Decision Survive) (Committed Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))
(Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Process_Pd) (Active)
(Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree)
(System_Turn User_Exists))

((Believes<>)
(Wants Sort_T_Goals_P)
(Goals Sort_T_Goals Select_T_Goals Address_T_Goals Review_Tactic Add_To_Response Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Committed Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))
(Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Process_Pd) <>))

((Believes<>)
(Wants)
(Goals Sort_T_Goals Select_T_Goals Address_T_Goals Review_Tactic Add_To_Response Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Committed Sort_T_Goals_P Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))
(Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Process_Pd) <>))

((Believes<>)
(Wants)
(Goals Select_T_Goals Address_T_Goals Review_Tactic Add_To_Response Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Committed Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))
(Explicit_Agree (Believes Objectives Price)) (Address_Now))

311
Appendix B

Design, Decisions and Dialogue

((Believes<>)
(Wants Select_T_Goals_P)
<>)

((Believes>)
(Wants)
((Goals Select_T_Goals Address_T_Goals Review_Tactic Add_To_Response
Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Committed Select_T_Goals_P Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof))
(Explicit_Agree (Believes Objectives Price)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Sort_T_Goals Process_Pd) (Active)
(Done Agree_Ps Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree)
(System_Turn User_Exists))

((Believes>)
(Wants)
((Goals Address_T_Goals Review_Tactic Add_To_Response Construct_Response
Make_Response Expect_Input Make_Decision Survive)
(Committed Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof))
(Explicit_Agree (Believes Objectives Price))
(Address_Now (Explicit_Agree (Believes Objectives Price)))
(Resolve_Conflict (Believes Objectives Sunroof)))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Select_T_Goals Sort_T_Goals Process_Pd) <>)

((Believes>)
(Wants Address_A_T_Goal Finish_P)
<>)

((Believes>)
(Wants)
((Goals Address_T_Goals Review_Tactic Add_To_Response Construct_Response
Make_Response Expect_Input Make_Decision Survive)
(Committed Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof))
(Explicit_Agree (Believes Objectives Price))
(Address_Now (Explicit_Agree (Believes Objectives Price)))
(Resolve_Conflict (Believes Objectives Sunroof)))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Select_T_Goals Sort_T_Goals Process_Pd <>))

((Believes>)
(Wants Active_T_Goal P)
((Goals Active_T_Goal_T_Goal Ts T_Goal_Goalist Address_T_Goals Review_Tactic
Add_To_Response Construct_Response Make_Response Expect_Input Make_Decision
Survive)
(Committed Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof))
(Explicit_Agree (Believes Objectives Price))
(Address_Now (Explicit_Agree (Believes Objectives Price)))
(Resolve_Conflict (Believes Objectives Sunroof)))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Select_T_Goals Sort_T_Goals Process_Pd <>))

((Believes>)
(Wants)
((Goals Active_T_Goal_T_Goal Ts T_Goal_Goalist Address_T_Goals Review_Tactic
Add_To_Response Construct_Response Make_Response Expect_Input Make_Decision
Survive)
(Committed Active_T_Goal_P Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof))
(Explicit_Agree (Believes Objectives Price)))
(Address_Now (Explicit_Agree (Believes Objectives Price)))
(Resolve_Conflict (Believes Objectives Sunroof)))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Select_T_Goals Sort_T_Goals Process_Pd)
(Active) (Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree)
(System_Turn User_Exists))

((Believes<>)
(Wants)
(Goals T_Goal_Ts T_Goal_Goalist Address_T_Goals Review_Tactic Add_To_Response
Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Committed Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))
(Explicit_Agree (Believes Objectives Price)))
(Address_Now (Explicit_Agree (Believes Objectives Price)))
(Resolve_Conflict (Believes Objectives Sunroof)))))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Active_T_Goal_P)
(Goalreached Active_T_Goal Select_T_Goals Sort_T_Goals Process_Pd)
(Active (Explicit_Agree (Believes Objectives Price)))
(Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree)
(System_Turn User_Exists))

((Believes<>)
(Wants T_Goal_Ts_P)
<>)

((Believes<>)
(Wants)
(Goals T_Goal_Ts T_Goal_Goalist Address_T_Goals Review_Tactic Add_To_Response
Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Committed T_Goal_Ts_P Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))
(Explicit_Agree (Believes Objectives Price)))
(Address_Now (Explicit_Agree (Believes Objectives Price)))
(Resolve_Conflict (Believes Objectives Sunroof)))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Active_T_Goal Select_T_Goals Sort_T_Goals Process_Pd)
(Active (Explicit_Agree (Believes Objectives Price)))
(Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree)
(System_Turn User_Exists))

((Believes<>)
(Topicstack (Believes Objectives Price) (Believes Objectives Sunroof))
(Wants)
(Goals T_Goal_Goalist Address_T_Goals Review_Tactic Add_To_Response
Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Committed Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))
(Explicit_Agree (Believes Objectives Price)))
(Address_Now (Explicit_Agree (Believes Objectives Price)))
(Resolve_Conflict (Believes Objectives Sunroof)))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done T_Goal_Ts_P)
(Goalreached T_Goal_Ts Active_T_Goal Select_T_Goals Sort_T_Goals Process_Pd)
(Active (Explicit_Agree (Believes Objectives Price)))
(Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree)
(System_Turn User_Exists))

((Believes<>)
(Wants T_Goal_Goalist_P)
<>)

((Believes<>)
(Wants)
Appendix B Design, Decisions and Dialogue

(Goals T_Goal_Goalist Address_T_Goals Review_Tactic Add_To_Response
Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Commited T_Goal_Goalist P Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof))
(Explicit_Agree (Believes Objectives Price))
(Address_now (Explicit_Agree (Believes Objectives Price)))
(Resolve_Conflict (Believes Objectives Sunroof)))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached T_Goal_Goalist T_Goal_Goalist T_Goal_Goalist T_Goal
Select_T_Goals Sort_T_Goals Process_Pd)
(Active (Explicit_Agree (Believes Objectives Price)))
(D_done Agree_Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree)
(System_Turn User_Exists))

((Believes<>)
(Wants)
(Goals Explicit_Agree Address_T_Goals Review_Tactic Add_To_Response
Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Commited Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))
(Explicit_Agree (Believes Objectives Price))
(Address_now (Resolve_Conflict (Believes Objectives Sunroof))))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached T_Goal_Goalist T_Goal_Goalist T_Goal_Goalist T_Goal
Select_T_Goals Sort_T_Goals Process_Pd)
(Active (Explicit_Agree (Believes Objectives Price)))
(D_done Agree_Objectives Eliminate_Fail_Ps)
(Recent Elicit_Belief Impart_New_Belief Elicit_Belief Elicit_Agree)
(System_Turn User_Exists))

((Believes<>)
(Wants)
(Goals Explicit_Agree Address_T_Goals Review_Tactic Add_To_Response
Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Commited Imp_Ag_Justn_P Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)))
(Explicit_Agree (Believes Objectives Price))
(Address_now (Resolve_Conflict (Believes Objectives Sunroof))))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached T_Goal_Goalist T_Goal_Goalist T_Goal_Goalist T_Goal
Select_T_Goals Sort_T_Goals Process_Pd)
(Active (Explicit_Agree (Believes Objectives Price)))
(D_done Agree_Objectives Eliminate_Fail_Ps)
(Recent Imp_Ag_Justn_P Elicit_Belief Impart_New_Belief Elicit_Belief)
(System_Turn User_Exists))

((Believes)
(Problem (Kws Family Only Car) (Need (Price Max 8000)) (Wants Safety Comfort))
(Objectives (Safety (Justification (In Ps)) (Agree))
(Comfort (Justification (In Ps)) (Agree)) (Running_Cost (Agree))
(Price (Agree) (Justification (Not Rich)) (Sunroof) (Not) (Disagree))
(Working_Objectives (Running_Cost (Parameter) (Scale)) (Weight) (Alternatives))
(Comfort (Parameter) (Scale) (Weight) (Alternatives))
(Safety (Parameter) (Scale) (Weight) (Alternatives))
(Alternatives (D) (Reject (Justification (Price))))
(Tactic (Eliminate_Fail_Ps (Agree))
(Justification (Take_Problem_Spec_Into_Account)) (Agree))
(Agree_Lifestyle_Objectives (Agree)))
(User)
(Believes)
(Tactic (Eliminate_Fail_Ps) (Agree_Ps_Objectives (Agree)) (Agree_Lifestyle_Objectives)
(Objectives (Safety) (Comfort) (Price) (Sunroof) (Running_Cost)))

314
Design, Decisions and Dialogue

(D_Committed (Agree_Lifestyle_Objectives))
(Dh <>)
(Sentences (Impart Believes Objectives Price Justification Not_Rich)
  (Import Believes Objectives Price Agree))
(Tsold (Believes Objectives) (Believes Objectives Running_Cost)
  (D_Committed Agree_Lifestyle_Objectives)
  (Believes Tactic Agree_Lifestyle_Objectives))
(Topicstack (Believes Objectives Price Justification Not_Rich)
  (Believes Objectives Sunroof))
(Wants)
(Goals Address_T_Goals Review_Tactic Add_To_Response Construct_Response
  Make_Response Expect_Input Make_Decision Survive)
(Committed Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof))
  (Address_Now (Resolve_Conflict (Believes Objectives Sunroof))))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Imp_Ag_Imp_Justn_P)
  (Goalreached Explicit_Agree T_Goal_Goalist T_Goal Ts Active_T_Goal
   Select_T_Goals Sort_T_Goals Process_Pd)
  (Active) (Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
  (Recent Imp_Ag_Imp_Justn_P Elicit_Belief Impart_New_Belief Elicit_Belief)
  System_Turn User_Exists))
(((Believes<>)
(Wants)
(Goals Address_T_Goals Review_Tactic Add_To_Response Construct_Response
  Make_Response Expect_Input Make_Decision Survive)
(Committed Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof))
  (Address_Now (Resolve_Conflict (Believes Objectives Sunroof))))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Address_A_T_Goal Imp_Ag_Imp_Justn_P)
  (Goalreached Select_T_Goals Sort_T_Goals Process_Pd) (Active)
  (Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
  (Recent Imp_Ag_Imp_Justn_P Elicit_Belief Impart_New_Belief Elicit_Belief)
  System_Turn User_Exists))
(((Believes<>)
(Wants Address_A_T_Goal Finish_P)
<>)
(((Believes<>)
(Wants)
(Goals Address_T_Goals Review_Tactic Add_To_Response Construct_Response
  Make_Response Expect_Input Make_Decision Survive)
(Committed Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof))
  (Address_Now (Resolve_Conflict (Believes Objectives Sunroof))))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Select_T_Goals Sort_T_Goals Process_Pd))
(((Believes<>)
(Wants Active_T_Goal_P)
<>)
(((Believes<>)
(Wants)
(Goals Active_T_Goal T_Goal Ts T_Goal_Goalist Address_T_Goals Review_Tactic
  Add_To_Response Construct_Response Make_Response Expect_Input Make_Decision
  Survive)
(Committed Active_T_Goal_P Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof))
  (Address_Now (Resolve_Conflict (Believes Objectives Sunroof))))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Select_T_Goals Sort_T_Goals Process_Pd)
  (Active) (Ddone Agree_Ps_Objectives Eliminate_Fail_Ps)
  (Recent Imp_Ag_Imp_Justn_P Elicit_Belief Impart_New_Belief Elicit_Belief)
  System_Turn User_Exists))
(((Believes<>)

315
(Wants)
(Goals Resolve_Conflict Address_T_Goals Review_Tactic Add_To_Response
ConstrucResponse Make_Response Expect_Input Make_Decision Survive)
(Committed Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)) (Address_Now))
(D_Goals Make_Decision (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done T_Goal_Goalist Ts) (Goalreached T_Goal_Goalist Ts) Active_T_Goal Select_T_Goals
Sort_T_Goals Process_Pd)
(Active (Resolve_Conflict (Believes Objectives Sunroof)))
(Done Agree_Ps_Objectives Eliminate_Fail_Ps)
(Recent Imp_Ag_Imp_Justn_P Elicit_Belief Impart_New_Belief Elicit_Belief
System_Turn User_Exists))

((Believes<>)
(Wants
(Goals Resolve_Conflict Address_T_Goals Review_Tactic Add_To_Response
ConstrucResponse Make_Response Expect_Input Make_Decision Survive)
(Committed Imp_Dis_impj_Elj_P Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)) (Address_Now))
(D_Goals Make_Decision (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached T_Goal_Goalist Ts) (Active_T_Goal Select_T_Goals
Sort_T_Goals Process_Pd)
(Active (Resolve_Conflict (Believes Objectives Sunroof)))
(Done Agree_Ps_Objectives Eliminate_Fail_Ps)
(Recent Imp_Dis_impj_Elj_P Imp_Ag_Imp_Justn_P Elicit_Belief Impart_New_Belief
System_Turn User_Exists))

Note that addressing the resolve_conflict t_g Goal does not cause the t_goal to be
removed in the way that the explicit_agree t_g Goal did:
((Believes
(User
(Believes
(Tactic (Eliminate_Fail_Ps) Agree_Ps_Objectives Agree)
(Agree_Lifestyle_Objectives Agree))
(Objectives (Safety (Cost) (Price) (Justification (Resistant)))
(Price (Agree) (Cost) (Weight) (Alternatives)))
(Tactic (Take_Problem_Spec_Into_Account) Agree)
(Agree_Lifestyle_Objectives Agree))
(Believes
(Tactic (Eliminate_Fail_Ps) (Justification (Cost) (Price) (Resistant)))
(Agree_Lifestyle_Objectives Agree)
(Objectives (Safety (Cost) (Price) (Justification (Resistant)))
(Price (Agree) (Cost) (Weight) (Alternatives)))
(Tactic (Take_Problem_Spec_Into_Account) Agree)
(Agree_Lifestyle_Objectives Agree))
(Dh <>)
(Sentences (Elicit Believes Objectives Sunroof Justification)
(Impart Believes Objectives Sunroof Not Justification Tend_To_Leak))
(Impart Believes Objectives Sunroof Disagree)
(Impart Believes Objectives Price Justification Not_Rich)
(Impart Believes Objectives Price Agree)
(Tsold (Believes Objectives) (Believes Objectives Running_Cost)
(D_Committed Agree_Lifestyle_Objectives)
(Believes Tactic Agree_Lifestyle_Objectives)
(Topicstack (Believes Objectives Sunroof Justification)
(Believes Objectives Price Justification Not_Rich)))
(Wants)
(Goals Address_T_Goals Review_Tactic Add_To_Response Construct_Response
Make_Response Expect_Input Make_Decision Survive)
(Committed Address_A_T_Goal Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Imp_Dis_Impj_Ell_P))
(Goalreached Resolve_Conflict T_Goal_Goalset T_Goal Ts Active_T_Goal
  Select_T_Goals Sort_T_Goals Process_Pd)
(Active (Resolve_Conflict (Believes Objectives Sunroof)))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Address_A_T_Goal Imp_Dis_Impj_Ell_P))
(Goalreached Select_T_Goals Sort_T_Goals Process_Pd)
(Active (Resolve_Conflict (Believes Objectives Sunroof)))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done) (Goalreached Select_T_Goals Sort_T_Goals Process_Pd))
(Wants Address_A_T_Goal Finish_P)
<>
(Believes<>)
(Wants Address_A_T_Goal Finish_P)
<>
(Believes<>)
(Wants Address_A_T_Goal Finish_P)
<>
(Believes<>)
(Wants Address_A_T_Goal Finish_P)
<>
(Believes<>)
(Wants Address_A_T_Goal Finish_P)
<>
((Believes<>)
(Wants Elicit_Belief Elicit_Agree Impart_New_Belief Do_New_Action Elicit_Action Elicit_Proposal Impart_Proposal Finish_P)
<>)

((Believes<>)
(Wants
(Goals Add_To_Response Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Committed Finish_P Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)) (Address_NOW))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Add_To_Response Review_Tactic Address_T_Goals Select_T_Goals Sort_T_Goals Process_Pd)<>))

((Believes<>)
(Wants
(Goals Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Committed Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)) (Address_NOW))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Add_To_Response Review_Tactic Address_T_Goals Select_T_Goals Sort_T_Goals Process_Pd)<>))

((Believes<>)
(Wants Construct_Response_P)
<>)

((Believes<>)
(Wants
(Goals Construct_Response Make_Response Expect_Input Make_Decision Survive)
(Committed Construct_Response_P Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)) (Address_NOW))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done)
(Goalreached Add_To_Response Review_Tactic Address_T_Goals Select_T_Goals Sort_T_Goals Process_Pd)<>)

((Believes<>)
(Dh
(Impart Believes Objectives Price Agree Impart Believes Objectives Price Justification Not_Rich Impart Believes Objectives Sunroof Disagree Impart Believes Objectives Sunroof Not Justification Tend_To_Leak Believes Objectives Sunroof Justification)
(Impart Believes Objectives Price Impart Believes Objectives Sunroof)
(Impart D_Committed Agree_Lifestyle_Objectives Impart Believes Objectives Running_Cost Elicit Believes Objectives)
(Impart Believes Tactic Agree_Lifestyle_Objectives)
(Impart D_Committed Agree_Ps_Objectives Impart Believes Objectives Safety Justification In_Ps Impart Believes Objectives Comfort Justification In_Ps Elicit Believes Tactic)
(Impart Believes Agree)
(Impart Believes Tactic Agree_Ps_Objectives Justification Take_Problem_Spec_Into_Account Elicit Believes Tactic Agree_Ps_Objectives Agree)
(Elicit Believes Justification)
(Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject Justification Price Impart Believes Tactic Agree_Ps_Objectives Elicit Believes Tactic Agree_Ps_Objectives Agree)
(Impart Believes Tactic Eliminate_Fail_Ps) (Elicit Believes Tactic))
Appendix B: Design, Decisions and Dialogue

(Wants) (Goals Make_Response Expect_Input Make_Decision Survive)
(Committed Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done Construct_Response_P))
(Goalreached Construct_Response Add_To_Response Review_Tactic Address_T_Goals
Select_T_Goals Sort_T_Goals Process_Pd))
 Bewertungs
(Believes<>)
(Wants Make_Response_P)
<>

(Believes<>)
(Wants) (Goals Make_Response Expect_Input Make_Decision Survive)
(Committed Make_Response_P Respond Teach_User)
(T_Goals (Resolve_Conflict (Believes Objectives Sunroof)) (Address_Now))
(D_Goals Make_Decision) (D_Committed Agree_Lifestyle_Objectives)
(Worldstate (Done))
(Goalreached Construct_Response Add_To_Response Review_Tactic Address_T_Goals
Select_T_Goals Sort_T_Goals Process_Pd))

... and the system output is:

system: (Impart Believes Objectives Price Agree Impart Believes Objectives Price Justification Not_Rich Impart Believes Objectives Sunroof Disagree Impart Believes Objectives Sunroof Not Justification Tend_To_Leak Elicit Believes Objectives Sunroof Justification)

At this point the agent state is:

(Believes-
Problem (Kws Family Only_Car) (Need (Price Max 8000)) (Wants Safety Comfort))
(Objectives (Safety (Justification (In_Ps)) (Agree))
(Comfort (Justification (In_Ps)) (Agree)) (Running_Cost (Agree))
(Price (Agree) (Justification (Not_Rich)))
(Sunroof (Not (Justification (Tend_To_Leak))) (Disagree))
(Working_Objectives (Running_Cost (Parameter) (Scale) (Weight) (Alternatives))
(Comfort (Parameter) (Scale) (Weight) (Alternatives))
(Safety (Parameter) (Scale) (Weight) (Alternatives))
(Alternatives (D (Reject (Justification (Price))))
(Tactic (Eliminate_Fail_Ps (Agree))
(Agree_Ps_Objectives (Justification (Take_Problem_Spec_Into_Account)) (Agree))
(Agree_Lifestyle_Objectives (Agree))

User
(Believes
(Tactic (Eliminate_Fail_Ps (Agree_Ps_Objectives (Agree)) (Agree_Lifestyle_Objectives))
(Objectives (Safety) (Comfort) (Price) (Sunroof) (Running_Cost)))
(D_Committed (Agree_Lifestyle_Objectives))

(Dh
Impart Believes Objectives Price Agree Impart Believes Objectives Price
Justification Not_Rich Impart Believes Objectives Sunroof Disagree Impart
Believes Objectives Sunroof Not Justification Tend_To_Leak Elicit Believes
Objectives Sunroof Justification)

Impart Believes Objectives Price Impart Believes Objectives Sunroof

Impart D_Committed Agree_Lifestyle_Objectives Impart Believes Objectives
Running_Cost Elicit Believes Objectives)

Impart Believes Tactic Agree_Lifestyle_Objectives)

Impart D_Committed Agree_Ps_Objectives Impart Believes Objectives Safety
Justification In_Ps Impart Believes Objectives Comfort Justification In_Ps
Elicit Believes Tactic)

Impart Believes Agree)

Impart Believes Tactic Agree_Ps_Objectives Justification
Take_Problem_Spec_Into_Account Elicit Believes Tactic Agree_Ps_Objectives
Agree)
(Elicit Believes Justification)

Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject
Justification Price Impart Believes Tactic Agree_Ps_Objectives Elicit
Believes Tactic Agree_Ps_Objectives Agree)

Impart Believes Tactic Eliminate_Fail_Ps (Elicit Believes Tactic)

(Sentences)

(Tsold (Believes Objectives) (Believes Objectives Running_Cost)
(D_Committed Agree_Lifestyle_Objectives)

320
Appendix C: Evaluation of WOMBAT (§8)

C.1 Evaluation of WOMBAT prototype: user's notes

For every session (under all experimental conditions) users were asked to read through notes similar to the following at the beginning of the session. The precise wording varied between experimental setups, but the basic content remained the same.

Background

WOMBAT is a prototype for an Intelligent Educational System to support the development of decision making skills within engineering design. The type of decision making being addressed is selection between alternatives - for example, selecting between alternative conceptual designs prior to detail design. Several simplifications are made - for example, it is assumed that the properties of the alternatives are 'known' (probability is not dealt with), and the matter of independence of decision criteria is not dealt with.

Using WOMBAT

The decision problem facing you is one of selecting between 6 alternative cars. A minimal amount of information about each is included on the attached sheet. The system has access to a fuller database containing sufficient information about the alternatives to allow you and it to make a collaborative decision. The default problem is to select the best car for a family; it is to be their only car. They consider safety and comfort to be important, and can afford to spend up to £8000. This is expressed as a list of keywords (which the system uses to make inferences about additional decision criteria not explicitly stated in the problem spec.), a list of needs (absolute criteria which any solution must satisfy), and a list of wants (criteria which should be optimised). You can define a different problem if you prefer (though the problem cannot be changed in mid-stream), but the data on available alternatives is fixed in this prototype.

You have access to two aspects of the decision problem solving; the first of these is the selection of decision making tactics. The system has information about 17 decision making and data organising tactics which are appropriate at different stages in the decision process. These are listed on a following page, and in a pull-down menu. Some of these are fully automated so that you have no control over their execution - this is, after all, only a first prototype.

The second aspect of the problem which you can discuss fully is the identification of the criteria on which the decision is to be made. The system has information about 26 possible criteria, including justifications for including (or indeed for deciding not to include) each criterion. It refers to 'objectives', which are assumed to be the maximising or minimising of criteria - we assume it's obvious which! (for example, for the set problem of selecting a car, we assume that if the criterion is price, then the objective is to minimise it).

In this prototype, you cannot manipulate data in the matrix directly. All agreements are reached through the dialogue, which then updates the matrix accordingly.

Feel free to ask any questions or make any comments during the interaction. In particular, I would like you to consider the following questions after each of the system's utterances, though you need only explicitly comment about any utterances which strike you as being 'wrong' in any way:

• is the utterance sensible?
• is it relevant to what you believe you are talking about?
• does the utterance surprise you? If so, is it because of its 'form' (e.g. failing to answer a question), or because of the 'content' (you think it presents a strange line of argument)?

1possible ones are family, only_car, main_car, mechanic, diy, young_children, children_drive, infirm, many_occupants.
• would you have preferred a different (more appropriate or more helpful) response? If so, what?

Outline information on alternatives

<table>
<thead>
<tr>
<th>Car</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Sporty hatchback</td>
<td>£7895 on the road</td>
</tr>
<tr>
<td>B</td>
<td>Large boot, 1100cc</td>
<td>Runs on lead-free petrol, £6500 on the road</td>
</tr>
<tr>
<td>C</td>
<td>Compact, medium boot, good acceleration</td>
<td>£7500 on the road</td>
</tr>
<tr>
<td>D</td>
<td>£7950 list price, very safe</td>
<td>Takes either leaded or unleaded petrol (delivery charge £195)</td>
</tr>
<tr>
<td>E</td>
<td>£5980 list price, stereo and sunroof</td>
<td>Takes lead-free petrol (£225 extra charges)</td>
</tr>
<tr>
<td>F</td>
<td>£7095 on the road</td>
<td>Large family hatchback, seats 5</td>
</tr>
</tbody>
</table>
Information on tactics:
(The tactics marked with an asterisk (*) can be discussed, but have not been implemented. Only agree_lifestyle_objectives has been fully implemented so that you can discuss it completely.)

- list_alternatives - make a simple list of the available alternatives *
- list_objectives - make a list of the objectives agreed so far *
- matrix - set up a matrix of alternatives against objectives
- agree_ps_objectives - agree to include all the objectives listed in the problem specification
- agree_lifestyle_objectives - agree objectives based on the likely lifestyle of the purchaser of the car
- identify_parameters - agree parameters to measure each objective (e.g. time taken to accelerate from 0mph to 60mph, or time taken to cover 0.25 mile from standing start, as measure of acceleration)
- get_parameter_values - note values of parameters used to measure all objectives for all alternatives
- rank_objectives - list objectives in order of importance *
- eliminate_fail_ps - agree to eliminate all alternatives which fail to satisfy the absolute criteria specified in the problem specification
- eliminate_worst_on_1_objective - agree to eliminate the alternative which scores worst on one (to be agreed) objective (e.g. agree to eliminate the alternative with the worst reliability record) *
- select_best_on_1_objective - agree to select the alternative which scores best on one (to be agreed) objective *
- elimination_by_aspects - agree to eliminate all alternatives which score worse than an agreed limit on one (to be agreed) objective (e.g. agree to eliminate all alternatives with a worse fuel consumption than 35mpg) *
- pairwise_comparison - compare just two alternatives, and eliminate the one which scores worse on more (agreed) objectives
- scale_parameters - agree a scaling system for each parameter so that an excellent value of a criterion gets a score of 10, a dreadful value gets a score of 0, and an average value gets a score of 5 etc.
- assign_values - assign scaled values to all parameters for all objectives and all alternatives
- assign_weights - assign weighting values to all objectives
- do_worn - calculate the sum of the product of all weighting values and scaled values for each alternative, and select the alternative which gets the highest score.
- finish_tactic - finish the currently active tactic (currently only applicable to agree_lifestyle_objectives)
- continue_tactic - continue with the currently active tactic (currently only applicable to agree_lifestyle_objectives)
- end - quit, finish decision making, abandon exercise.
C.2 Subject profiles

All the subjects who took part in this evaluation are involved to a greater or lesser extent in either engineering education or the use of computers in education (or both!). The subject group includes 4 engineers from traditional (i.e. not distance teaching) universities (D, B, P and K) who have regular direct contact with undergraduates.

Subject L is a Senior Lecturer in Educational Technology, specialising in the evaluation of educational materials. Her work has involved very little contact with design education per se. She uses computers, but does not develop software herself. L used the original software and study set-up.

Subject R is a research student, working in the area of microworlds for use in science education. He has been involved in the development of software for use in teaching for several years. R used the original software and study set-up.

Subject J is a Senior Research Fellow in Design. While not extensively involved with teaching, he was involved with the development of the CADPAC software which supports OU course T363, described in §2.1.3. J used the original software and study set-up.

Subject M is a Lecturer in Material Science. He was involved with the development of the software to support OU course T201, described in §2.3.3, and has developed his own expert-system-based selection programs. He used the original software and study set-up.

Subject D is a lecturer in Aeronautical Engineering. He makes use of commercial spreadsheet software in his design teaching, but does not use computers extensively. In particular, he has not made use of computers to support selection between alternatives, and had not included formal teaching of decision processes in his design teaching. He had not used a Macintosh before, and so was not familiar with the style of the interface. D was the first subject to use the second version of the software and experimental design.

Subject C is an Applications Programmer. He was involved in the development of T201 (§2.3.3). He has been involved in Summer Schools using software for materials selection. C used the second version of the experiment.

Subject B is Professor in Mechanical Engineering. He is familiar with computing techniques (e.g. F.E. packages) and supervises students using computers in design, but his role as Head of School has limited the time he has spent using computers in recent years. B used the final software and study set-up.

Subject P is a Senior Lecturer in Mechanical Engineering. He is Director of the Computer Based Teaching Facility in Engineering at his institution and also of the (national) Computers in Teaching Initiative (CTI) Engineering Centre. He was Project Director for the PROSPECT project described in §2.3.3. He is not directly involved (in the 'hands on' sense) with software development, and neither is he directly involved with teaching Design per se to undergraduates. He used the final software and study set-up.

Subject K is a Senior Lecturer in Mechanical Engineering, and also Director of a SERC Engineering Design Centre. As such, his work involves use of computers, but he is not involved with software development. He used the final software and study set-up.

Subject G is a Professor of Engineering Design. He is active in research in mechatronics and neural nets. He used the final software and study set-up.
C.3 Traces of interactions

Full traces of all sessions, with the exception of that with P (which is included in §8.3.2), are presented here. The form of presentation is as encoded by the system (rather than being an English language interpretation); it is assumed that the interested reader will be able to understand this form!

L: Educationalist:

system: Elicit Believes Tactic
user: Impart Believes Tactic Agree_Ps_Objectives
system: Impart D_Committed Agree_Ps_Objectives Impart Believes Objectives Safety Justification In_Ps Impart Believes Objectives Comfort Justification In_Ps Impart Believes Tactic Eliminate_Fail_Ps Impart Believes Tactic Eliminate_Fail_Ps Agree
user: Impart Believes Agree
system: Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject Justification Price Impart Believes Tactic
user: Impart Believes Tactic Matrix
system: Impart D_Committed Matrix Impart Believes Tactic
user: Impart Believes Tactic Identify_Parameters
system: Impart D_Committed Identify_Parameters Impart Believes Tactic
user: Impart Believes Tactic Get_Alternative_Values
system: Impart D_Committed Get_Alternative_Values Impart Believes Tactic Agree_Lifestyle_Objectives Impart Believes Tactic Agree_Lifestyle_Objectives Agree
user: Impart Believes Agree
system: Impart D_Committed Agree_Lifestyle_Objectives Impart Believes Objectives
user: Impart Believes Objectives Luggage Impart Believes Objectives Noise Impart Believes Objectives Reliability Impart Believes Objectives Bodywork
system: Impart Believes Objectives Luggage Agree Impart Believes Objectives Noise Agree Impart Believes Objectives Reliability Agree Impart Believes Objectives Bodywork Agree Elicit Believes Objectives
user: Impart Believes Objectives Min_External_Size
system: Impart D_Committed Continue_Tactic Impart Believes Objectives Min_External_Size Agree Impart Believes Objectives Min_External_Size Justification Narrow_Roads
user: Impart Believes Objectives Sunroof
system: Impart Believes Objectives Sunroof Agree Impart Believes Objectives Sunroof Not Justification Tend_To_Leak Impart Believes Objectives Sunroof Justification
user: Impart Believes Objectives Sunroof Justification Fun_In_Nice_Weather Agree Impart Believes Objectives Sunroof Disagree Elicit Believes Objectives Sunroof Confirmation
user: Impart Believes Confirmation Elicit Believes Objectives
system: Impart Believes Objectives Running_Cost Impart Believes Objectives Running_Cost Justification Not_Rich Elicit Believes Objectives
user: Impart Believes Tactic Finish_Tactic
system: Impart Believes Tactic Finish_Tactic Disagree Elicit Believes Tactic Continue_Tactic Agree Elicit Believes Tactic Continue_Tactic Agree
R: Research Student:

user: Impart Believes Tactic
system: Elicit Believes Tactic
user: Impart Believes Tactic Eliminate_Fail_Ps
system: Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject
Justification Price Impart Believes Tactic Agree_Ps_Objectives Elicit Believes Tactic
Agree_Ps_Objectives Agree
user: Impart Believes Agree
system: Impart D_Committed Agree_Ps_Objectives Impart Believes Objectives Safety
Justification In_Ps Impart Believes Objectives Comfort Justification In_Ps Elicit
Believes Tactic
user: Impart Believes Tactic Matrix
system: Impart D_Committed Matrix Elicit Believes Tactic
user: Elicit Believes Tactic
system: Impart Believes Tactic Agree_Lifestyle_Objectives Impart Believes Tactic
Agree_Lifestyle_Objectives Justification Take Account Of Purchaser Elicit
Believes Tactic Agree_Lifestyle_Objectives Agree
user: Impart Believes Agree
system: Impart D_Committed Agree_Lifestyle_Objectives Elicit Believes Objectives
user: Impart Believes Objectives Fuel_Economy Elicit Believes Objectives
system: Impart Believes Objectives Fuel_Economy Agree Impart Believes Objectives
Fuel_Economy Justification Minimise_Environmental_Damage Impart Believes

Confirmation Elicit Believes Tactic Finish_Tactic Agree
user: Impart Believes Agree
system: Impart D_Committed Finish_Tactic Impart Believes Tactic Finish_Tactic Disagree
Impart Believes Tactic Finish_Tactic Not Justification No_Active_Tactic Elicit
Believes Tactic Finish_Tactic Justification
user: Impart Believes Justification Enough_Factors
system: Impart Believes Tactic Finish_Tactic Justification Enough_Factors Not-Known
Impart Believes Tactic Finish_Tactic Justification Enough_Factors Impossible
Impart Believes Tactic Finish_Tactic Disagree Elicit Believes Tactic Finish_Tactic
Confirmation
user: Impart Believes Confirmation
system: Elicit Believes Tactic
user: Impart Believes Tactic Identify_Parameters
system: Impart D_Committed Identify_Parameters Impart Believes Tactic
Get_Alternative_Values Elicit Believes Tactic Get_Alternative_Values Agree
user: Impart Believes Agree
system: Impart D_Committed Get_Alternative_Values Elicit Believes Tactic
user: Impart Believes Tactic Assign_Weights
system: Impart D_Committed Assign_Weights Elicit Believes Tactic
user: Impart Believes Tactic Do_Wom
system: Impart D_Committed Do_Wom Disagree Impart Believes Tactic Scale_Parameters
Elicit Believes Tactic Scale_Parameters Agree
user: Impart Believes Agree
system: Impart D_Committed Scale_Parameters Elicit Believes Tactic
user: Impart Believes Tactic Do_Wom
system: Impart D_Committed Do_Wom Disagree Impart Believes Tactic Assign_Values
Elicit Believes Tactic Assign_Values Agree Elicit Believes Tactic Assign_Values
Agree
user: Elicit Believes Justification
system: Impart Believes Tactic Assign_Values Justification
For_Comparing_Different_Factors Elicit Believes Tactic Assign_Values Agree
user: Impart Believes Agree
system: Impart D_Committed Assign_Values Elicit Believes Tactic
user: Impart Believes Tactic Do_Wom
system: Impart D_Committed Do_Wom Impart Believes Alternatives A Best Impart Believes
Tactic End Elicit Believes Tactic End Agree
user: Impart Believes Agree
Appendix C: Evaluation of WOMBAT

Design, Decisions and Dialogue

Objectives Running_Cost Elicit Believes Objectives

user: Impart Believes Objectives Speed Impart Believes Objectives Acceleration Impart Believes Objectives Sunroof

system: Impart Believes Objectives Speed Agree Impart Believes Objectives Speed

Justification Fun Impart Believes Objectives Acceleration Agree Impart Believes Objectives Sunroof Disagree Impart Believes Objectives Sunroof Not Justification

Tend_To_Leak Elicit Believes Objectives Sunroof Justification

user: Impart Believes Justification Fun_In_Nice_Weather

system: Impart Believes Objectives Sunroof Justification Fun_In_Nice_Weather Agree

Impart Believes Objectives Sunroof Disagree Elicit Believes Objectives Sunroof Confirmation

user: Impart Believes Confirmation

system: Elicit Believes Objectives

user: Impart Believes Tactic Finish_Tactic

system: Impart Believes Tactic Finish_Tactic Disagree Impart Believes Tactic

Continue_Tactic Elicit Believes Tactic Continue_Tactic Agree

user: Elicit Believes Justification

system: Impart Believes Tactic Continue_Tactic Justification Better_To_Have_More_Factors

Elicit Believes Tactic Continue_Tactic Agree

user: Impart Believes Agree

system: Impart D_Committed Continue_Tactic Elicit Believes Objectives

user: Impart Believes Objectives Pollution

system: Impart Believes Objectives Pollution Agree Impart Believes Objectives Pollution

Justification Minimise_Environmental_Damage Elicit Believes Objectives

user: Impart Believes Objectives Range

system: Impart Believes Objectives Range Not-Known Elicit Believes Objectives Range Confirmation

user: Impart Believes Confirmation

system: Impart Believes Tactic Finish_Tactic Elicit Believes Tactic Finish_Tactic Agree

user: Impart Believes Agree

system: Impart D_Committed Finish_Tactic Elicit Believes Tactic

user: Impart Believes Tactic Identify_Parameters

system: Impart D_Committed Identify_Parameters Elicit Believes Tactic

Get_Parameter_Values Elicit Believes Tactic Get_Parameter_Values Agree

user: Impart Believes Agree

system: Impart D_Committed Get_Parameter_Values Elicit Believes Tactic

user: Impart Believes Tactic Scale_Parameters

system: Impart D_Committed Scale_Parameters Elicit Believes Tactic

user: Impart Believes Tactic Assign_Values

system: Impart D_Committed Assign_Values Elicit Believes Tactic

user: Impart Believes Tactic Assign_Weights

system: Impart D_Committed Assign_Weights Elicit Believes Tactic Do_Wom Elicit Believes Tactic Do_Wom Agree

user: Impart Believes Agree

system: Impart D_Committed Do_Wom Impart Believes Alternatives A Best Elicit Believes Tactic

user: Impart Believes Tactic End

system: Impart D_Committed End

J: RF in Design:

system: Elicit Believes Tactic

user: Impart Believes Tactic Eliminate_Fail_Ps

system: Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject

Justification Price Impart Believes Tactic Agree_Ps_Objectives Elicit Believes Tactic Agree_Ps_Objectives Agree

user: Impart Believes Agree

system: Impart D_Committed Agree_Ps_Objectives Elicit Believes Objectives Safety

Justification In_Ps Impart Believes Objectives Comfort Justification In_Ps Elicit Believes Tactic

328
user: Impart Believes Tactic Matrix
system: Impart D_Committed Matrix Elicit Believes Tactic
user: Impart Believes Tactic Identify_Parameters
system: Impart D_Committed Identify_Parameters Elicit Believes Tactic
user: Impart Believes Tactic Get_Parameter_Values
system: Impart D_Committed Get_Parameter_Values Impart Believes Tactic Agree_Lifestyle_Objectives Elicit Believes Tactic Agree_Lifestyle_Objectives Agree
user: Impart Believes Agree
system: Impart D_Committed Agree_Lifestyle_Objectives Elicit Believes Objectives
user: Impart Believes Objectives 4_Doors Impart Believes Objectives Flexibility Impart Believes Objectives Fuel_Economy Impart Believes Objectives Running_Cost
system: Impart Believes Objectives 4_Doors Agree Impart Believes Objectives 4_Doors Justification People_In_Back Impart Believes Objectives Fuel_Economy Agree Impart Believes Objectives Running_Cost Agree Impart Believes Objectives Flexibility Disagree Impart Believes Objectives Flexibility Not Justification Unimportant_Most_Of_The_Time Elicit Believes Objectives Flexibility Justification
user: Impart Believes Justification Varied_Lifestyle
system: Impart Believes Objectives Flexibility Justification Varied_Lifestyle Not-Known Impart Believes Objectives Flexibility Disagree Elicit Believes Objectives Flexibility Confirmation
user: Impart Believes Confirmation
system: Elicit Believes Objectives
user: Impart Believes Tactic Identify_Parameters
system: Impart D_Committed Identify_Parameters Impart Believes Objectives Acceleration Elicit Believes Objectives
user: Impart Believes Objectives Acceleration Disagree Impart Believes Tactic Get_Parameter_Values
system: Impart D_Committed Get_Parameter_Values Impart Believes Objectives Acceleration Not Disagree Elicit Believes Objectives Acceleration Not Confirmation
user: Impart Believes Confirmation
system: Elicit Believes Objectives
user: Impart Believes Tactic Do_Wom
system: Impart Believes Tactic Do_Wom Disagree Impart Believes Tactic Scale_Parameters Elicit Believes Tactic Scale_Parameters Agree
user: Impart Believes Agree
system: Impart D_Committed Scale_Parameters Elicit Believes Objectives
user: Impart Believes Tactic Assign_Weights
system: Impart D_Committed Tactic Assign_Weights Disagree Impart Believes Tactic Assign_Weights Not Justification Should_Agree_Lifestyle_Objectives Elicit Believes Tactic Assign_Weights Not Justification
user: Impart Believes Past Elicit Believes Objectives
user: Impart Believes Objectives No_Ideas
system: Impart Believes Objectives Reliability Elicit Believes Objectives
user: Impart Believes Agree
system: Impart D_Committed Tactic Assign_Weights Elicit Believes Objectives
user: Impart Believes Objectives Reliability Agree Impart Believes Objectives Reliability Justification Inconvenience Elicit Believes Objectives
user: Elicit Believes Objectives
system: Impart D_Committed Tactic Assign_Weights Elicit Believes Objectives Pollution Impart Believes Objectives Pollution Justification Minimise_Environmental_Damage Impart Believes Tactic Finish_Tactic Agree
user: Impart Believes Objectives Pollution Disagree
system: Impart D_Committed Tactic Assign_Weights Elicit Believes Objectives Pollution Not Disagree Elicit Believes Objectives Pollution Not Confirmation
user: Impart Believes Confirmation
system: Elicit Believes Tactic
user: Impart Believes Tactic Identify_Parameters
system: Impart D_Committed Identify_Parameters Impart Believes Tactic
Appendix C: Evaluation of WOMBAT

Design, Decisions and Dialogue

Get_Parameter.Values Elicit Believes Tactic Get_Parameter.Values Agree
user: Impart Believes Agree
system: Impart D_Committed Get_Parameter.Values Elicit Believes Tactic
user: Impart Believes Tactic Scale_Parameters
system: Impart D_Committed Scale_Parameters Elicit Believes Tactic
user: Impart Believes Tactic Assign.Values
system: Impart D_Committed Assign.Values Elicit Believes Tactic
user: Impart Believes Tactic Do_Wom
system: Impart Believes Tactic Do_Wom Disagree Impart Believes Tactic Do_Wom Not
Justification Need_To_Assign_Weights First Elicit Believes Tactic Do_Wom
Justification
user: Impart Believes Past Impart Believes Tactic Assign_Weights
system: Impart Believes Tactic Do_Wom Not Agree Impart D_Committed Assign_Weights
Elicit Believes Tactic
user: Impart Believes Tactic Do_Wom
system: Impart D_Committed Do_Wom Impart Believes Alternatives F Best Impart Believes
Tactic End Elicit Believes Tactic End Agree
user: Impart Believes Agree
system: Impart D_Committed End

M: Lecturer in Materials:

system: Elicit Believes Tactic
user: Impart Believes Tactic Eliminate_Fail_Ps
system: Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject
Justification Price Impart Believes Tactic Agree_Ps_Objectives Elicit Believes Tactic
Agree_Ps_Objectives Agree
user: Impart Believes Agree
system: Impart D_Committed Agree_Ps_Objectives Impart Believes Objectives Safety
Justification In_Ps Impart Believes Objectives Comfort Justification In_Ps Elicit
Believes Tactic
user: Impart Believes Tactic Matrix
system: Impart D_Committed Matrix Elicit Believes Tactic
user: Impart Believes Tactic Identify_Parameters
system: Impart D_Committed Identify_Parameters Elicit Believes Tactic
user: Impart Believes Tactic Get_Parameter.Values
system: Impart D_Committed Get_Parameter.Values Impart Believes Tactic
Agree_Lifestyle_Objectives Elicit Believes Tactic Agree_Lifestyle_Objectives Agree
user: Impart Believes Disagree Impart Believes Tactic Assign_Weights
system: Impart Believes Tactic Agree_Lifestyle_Objectives Not Disagree Impart Believes
Tactic Assign_Weights Impart Believes Tactic Assign_Weights Not
Justification Should_Agree_Lifestyle_Objectives Elicit Believes Tactic
Assign_Weights Justification
user: Impart Believes Past
system: Impart Believes Tactic Assign_Weights Not Agree Impart Believes Tactic
Agree_Lifestyle_Objectives Not Disagree Elicit Believes Tactic
Agree_Lifestyle_Objectives Not Confirmation
user: Impart Believes Past
system: Impart D_Committed Agree_Lifestyle_Objectives Elicit Believes Objectives
user: Impart Believes Objectives Fuel_Economy Impart Believes Objectives Luggage
system: Impart Believes Objectives Fuel_Economy Agree Impart Believes Objectives
Fuel_Economy Justification Minimise_Environmental_Damage Impart Believes
Objectives Luggage Agree Elicit Believes Objectives Luggage Justification
user: Impart Believes Justification Lots_Of_Children
system: Impart Believes Objectives Luggage Justification Lots_Of_Children Not-Known
Elicit Believes Objectives Luggage Justification Lots_Of_Children Confirmation
user: Impart Believes Confirmation
system: Elicit Believes Objectives
user: Impart Believes Tactic Finish_Tactic
system: Impart Believes Tactic Finish_Tactic Disagree Impart Believes Tactic
Design, Decisions and Dialogue
Appendix C: Evaluation of WOMBAT

Continue_Tactic Elicit Believes Tactic Continue_Tactic Agree
user: Impart Believes Disagree Impart Believes Tactic Finish_Tactic
system: Impart Believes Tactic Continue_Tactic Not Disagree Impart Believes Tactic Finish_Tactic Not Justification Better_To_Have_More_Factors Elicit Believes Tactic Finish_Tactic Justification
user: Impart Believes Justification Enough_Factors
user: Impart Believes Confirmation
system: Impart Believes Tactic Continue_Tactic Not Disagree Elicit Believes Tactic Continue_Tactic Not Confirmation
user: Impart Believes Confirmation
system: Elicit Believes Tactic
user: Impart Believes Tactic Identify_Parameters
system: Impart D_Committed Identify_Parameters Impart Believes Tactic
    Get_Parameter_Values Elicit Believes Tactic Get_Parameter_Values Agree
user: Impart Believes Tactic Get_Parameter_Values
system: Elicit Believes Tactic
user: Impart Believes Tactic Assign_Weights
system: Impart D_Committed Assign_Weights Elicit Believes Tactic
user: Impart Believes Tactic Scale_Parameters
system: Impart D_Committed Scale_Parameters Elicit Believes Tactic
user: Impart Believes Tactic Assign_Agrees
system: Impart D_Committed Assign_Agrees Impart Believes Tactic Do_Worn Elicit Believes Tactic Do_Worn Agree
user: <Null input>
system: Impart Believes Alternatives F Best Elicit Believes Tactic
user: Impart Believes Tactic Agree_Lifestyle_Objectives
system: Impart D_Committed Agree_Lifestyle_Objectives Elicit Believes Objectives
user: Impart Believes Objectives Speed Agree Impart Believes Objectives Speed Justification Fun Elicit Believes Objectives
user: Impart Believes Tactic Finish_Tactic
system: Impart Believes Tactic Finish_Tactic Disagree Elicit Believes Tactic Finish_Tactic Confirmation
user: Impart Believes Confirmation
system: Impart Believes Tactic Identify_Parameters Elicit Believes Tactic Identify_Parameters Agree
user: Impart Believes Agree
system: Impart D_Committed Identify_Parameters Elicit Believes Tactic
user: Impart Believes Tactic Get_Parameter_Values
system: Impart D_Committed Get_Parameter_Values Elicit Believes Tactic
user: Impart Believes Tactic Assign_Weights
system: Impart D_Committed Assign_Weights Elicit Believes Tactic
user: Impart Believes Tactic Scale_Parameters
system: Impart D_Committed Scale_Parameters Elicit Believes Tactic Scale_Parameters Not-Known Impart Believes Tactic Scale_Parameters Impossible Impart Believes Tactic Scale_Parameters Agree
user: Impart Believes Disagree Impart Believes Tactic Scale_Parameters
system: Impart D_Committed Scale_Parameters Impart Believes Tactic Scale_Parameters Not Disagree Elicit Believes Tactic Scale_Parameters Not Confirmation
user: Impart Believes Confirmation Impart Believes Tactic Assign_Agrees
system: Impart D_Committed Assign_Agrees Elicit Believes Tactic
user: Impart Believes Tactic Do_Worn
system: Impart D_Committed Do_Worn Impart Believes Alternatives F Best Elicit Believes Tactic
user: Impart Believes Tactic End
system: Impart D_Committed End
D: Lecturer in Aeronautical Engineering:

system: Elicit Believes Tactic
user: Impart Believes Tactic Eliminate_Fail_Ps
system: Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject
Justice Price Impart Believes Tactic Agree_Ps_Objectives Elicit Believes Tactic Agree_Ps_Objectives Agree
user: Impart Believes Agree
system: Impart D_Committed Agree_Ps_Objectives Impart Believes Objectives Safety
Justice In_Ps Impart Believes Objectives Comfort Justice In_Ps Elicit Believes Tactic
user: Impart Believes Tactic Agree_Lifestyle_Objectives
system: Impart D_Committed Agree_Lifestyle_Objectives Elicit Believes Objectives
user: Impart Believes Objectives Pollution
system: Impart Believes Objectives Pollution Agree Impart Believes Objectives Pollution
Justice Minimise_Environmental_Damage Elicit Believes Objectives
user: Impart Believes Objectives Running_Cost
system: Impart Believes Objectives Running_Cost Agree Elicit Believes Objectives
Running_Cost Justification
user: Impart Believes Justification Not_Rich
system: Impart Believes Objectives Running_Cost Justification Not_Rich Agree Impart
Believes Objectives Running_Cost Justification Not_Rich Justification Max_Price
Elicit Believes Objectives
user: Impart Believes Tactic Finish_Tactic
system: Impart Believes Tactic Finish_Tactic Disagree Impart Believes Tactic Matrix Elicit
Believes Tactic Matrix Agree
user: Impart Believes Agree
system: Impart D_Committed Matrix Elicit Believes Objectives
user: Impart Believes Objectives 4_Doors
system: Impart Believes Objectives 4_Doors Agree Elicit Believes Objectives 4_Doors
Justice
user: Impart Believes Justification People_In_Back
system: Impart Believes Objectives 4_Doors Justification People_In_Back Agree Elicit
Believes Objectives
user: Elicit Believes Objectives
system: Impart Believes Objectives Acceleration Impart Believes Objectives Acceleration
Justice Get_Out_Of_Trouble Elicit Believes Objectives
user: Impart Believes Disagree
system: Impart Believes Objectives Acceleration Justification Get_Out_Of_Trouble Not
Disagree Elicit Believes Objectives Acceleration Justification Get_Out_Of_Trouble Not
Confirmation
user: Impart Believes Confirmation
system: Impart Believes Objectives Price Elicit Believes Objectives
user: Impart Believes Tactic Finish_Tactic
system: Impart Believes Tactic Finish_Tactic Disagree Impart Believes Tactic
Continue_Tactic Elicit Believes Tactic Continue_Tactic Agree
user: Impart Believes Disagree Impart Believes Tactic Finish_Tactic
system: Impart Believes Tactic Continue_Tactic Not Disagree Impart Believes Tactic
Finish_Tactic Disagree Impart Believes Tactic Finish_Tactic Not Justification
Better_To_Have_More_Factors Elicit Believes Tactic Finish_Tactic Justification
user: Impart Believes Justification No_Ideas
system: Impart Believes Tactic Continue_Tactic Not Disagree Impart Believes Tactic
Finish_Tactic Disagree Elicit Believes Tactic Finish_Tactic Confirmation
user: Impart Believes Confirmation
system: Impart Believes Tactic Continue_Tactic Not Disagree Elicit Believes Tactic
Continue_Tactic Not Confirmation
user: Impart Believes Confirmation
system: Elicit Believes Tactic
user: Impart Believes Tactic Identify_Parameters
system: Impart D_Committed Identify_Parameters Impart Believes Tactic
   Get_Parameter_Values Elicit Believes Tactic Get_Parameter_Values Agree
user: Impart Believes Agree
system: Impart D_Committed Get_Parameter_Values Elicit Believes Tactic
user: Impart Believes Tactic Assign_Weights
system: Impart D_Committed Assign_Weights Elicit Believes Tactic
user: Impart Believes Tactic Scale_Parameters
system: Impart D_Committed Scale_Parameters Elicit Believes Tactic
user: Impart Believes Tactic Assign_Values
system: Impart D_Committed Assign_Values Impart Believes Tactic Do_Wom Elicit
Believes Tactic Do_Wom Agree
user: Impart Believes Agree
system: Impart D_Committed Do_Wom Impart Believes Alternatives F Best Elicit Believes
Tactic
user: Impart Believes Tactic End
system: Impart D_Committed End

C: Programmer:

In this session, the program crashed due to a bug which was subsequently repaired, so both interactions are included.

system: Elicit Believes Tactic
user: Impart Believes Not-Known
system: Impart Believes Not-Known Not-Known Elicit Believes Confirmation
user: Impart Believes Past Impart Believes Tactic No_Ideas
system: Impart Believes Not Disagree Impart Believes Not Not-Known Not-Known Elicit
Believes Not Confirmation
user: Impart Believes Confirmation
system: Elicit Believes Tactic
user: Impart Believes Tactic No_Ideas
system: Impart Believes Tactic Agree_Ps_Objectives Elicit Believes Tactic
   Agree_Ps_Objectives Agree
user: Impart Believes Agree
system: Impart D_Committed Agree_Ps_Objectives Impart Believes Objectives Safety
   Justification In_Ps Impart Believes Objectives Comfort Justification In_Ps Elicit
Believes Tactic
user: Impart Believes Agree
system: Elicit Believes Tactic
user: Impart Believes Tactic No_Ideas
system: Impart Believes Tactic Eliminate_Fail_Ps Elicit Believes Tactic Eliminate_Fail_Ps
   Agree
user: Impart Believes Agree
system: Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject
   Justification Price Elicit Believes Tactic
user: Impart Believes Agree
system: Elicit Believes Tactic
user: Impart Believes Tactic No_Ideas
system: Impart Believes Tactic Matrix Elicit Believes Tactic Matrix Agree
user: Impart Believes Agree
system: Impart D_Committed Matrix Elicit Believes Tactic
user: Impart Believes Agree
system: Elicit Believes Tactic
user: Impart Believes Tactic No_Ideas
system: Impart Believes Tactic Agree_Lifestyle_Objectives Elicit Believes Tactic
   Agree_Lifestyle_Objectives Agree
user: Impart Believes Agree
system: Impart D_Committed Agree_Lifestyle_Objectives Elicit Believes Objectives
user: Impart Believes Objectives Safety
system: Impart Believes Objectives Safety Agree Impart Believes Objectives Safety
Appendix C: Evaluation of WOMBAT
Design, Decisions and Dialogue

Justification Saves Lives Elicit Believes Objectives

> Error: Unexpected end of file encountered.

? (agent 't((kw family only_car) (need (price max 8000)) (wants safety comfort))))

system: Elicit Believes Tactic

user: Impart Believes Tactic Eliminate_Fail_Ps Impart Believes Tactic Agree_Ps_Objectives
Impart Believes Tactic Matrix

system: Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject
Justification Price Impart D_Committed Agree_Ps_Objectives Impart Believes Objectives Safety
Justification In_Ps Impart Believes Objectives Comfort
Justification In_Ps Impart D_Committed Matrix

user: Impart Believes Tactic Agree_Lifestyle_Objectives

system: Impart D_Committed Agree_Lifestyle_Objectives Impart Believes Objectives
Running_Cost Elicit Believes Objectives

user: Impart Believes Agree

system: Impart Believes Objectives Running_Cost Agree Impart Believes Objectives
Running_Cost Justification Not_Rich Elicit Believes Objectives

user: Impart Believes Agree

system: Impart Believes Objectives Running_Cost Justification Not_Rich Agree Elicit
Believes Objectives Running_Cost Justification Not_Rich Justification

user: Impart Believes Justification Not_Rich

system: Impart Believes Objectives Running_Cost Justification Not_Rich Justification
Not_Rich Not-Known Elicit Believes Objectives Running_Cost Justification
Not_Rich Justification Not_Rich Confirmation

user: Impart Believes Confirmation

system: Elicit Believes Objectives

user: Impart Believes Objectives Max_Internal_Size

system: Impart Believes Objectives Max_Internal_Size Agree Impart Believes Objectives
Max_Internal_Size Justification Comfort Elicit Believes Objectives

user: Impart Believes Objectives Reliability

system: Impart Believes Objectives Reliability Agree Impart Believes Objectives Reliability
Justification Inconvenience Elicit Believes Objectives

user: Impart Believes Tactic Finish_Tactic

system: Impart Believes Tactic Finish_Tactic Disagree Impart Believes Tactic
Continue_Tactic Elicit Believes Tactic Continue_Tactic Agree

user: Impart Believes Disagree

system: Impart Believes Tactic Continue_Tactic Not Disagree Elicit Believes Tactic
Continue_Tactic Not Confirmation

user: Impart Believes Confirmation Impart Believes Tactic Finish_Tactic

system: Impart Believes Tactic Finish_Tactic Disagree Impart Believes Tactic Finish_Tactic
Not Justification Better_To_Have_More_Factors Elicit Believes.Tactic Finish_Tactic
Justification

user: Impart Believes Justification No_Ideas

system: Impart Believes Tactic Finish_Tactic Disagree Elicit Believes Tactic Finish_Tactic
Confirmation

user: Impart Believes Confirmation

system: Elicit Believes Tactic

user: Impart Believes Tactic No_Ideas

system: Impart Believes Tactic Identify_Parameters Elicit Believes Tactic
Identify_Parameters Agree

user: Impart Believes Agree

system: Impart D_Committed Identify_Parameters Elicit Believes Tactic

user: Impart Believes Tactic No_Ideas

system: Impart Believes Tactic Get_Parameter_Values Elicit Believes Tactic
Get_Parameter_Values Agree

user: Impart Believes Agree

system: Impart D_Committed Get_Parameter_Values Elicit Believes Tactic

user: Impart Believes Tactic No_Ideas

system: Impart Believes Tactic Scale_Parameters Elicit Believes Tactic Scale_Parameters
Agree

user: Impart Believes Agree
Design, Decisions and Dialogue

Appendix C: Evaluation of WOMBAT

system: Impart D_Committed Scale_Parameters Elicit Believes Tactic
user: Impart Believes Agree
system: Elicit Believes Tactic
user: Impart Believes Tactic No_Ideas
system: Impart Believes Tactic Assign_Weights Elicit Believes Tactic Assign_Weights Agree
user: Impart Believes Agree
system: Impart D_Committed Assign_Weights Elicit Believes Tactic
user: Impart Believes Agree
system: Elicit Believes Tactic
user: Impart Believes Tactic No_Ideas
system: Impart Believes Tactic Do_Wom Impart Believes Tactic Do_Wom Agree
user: Impart Believes Agree
system: Impart D_Committed Do_Wom Impart Believes Alternatives A Best Impart Believes Tactic
user: Impart Believes Tactic End
system: Impart D_Committed End

B: Professor of Mechanical Engineering:

system: Elicit Believes Tactic
user: Impart Believes Tactic Eliminate_Fail_Ps
system: Impart D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject Justification Price Impart Believes Tactic Agree_Ps_Objectives Elicit Believes Tactic Agree_Ps_Objectives Agree
user: Impart Believes Justification
system: Impart Believes Tactic Agree_Ps_Objectives Justification Take_Problem_Spec_Into_Account Elicit Believes Tactic Agree_Ps_Objectives Agree
user: Impart Believes Agree
system: Impart D_Committed Agree_Ps_Objectives Impart Believes Objectives Safety Justification In_Ps Impart Believes Objectives Comfort Justification In_Ps Elicit Believes Tactic
user: Impart Believes Tactic
system: Impart Believes Tactic Matrix Impart Believes Tactic Matrix Justification Organise_Data Elicit Believes Tactic Matrix Agree
user: Impart Believes Agree
system: Impart D_Committed Matrix Impart Believes Tactic
user: Impart Believes Tactic Agree_Lifestyle_Objectives
system: Impart D_Committed Agree_Lifestyle_Objectives Impart Believes Objectives
user: Impart Believes Tactic Runnin_Cost Elicit Believes Objectives
system: Impart Believes Objectives 4_Doors Impart Believes Objectives Central_Locking
user: Impart Believes Objectives Holidays Agree Impart Believes Objectives Luggage
user: Impart Believes Justification Holidays Agree Impart Believes Objectives Luggage Justification Holidays Justification Only_Car Elicit Believes Objectives
user: Impart Believes Objectives Availability_Of_Spares
system: Impart Believes Objectives Availability_Of_Spares Not-Known Elicit Believes Objectives Availability_Of_Spares Confirmation
user: Impart Believes Confirmation Impart Believes Objectives Flexibility
system: Impart Believes Objectives Flexibility Disagree Impart Believes Objectives Flexibility Not Justification Unimportant_Most_Of_The_Time Elicit Believes Objectives Flexibility Justification
user: Impart Believes Justification Diy
system: Impart Believes Objectives Flexibility Justification Diy Not-Known Impart Believes Objectives Flexibility Disagree Elicit Believes Objectives Flexibility Confirmation
user: Impart Believes Confirmation
system: Impart Believes Tactic Finish_Tactic Elicit Believes Tactic Finish_Tactic Agree
user: Impart Believes Agree
system: Impart_D_Committed Finish_Tactic Elicit Believes Tactic
user: Impart Believes Tactic Identity_Parameters
system: Impart_D_Committed Identify_Parameters Impart Believes Tactic Get_Parameter_Values Elicit Believes Tactic Get_Parameter_Values Agree
user: Impart Believes Agree
system: Impart_D_Committed Get_Parameter_Values Elicit Believes Tactic
user: Impart Believes Tactic Assign_Weights
system: Impart_D_Committed Assign_Weights Elicit Believes Tactic
user: Impart Believes Tactic Change_Weights
Flexibility: 2, Availability_Of_Spares: 0, Luggage: 3, Max_Internal_Size: 4, 4_Doors: 1, Running_Cost: 3, Comfort: 6, Safety: 5
system: Impart_D_Committed Change_Weights Elicit Believes Tactic
user: Impart Believes Tactic Scale_Parameters
system: Impart_D_Committed Scale_Parameters Impart Believes Tactic Assign_Values Elicit Believes Tactic Assign_Values Agree
user: Impart Believes Agree
system: Impart_D_Committed Assign_Values Elicit Believes Tactic
user: Impart Believes Tactic Do_Wom
system: Impart Believes Tactic Do_Wom Impart Believes Tactic Do_Wom Justification Allows_Different_Factors_To_Be_Compared_And_Reflects_Their_Relative_Importance Elicit Believes Tactic Do_Wom Agree
user: Impart Believes Agree
system: Impart_D_Committed Do_Wom Elicit Believes Alternatives F Best Impart Believes Tactic
user: Impart Believes Tactic End
system: Impart_D_Committed End

K: Lecturer in Engineering Design:

The system crashed while being used with K; as there has been evidence in the past that the version of Lisp being used is not 100% reliable on the Mac SE/30, and as the error could not be reproduced, it seems reasonable to conclude that this crash was the result of a Lisp interpreter problem rather than being caused by a bug in WOMBAT. Therefore, what follows is a reconstruction of the interaction, and not the actual interaction between K and WOMBAT:

system: Elicit Believes Tactic
user: Impart Believes Tactic Eliminate_Fail_Ps
system: Impart_D_Committed Eliminate_Fail_Ps Impart Believes Alternatives D Reject Justification Price Impart Believes Tactic Agree Ps_Objectives Agree
user: Elicit Believes Justification
system: Impart Believes Tactic Agree Ps_Objectives Justification Take_Problem_Spec_Into_Account Elicit Believes Tactic Agree Ps_Objectives Agree
user: Impart Believes Agree
system: Impart_D_Committed Agree Ps_Objectives Elicit Believes Objectives Safety Justification In_PS Impart Believes Objectives Comfort Justification In_PS Elicit

336
**Believes Tactic**

**user:** Elicit Believes Tactic

**system:** Impart Believes Tactic Matrix Impart Believes Tactic Matrix Justification

Organise Data Elicit Believes Tactic Matrix Agree

**user:** Impart Believes Agree

**system:** Impart D_Committed Matrix Elicit Believes Tactic

user: Elicit Believes Tactic

system: Impart Believes Tactic Agree_Lifestyle_Objectives Elicit Believes Tactic Agree_Lifestyle_Objectives Agree

user: Elicit Believes Justification

system: Impart Believes Tactic Agree_Lifestyle_Objectives Justification

Take_Account_Of_Purchaser Elicit Believes Tactic Agree_Lifestyle_Objectives Agree

user: Impart Believes Agree

system: Impart D_Committed Agree_Lifestyle_Objectives Elicit Believes Objectives

user: Impart Believes Objectives Pollution Impart Believes Objectives Fuel_Economy Impart Believes Objectives Max_Internal_Size Impart Believes Objectives Luggage Impart Believes Objectives 4_Doors Impart Believes Objectives Flexibility Impart Believes Objectives Reliability Impart Believes Objectives Colour

system: Impart Believes Objectives 4_Doors Agree Impart Believes Objectives 4_Doors Justification People_In_Back Impart Believes Objectives Flexibility Disagree Impart Believes Objectives Reliability Agree Impart Believes Objectives Reliability Justification Inconvenience Impart Believes Objectives Colour Disagree Impart Believes Objectives Colour Not Justification Various_Colours_Available Elicit Believes Objectives Colour Justification

user: Impart Believes Past

system: Impart Believes Objectives Colour Not Agree Impart Believes Objectives Flexibility Disagree Impart Believes Objectives Flexibility Not Justification Unimportant_Most_Of_The_Time Elicit Believes Objectives Flexibility Justification

user: Impart Believes Justification Windsurfing

system: Impart Believes Objectives Flexibility Justification Windsurfing Not-Known Impart Believes Objectives Flexibility Disagree Elicit Believes Objectives Flexibility Confirmation

user: Impart Believes Confirmation

system: Impart Believes Tactic Finish_Tactic Elicit Believes Tactic Finish_Tactic Agree

user: Elicit Believes Justification

system: Impart Believes Tactic Finish_Tactic Justification No_Ideas Elicit Believes Tactic Finish_Tactic Agree

user: Impart Believes Agree

system: Impart D_Committed Finish_Tactic Elicit Believes Tactic

user: Impart Believes Tactic Identify_Parameters

system: Impart D_Committed Identify_Parameters Elicit Believes Tactic Get_Parameter_Values Elicit Believes Tactic Get_Parameter_Values Agree

user: Impart Believes Agree

system: Impart D_Committed Get_Parameter_Values Elicit Believes Tactic

user: Impart Believes Tactic Eliminate_Worst_On_1_Objective


user: Impart Believes Past

system: Impart Believes Tactic Eliminate_Worst_On_1_Objective Not Agree Elicit Believes Tactic

user: Impart Believes Tactic Assign_Weights

system: Impart D_Committed Assign_Weights Elicit Believes Tactic Scale_Parameters Elicit Believes Tactic Scale_Parameters Agree

user: Impart Believes Agree

system: Impart D_Committed Scale_Parameters Elicit Believes Tactic

user: Impart Believes Tactic Change_Weights

Flexibility: 7, Pollution: 7, Fuel_Economy: 6, Max_Internal_Size: 4, Luggage: 5, 4_Doors:
Appendix C: Evaluation of WOMBAT


system: Impart D_Committed Change_Weights Elicit Believes Tactic
user: Impart Believes Tactic Assign_Values
system: Impart D_Committed Assign_Values Elicit Believes Tactic
user: Impart Believes Tactic Do_Wom
system: Impart D_Committed Do_Wom Import Believes Alternatives F Best Impart Believes Tactic End Elicit Believes Tactic End Agree
user: Impart Believes Agree
system: Impart D_Committed End

G: Professor of Engineering Design:

system: Elicit Believes Tactic
user: Impart Believes Tactic Eliminate_Fail_Ps
system: Impart D_Committed Eliminate_Fail_Ps Import Believes Alternatives D Reject
Justification Price Impart Believes Tactic Agree_Ps_Objects Agree Elicit Believes Tactic Agree_Ps_Objects Agree
user: Elicit Believes Justification
system: Impart Believes Tactic Agree_Ps_Objects Justification
Take_Problem_Spec_Into_Account Elicit Believes Tactic Agree_Ps_Objects Agree
user: Impart Believes Agree
system: Impart D_Committed Agree_Ps_Objects Import Believes Objectives Safety
Justification In_Ps Import Believes Objectives Comfort Justification In_Ps Elicit Believes Tactic
user: Impart Believes Tactic
system: Impart Believes Tactic Matrix Import Believes Tactic Matrix Justification
Organise_Data Elicit Believes Tactic Matrix Agree
user: Impart Believes Agree
system: Impart D_Committed Matrix Import Believes Tactic
user: Impart Believes Tactic Agree_Lifestyle_Objectives
system: Impart D_Committed Agree_Lifestyle_Objectives Import Believes Objectives
Running_Cost Elicit Believes Objectives
user: Impart Believes Objectives Style
system: Impart Believes Objectives Style Not-Known Elicit Believes Objectives Style Confirmation
user: Impart Believes Confirmation
system: Elicit Believes Objectives
user: Impart Believes Objectives Image
system: Impart Believes Objectives Image Agree Import Believes Objectives Image
Justification Social_Status Elicit Believes Objectives
user: Impart Believes Tactic Finish_Tactic
system: Impart D_Committed Finish_Tactic Import Believes Tactic Identify_Parameters
Elicit Believes Tactic Identify_Parameters Agree
user: Impart Believes Agree
system: Impart D_Committed Identify_Parameters Import Believes Tactic
user: Impart Believes Tactic Get_Parameter_Values
system: Impart D_Committed Get_Parameter_Values Import Believes Tactic
user: Elicit Believes Tactic
system: Impart Believes Tactic Scale_Parameters Import Believes Tactic Scale_Parameters
Justification Make_Comparisons_Possible Elicit Believes Tactic Scale_Parameters Agree
user: Impart Believes Agree
system: Impart D_Committed Scale_Parameters Import Believes Tactic
user: Elicit Believes Tactic
system: Impart Believes Tactic Assign_Values Import Believes Tactic Assign_Values Agree
user: Impart Believes Agree
system: Impart D_Committed Assign_Values Import Believes Tactic
user: Elicit Believes Tactic
system: Impart Believes Tactic Assign_Weights Import Believes Tactic Assign_Weights
Justification Reflect_Importance_Of_Objectives Elicit Believes Tactic

338
C.4 Answers to questions

The question sheet filled in by the first four subjects omitted two of the questions (6 and 9) which were included for later subjects, so what follows is the answers of 10 subjects to most questions, and 6 to questions 6 and 9:

1) The experience: has it been tortuous, tedious, tremendous, OK or what?

L: Helpful.
R: Mildly tortuous; I'd say the experience has been - but probably that's not due to the interface - or the indirectness of the interface going through you - it's probably more that I didn't really have an understanding of what the problem comprised. Choosing a car - but I didn't have the idea of it being this linear process with no back-tracking, with no going back to where we started from and so on. It would have been nice to have got an initial... what I would have imagined that we would have been able to do was see what it thought was a good choice and then add another criterion or weight a criterion differently and then see how that changed the choice. We seemed to go through it in a linear way.

J: Interesting, and fun. It was good to use a familiar example. For students, a younger example (or a design problem known to the students) should be used.

M: Confusing. Comments below expand on this. Your items here interact in a way which makes it difficult not to repeat things.

D: With the author at my shoulder to guide me, the process was straightforward. Left to my own devices, I would have floundered.

C: OK. Perhaps slightly tedious - but only because of the interface limitations.

B: Interesting and enlightening.

P: OK

K: OK. The terminology niggles (e.g. "thinks" instead of "believes"). Should split dialogue into sentences. Would be tedious after a time - better to have on-screen editing.

G: OK, but a bit tedious because of the heavy use of unnecessary words such as 'I believe'.

2) The dialogue: was it quirky, sensible, helpful, useless, confusing, flexible...? Ignoring the strange sentence construction, did any of the system's utterances strike you as surprising in any way? If so, how?

L: Quirky, helpful, occasionally incoherent.
R: Not quirky, certainly not useless, it's not confusing, but um...probably helpful. I don't think you could say it's that flexible, but it is helpful.

J: The dialogue was OK. The form could be improved 'incrementally'.

M: Dialogue itself was manageable if idiosyncratic. Perhaps not as informative as it should be.

D: A little quirky, but it would be difficult to produce a system of this kind that would not be. The use of a mouse made things easier.

C: Sensible but idiosyncratic. None of the dialogue was surprising. However, I agreed with the findings most of the time - there was no reason not to agree. A less clearly
defined problem area might have produced a more meaningful dialogue.

B: Slightly difficult to follow. I found the term 'tactic' odd.
P: It would have been difficult to understand without expert interpretation or more familiarity. Apart from this, it appeared to be functional. I was surprised that the system insisted (correctly) on having more parameters. It got slightly muddled about ending.

K: It should have known about windsurfing! It was a bit tricky at times.

G: OK, and No.

3) Guidance and control: did you feel that the system allowed you the degree of control you wanted over the interaction? Did it provide appropriate guidance when you required it?

L: Yes, gave justifications, suggestions.
R: At the time, I didn't feel that the system allowed me the degree of control, but as you explain, you can do all those things that I would have thought that you could do, so it does - it does, and it does provide appropriate guidance - you can ask it what it thinks you should do.

J: The Ann interface was very effective!
M: I did not feel in control until some way into the session. The system's apparent reluctance to accept my choices was frustrating.

D: Its reluctance to allow me to proceed as I wished was a little disconcerting, but otherwise the guidance was fine.
C: No - interface limitations discouraged exploration of the flexibility in control. The control features could be made more explicit. Greater help and guidance is needed during the initial stages of the interaction.

B: Yes and yes.
P: All necessary control seemed to be there. The guidance was somewhat rudimentary, but this was possibly due to lack of persistent questions!
K: Ann interacted.
G: It allowed me some degree of control. It provided an appropriate guidance.

4) The learning environment (i.e. matrix and pull-down menus): did you have access to the information you wanted? Was it readily available? Was there information you wanted to see but couldn't? Any suggestions for improvement?

L: No access to comfort and safety information when I wanted it. Need to be able to see assigned values. Would like to be able to see three columns for each alternative, also left hand columns etc. for all alternatives at once.
R: I'd have liked to see a list of the criteria that are possible, the weighting that are attached to those criteria - of course, that would be a lot of information - how would one browse that amount of information in a sensible way? It would be difficult. There is a lot of information that I'd like to see but couldn't, but on the other hand how do you present that sort of information? It's a difficult trade-off I guess.
J: It is not realistic to expect 'full' information, especially in a pedagogic system. (What would happen with a commercial system?) There should be 'meta-information' about what it knows about.
M: I have no idea. I don't think we ever discovered what the system knew and what I was supposed to do.
D: In the brief time I had to play, I was not sure what information I needed, and was wary of getting too much information.
C: Information in the underlying database could be more accessible. Some kind of running summary of what steps you've already carried out would be helpful.
B: I would have liked to have added further factors which are not available (included) at present.
P: Always available when asked for except when retyping weights - e.g. previous values were not visible. Also, after the weighting calculation previous results were lost when proceeding to change parameters etc.
K: Didn't have enough information about cars. (Using cars forces you into a particular
area.) It's on the right lines, but it needs to be made easier to use, without so many changes of windows.

G: Yes, Yes, No, and A very good learning environment.

5) The approach of having both learning environment and dialogue component: does it or does it not have any pedagogical advantages over either a simple learning environment (e.g. a commercial spreadsheet) or a stand-alone dialogue component? Please consider potential advantages of all three possibilities - and of any other configurations you can think of.

L: It felt like learning about choosing a car, not invited to reflect or focus attention on ordering or type of tactics, or which included etc. Needs debriefing. Higher, meta-level of dialogue about dialogue, or set task to summarise what happened using printout of trace. Other configuration: add dialogue about trace.

R: It's a very general question - it's almost a philosophical question. What you're asking there is do I believe that computer based tutors can be useful, and I'd say I think so - I won't rule them out as a possibility. If you think of Excel and try and imagine a tutor that could produce a useful dialogue between a novice user and Excel I think you've got a problem that's 6 or 7 orders of magnitude harder than the one you've got now. It's one of these problems of scalability: can you scale that sort of technique up to some scenario that's only slightly more complex? And I just have this feeling that there's probably a lot of problems in going to a slightly more complex scenario that won't scale linearly.

To conceive of using Excel in the same way, they'd need a lot of background knowledge, whereas if they walk up to your system that background knowledge is sort of encapsulated - they're sort of led into it, which is probably an advantage. I think it depends a lot on the person who's forming the intention to do that task - what sort of background knowledge they have.

J: Learning environment and dialogue together are important. Especially when the system prompts for information the user did not think of themselves (but agreed was important).

M: I find it difficult to comment on this. The system appeared far removed from what I recognise as computer-aided learning or CBT and I am not familiar with other uses of computers in "learning" - but what do you mean?

D: The simple learning environment is valuable as it is without preconceived controls and allows students to develop ideas freely. The presence of the dialogue component should be judged to accelerate the learning process but without leading the student down a pre-ordained route. The dialogue component alone would be too constraining.

C: The dialogue environment is essential for naive users. However, you can also imagine the situation where the problem is clearly defined or the user is familiar with the problem area. Here the dialogue content becomes less useful and it is desirable to 'skip over' it. I like the idea of a combined system which is dynamically 'tuned' to the needs of the user - dialogue driven, learning environment driven, or both.

B: No answer. ("I'm not familiar with the options")

P: Advantage that with a pre-set problem the user can be guided to a more detailed and deeper analysis than he might have undertaken on an alternative system. Advantage is to some extent diminished by difficult dialogue structure, but that could be improved.

K: Yes, it does. It obviously does; a spreadsheet should be invisible. Dialogue on its own would be little better than a book. You need to bring them together. It's what we're trying to do here.

G: I think it is an excellent idea to have both.

6) The approach of making explicit the tactics adopted / decision steps taken: do you believe that this has any pedagogical value?

D: At an early stage in the design process it may not be realised what decision steps are necessary and so explicit guidance is useful. At later stages a freer hand may be
Appendix C: Evaluation of WOMBAT

better.
C: It is not clear to me what is meant by the term 'tactics' in this context. It is always useful to see and manipulate the 'rules and constraints' used during the decision making process. The usefulness varies according to the complexity of the subject area.
B: Yes - a structured way of thinking things through as in planning is very desirable
P: Particularly in respect of rating (I think he meant normalising) and weighting (i.e. the tactics he hadn't thought through before). Generally yes.
K: Yes.
G: Yes.

7) Potential for use in education: Does the prototype have any? If so, what do you consider are its potential strengths and weaknesses? In what ways do you believe it needs to be modified or extended to have any educational value?
L: See 5. Also extend to argument about other determinable and closed domains.
R: Again, it seems like a bit of a philosophical question, and again I'd say yes, I think there's scope for intelligent tutoring systems, and a component of that is dialogue. I think there is scope for it in education.
J: See the answer to next question.
M: Perhaps this is the question I addressed under point 5. I find it difficult to tell. I fear, yet again, we are speaking quite different languages.
D: Its main strength is that it gives the students a structure within which to work at the early stages. There would have to be, however, means for allowing the student to take greater control as confidence is gained.
C: Obviously the interface requires considerable improvement. I didn't really feel sufficiently at ease with the system to explore its flexibility. I think flexibility and the ability to explore 'what if' situations provide the educational potential.
B: Yes - it makes the user think about the factors that really matter, i.e. the specification - also about the sensitivities of these factors. It needs to be more user-friendly in terms of the display of questions and answers.
P: Definitely has potential for education. The dialogue needs to be cleaned up. Need to be able to look back to earlier stages of assessment, I am not sure how well an unguided user would cope with complexity of screens, menus etc.
K: General strengths are that computer tends to maintain interest longer, particularly if it's quick and flexible. But we underestimate the difference between a prototype and a final system.
G: Certainly. Its strength is that it demonstrates a correct way of deciding. Its weakness is that it doesn't explain why you should not do what you should not do.

8) In particular, do you believe this approach has any potential in the declared context of design education?
L: Good for learning about processes, procedures.
R: It's probable that design education is no different from other education - that having dialogue attached to a matrix is probably a good idea.
J: I think this kind of system has great pedagogic potential:-
1) It makes criteria and constraints explicit. Usually they are not, and design students often work with many implicit criteria and values.
2) It illustrates information elicitation (designers have to do this).
3) Students will be stimulated to consider this kind of system as a tool
   a) at the design stage (especially early on),
   b) at the retail end to make sure customers understand the relative advantages of the product.
4) Evaluation: this system normalises and adds numbers together to get a 'score'. Methodologically this is very dubious since it is based on chalk and cheese arithmetic. Good! Make students think about these things. How are trade-offs made?
5) Emphasise changing criteria.
6) Making a virtue out of necessity - exploit deficiencies to make teaching points.
M: NO! But a more detached reaction would be that you clearly have some formalised view
of selection between "alternatives" (more correctly, 'options') but it did not come through. Only in parts did it apparently coincide with mine. In OU teaching terms, some AIMS and/or OBJECTIVES would have been exceptionally helpful.

D: Design means beginning with a blank sheet of paper, a most intimidating experience for all students. Reiterating what I have already said, if the initial problem can be overcome by means of this type of program, and I think it can, then it will be extremely useful.

C: Absolutely. The pure 'learning environment', as you call it, whether it is a spreadsheet or a complex CAD package, has little values without guidance and feedback on what constitutes a 'correct formulation' of a solution to a problem.

B: Yes - as above, it makes the user think more deeply about the design specification.

P: See answer to 7.

K: Yes, because design is an iterative decision making process.

G: Yes, within a particular area of design.

9) if the answer to (8) was broadly affirmative, then which skills do you see it as having the potential to develop in students?

D: In the context of design, its main use will be to help the student define a specification for the object to be designed. This is a crucial step and needs to be done very carefully. Once a good specification has been drawn up, the detailed design process can be quite straightforward.

C: In the area of design, I would say this approach should allow users to painlessly get 'a feel for', and learn, what constitutes a 'correct' design. This allows more freedom to explore other aspects of design - such as aesthetics - which are less amenable to automated dialogue handling.

B: as above

P: Identification of key parameters. Assignment of rating and weightings as an aid to decision. (This relates to tactics, he said in follow-up discussion.)

K: Decision making!

G: Quantitative approach to selection of one out of several alternatives.