

**Intelligent Student Systems:
an Application of Viewpoints
to Intelligent Learning Environments**

David M. Nichols
BSc(Hons)

Submitted for the degree of
Doctor of Philosophy

Computing Department
Lancaster University

September 1993

**Intelligent Student Systems:
an Application of Viewpoints
to Intelligent Learning Environments**

David M. Nichols
BSc(Hons)

Submitted for the degree of
Doctor of Philosophy

September 1993

Abstract

Intelligent Student Systems are a class of Intelligent Learning Environments that place the learner in the role of a tutor rather than a student. In an analogy with the educational practice of peer tutoring users learn by teaching the computer – inverting the predominant ‘computer as tutor’ metaphor. Intelligent Student Systems emphasize the learner’s viewpoint in educational interactions in preference to the system’s conception of the domain. These systems are considered to be less complex than Intelligent Tutoring Systems and to have the potential to generate novel human-computer educational interactions.

Viewpoints also have an integral part in knowledge representation in Intelligent Learning Environments and they are utilised in the design and implementation of an Intelligent Student System in economics. Testing of the system produced insights into the future application of Intelligent Student Systems.

To my parents

Acknowledgements

I would like to thank:

my supervisor, John Self, for constant support, encouragement and informed comments

Jo Cove, Angela Lucas and the other people at Logica Cambridge

Michael Pengelly, Michael Twidale, Gary Spiers, Sean Butler and assorted other members of CSALT for numerous constructive discussions

Gerry Steele and Paul Ferguson of the Department of Economics in the Management School

Andy, Tracy, Richard, Nick and Chris for keeping me relatively sane

the experimental subjects for their cooperation

Financial support was provided by the Science and Engineering Research Council (SERC) and Logica Cambridge Ltd.

Intelligent Student Systems: an Application of Viewpoints to Intelligent Learning Environments

(Note: due to various formatting problems the page numbers below are wrong)

Contents

Chapter 1 – Intelligent Learning Environments

1.1 Introduction.....	1
1.2 The Space of Educational Interactions	2
1.2.1 Locating ILEs in Educational Space.....	3
1.2.2 Dialogue Structure.....	8
1.2.3 The Rationale for Intelligent Student Systems.....	10
1.3 Knowledge Use in Intelligent Learning Environments.....	11
1.3.1 Domain Knowledge.....	12
1.3.2 Learner Model.....	15
1.3.3 Communication Strategy/Pedagogical Knowledge.....	17
1.3.4 Interface.....	17
1.3.5 Summary	18
1.4 Chapter Summary	18
1.5 Thesis Structure	20

Chapter 2 – Implemented Viewpoints

2.1 Introduction.....	21
2.2 Implemented Viewpoints in Educational Systems.....	21
2.2.1 Viewpoints in Domain Models.....	21
2.2.2 Viewpoints in Learner Models.....	27
2.2.3 Viewpoints in the Communication Strategy.....	29
2.2.4 Viewpoints at the Interface.....	30
2.3 Non-Educational Implemented Viewpoints.....	31
2.3.1 Knowledge Representation Systems.....	31
2.3.2 Databases	36
2.3.4 Others.....	38
2.4 A Comparison of Implemented Viewpoints.....	39
2.5 Chapter Summary	40

Chapter 3 – Non-Implemented Viewpoints

3.1 Introduction.....	42
3.2 Intelligent Learning Environments	42
3.2.1 Wenger: Interpretative Contexts.....	42
3.2.2 Self: Belief Sets.....	43
3.2.3 Other ILE Work.....	44
3.3 Mental Models.....	45
3.4 Linguistics.....	46
3.5 Artificial Intelligence	48
3.6 Synthesis.....	48

Chapter 4 – Intelligent Student Systems

4.1 Introduction.....	51
4.2 Related Educational Research.....	53
4.2.1 Peer Tutoring.....	53
4.2.2 Learner Beliefs in Tutorial Groups.....	59
4.2.3 Socratic Tutoring.....	61
4.3 Related Systems.....	63
4.4 Intelligent Student Systems.....	65
4.4.1 The Components of an ISS.....	65
4.4.2 ISS Architecture.....	68
4.5 Synthesis.....	70

Chapter 5 – DENISE: An ISS in the Domain of Economics

5.1 Introduction.....	72
5.2 The Domain of Economics	72
5.2.1 The Structure of the Domain of Economics.....	72
5.2.2 Related Work in Computational Economics	75
5.2.3 Summary	80
5.3 DENISE: An Experimental ISS.....	81
5.3.1 The Learnt Model and Conceptual Syntax in DENISE.....	81
5.3.2 Model Functions in DENISE.....	83
5.3.3 The Interface in DENISE.....	86
5.3.4 Dialogue Strategy in DENISE	90
5.4 Summary.....	95

Chapter 6 – Experimentation

6.1 Introduction.....	96
6.2 The Experimental Dialogue Strategy.....	96
6.3 Experimental Methodology.....	100
6.4 Experimental Procedures.....	103

Chapter 7 – Experimental Results

7.1 Introduction.....	104
7.2 Experiences with the Interface.....	104
7.3 Dialogues.....	106
7.4 Models.....	109
7.5 Summary.....	111

Chapter 8 – Conclusions and Future Work

8.1 Conclusions.....	113
8.2 Future Work.....	116
8.3 Endpiece: A One Sentence Summary of the Thesis.....	118

References.....	119
------------------------	------------

Appendix A – Dialogue Strategy Functions in DENISE

A.1 Introduction.....	132
A.2 Model Access Functions.....	132
A.3 Model Maintenance Functions.....	133
A.4 Interface Functions.....	134
A.5 Programming Functions.....	134

Appendix B – The Experimental Dialogue Strategy

B.1 Strategy.....	136
B.2 ‘economics_exp1’.....	136

Appendix C – Experimental Examples

C.1 Examples.....	142
-------------------	-----

List of Figures

	Page
1.1 Agent Knowledge Space.....	4
1.2 Conventional Teaching in Agent Knowledge Space.....	4
1.3 Educational Interactions.....	6
1.4 Dialogue Structure in an ITS and a LBT system.....	9
1.5 A Spectrum of Formality	1
4	
2.1 An outline structure of a viewpoint from [Moyses, 1990]	2
5	
2.2 Beliefs about and of John from [Ballim, Wilks & Barnden, 1991]	3
2	
2.2 Classification of viewpoints from [Self, 1992]	4
4	
4.1 Theoretical Architecture of an ISS	6
9	
5.1 Three interview extracts with an economist from [Huxor, 1988]	7
3	
5.2 Examination questions from [Lancaster-University, 1990]	7
4	
5.3 The Action of a Model Access Function	8
4	
5.4 Some Model Access Functions	8
5	

5.5 A typical dialog window in DENISE	8
7	
5.6 A menu constraining input to the conceptual syntax	8
8	
5.7 The dictionary of previously used terms	8
8	
5.8 The user takes control of the dialogue	8
9	
5.9 The Structure of a Node in the Dialogue Strategy	9
0	
5.10 The Main Screen of Plan Manager	9
1	
5.11 Creating a New Node in the Node Editor	9
3	
5.12 The Structure of 'Simple Plan'	9
4	
5.13 DENISE's record of a dialogue running 'Simple Plan'	9
4	
7.1 A portion of DENISE's dialogue record of subject B	1
07	

List of Tables

	Page
1.1 Classes of Educational Interactions.....	3
1.2 Relative frequency of ILE domains	1
3	
2.1 View Types identified in [Acker, et al., 1991]	3
8	
2.2 A Comparison of Implemented Viewpoints	4
0	
4.1 Selected questionnaire responses from the 'Pimlico Connection' peer tutoring scheme from [Goodlad & Hirst, 1989]	5
6	
4.2 Fragments of a dialogue on growing grain from [Collins, 1977]	6
2	
6.1 Evaluation Methods for Exploratory ITS Research from [Murray, 1993]	1
01	
7.1 Feature use by the experimental subjects	1
05	
7.2 Percentage of unique concepts generated	1
10	

Chapter 1

Intelligent Learning Environments

1.1 Introduction

This is a thesis about Artificial Intelligence and Education – the production of knowledge-based software that is of educational benefit to its human users. This field is often referred to as Intelligent Tutoring Systems (ITSs) or Intelligent Computer-Aided Instruction (ICAI), however as neither tutoring nor instruction constitute an adequate description of educational activities we will follow [Dillenbourg, 1991] in using the term Intelligent Learning Environments (ILEs).

Any system which manipulates knowledge encounters a range of problems involved in representing, maintaining and reasoning with it. These problems are compounded in interactive systems by real-world constraints such as response time. There are many difficulties in dealing with knowledge, not least defining what it is, but we can be sure of one thing – people (or more generally agents) disagree about it.

When agents place different interpretations on information there exists the possibility of a failure in communication. These failures happen to all of us every day and are usually minor and easily rectified. However failures in interacting with computers are more difficult to recover from as computers have fewer communication modes, lower bandwidth, less common sense etc.

Successful communication is a pre-requisite for effective education – [Douglas, 1991] reports that 20% of human teachers' time is spent repairing communication failures arising from mistaken mutual beliefs. Communication is based on the knowledge of the agents involved in the dialogue. However agents' knowledge, about both the domain and other agents, can be (re)-interpreted in different ways and it follows that an ILE should support these alternative interpretations. The term that will be used to denote these different interpretations is *viewpoints*.

Two quotations will serve to outline recent viewpoints on viewpoints:

'In spite of its importance, this topic has not been addressed by the field in principled ways. In its full generality, the problem of viewpoints is admittedly

very complex and difficult. However, restricted versions can still be very useful in instructional contexts and the topic seems ripe for more research.'

[Wenger, 1987] pp 355

'These issues ... indicate that the use of multiple viewpoints is of profound and general importance in the design of tutoring systems.'

[Moyse, 1990] pp 2

This comment is overly specific: viewpoints are of profound and general importance in all knowledge-manipulating systems. Viewpoints are relevant to all the knowledge-based activities involved in constructing an ILE: problem solving, domain modelling, agent modelling, explanation, diagnosis, planning, knowledge acquisition etc.

Although the term viewpoint is used by several authors there is no commonly agreed definition so that even though they refer to the same general concept one author's viewpoint is not the same as another's. The definitions of viewpoint proposed by [Ballim & Wilks, 1991; Moyse, 1991; Self, 1992; Wenger, 1987] are not interchangeable. Similarly some authors refer to contexts, perspective, worlds, representations, conceptual point of view etc. which all are equally relevant to viewpoints. A broad informal understanding of viewpoints will be sufficient until some landmark paths have been laid through the jungle of terminology.

There are, however, two general senses of the term viewpoint:

- viewpoints in the technical sense of representing and using different interpretations in ILEs.
- the desire to explore the learner's viewpoint rather than impose the system's interpretation of the domain [Self, 1988a]. This second sense has been exemplified by systems that eschew 'tutoring' in favour of other activities such as collaboration [Dillenbourg & Self, 1992].

This thesis addresses both of these issues by utilising technical viewpoints in a system design that emphasises learner beliefs. These two approaches are complementary and take the form of examining the role that an ILE takes in an educational interaction (section 1.2) and the use of knowledge within an ILE (section 1.3).

1.2 The Space of Educational Interactions

The space of educational interactions contains every situation where at least one agent learns something and is infinite. However it is

Class	Components	Examples
1	Human-Human-Environment	Conventional Schools
2	Human-Computer-Environment	Intelligent Learning Environments
3	Computer-Computer-Environment	Computer Collaboration
4	Human-Environment	Computer-Assisted Instruction
5	Computer-Environment	Machine Learning

Table 1.1 Classes of Educational Interactions

possible to impose some structure to illuminate the important differences between interactions.

In this section the field of Artificial Intelligence and Education is located relative to conventional educational systems. The dialogue structure of an ITS is examined to identify the sources of misinterpretations and the rationale for an alternative ILE is outlined.

1.2.1 Locating ILEs in Educational Space

The educational universe can be divided into agents and the environment. Agents [Newell, 1982] have a body of knowledge, goals and actions; anything else is environment, non-agent computers (including microworlds), objects, geography etc. Agents can be of two types, human or computer-based. Given that the environment is always present there are five classes of possible educational interaction as shown in Table 1.1.

The classification distinguishes between computer systems that use explicit knowledge in interactions, such as ILEs, and those that follow instructions determined at compile-time. As we are interested in interactive systems the crucial criterion is that of modelling other agents. Any system in classes 2 or 3 must maintain some minimal model of other participants in the interaction.

As 1, 2 & 3 and 4 & 5 are similar in form we can expect to find analogies between their elements; the ITS-human tutor (2<-->1) analogy is particularly common. In this thesis we will not consider systems in class 5 and will only discuss classes 1 and 4 as a source of ideas for the *interesting* systems of classes 2 and 3. Some agents can interact with both human and computer-based agents and so can migrate between classes.

The next piece of order we can impose on the interaction space is based on the knowledge that different agents possess. We will make the simplifying assumption that agents' knowledge can be partitioned into two areas: communication/pedagogical knowledge and domain knowledge:

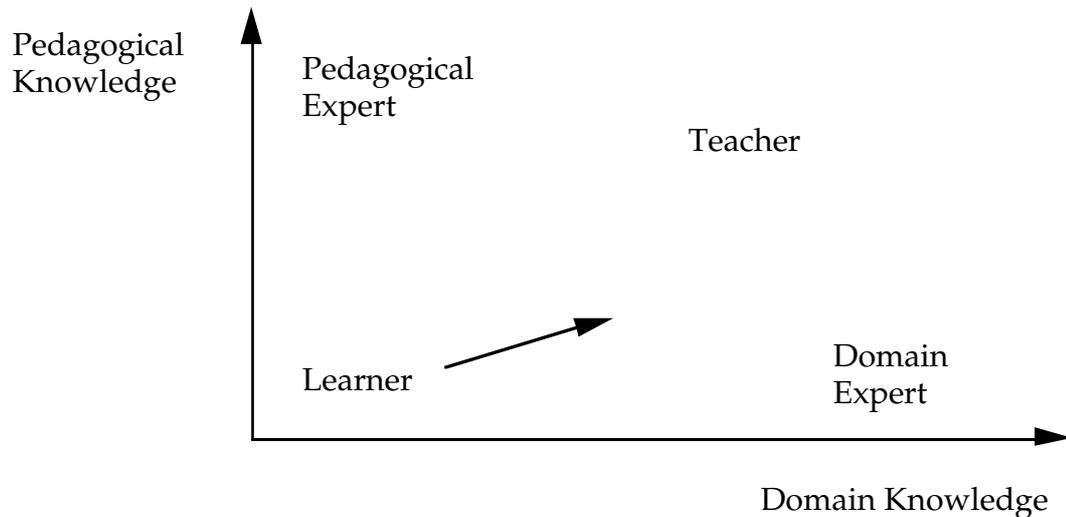


Figure 1.1 Agent Knowledge Space

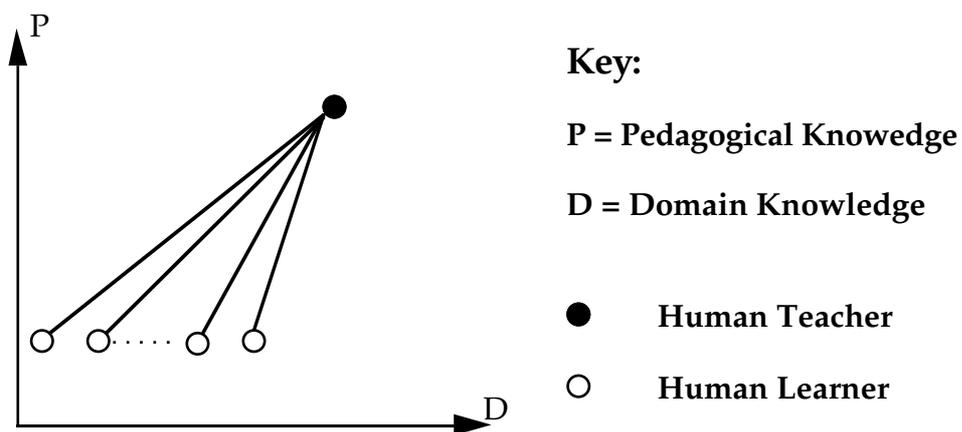


Figure 1.2 Conventional Teaching in Agent Knowledge Space

There are two key places for intelligence in an ITS. One is in the knowledge the system has of its subject domain. The second is in the principles by which it tutors and in the methods by which it applies these principles.

[Anderson, 1988]

Consequently agents can be located in a 2-dimensional knowledge space, see Figure 1.1. The arrow indicates a possible path for a learner.

This knowledge space can be used to describe educational interactions, a typical classroom can be represented as in Figure 1.2. The teacher (solid circle, top right) interacts with many learners (hollow circles, bottom left).

As teachers possess both knowledge of the domain and of pedagogy we will further restrict ourselves to considering only learners and teachers/tutors in two-agent scenarios. The two experts can be approximated as subsets of the teacher as both types of knowledge are required for successful tutoring [Anderson, 1988]. These two agents, learners and teachers, in interaction classes 1, 2 and 3 are shown in Figure 1.3.

Figure 1.3a shows three types of human-human educational interaction. 'Tutoring' is the prevalent style of teaching in a one-to-one relationship where the tutor agent has both pedagogical knowledge and domain knowledge. In 'Peer Tutoring'¹ the tutor agent is a learner who is only slightly more knowledgeable about the domain than the other 'learner' agent (or tutee) [Halff, 1988].

'Peer tutoring is a system of instruction in which learners help each other and learn by teaching.'

[Goodlad & Hirst, 1989] pp13

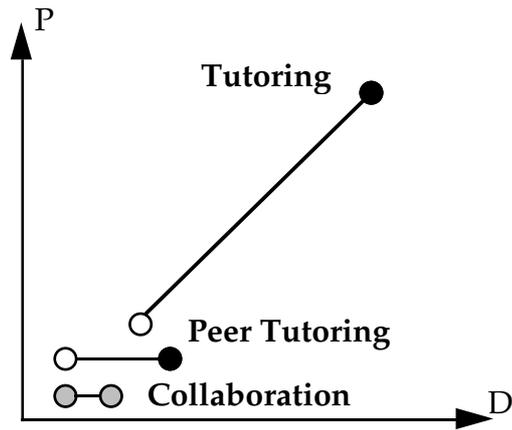
'Collaboration' refers to interactions where the learners cooperate over some problem and is 'characterised by a symmetrical interaction among learners' [Dillenbourg, 1991] pp60.

Figure 1.3b shows three types of human-computer educational interaction. 'ITS' shows the predominant type of ILE, one that takes the role of a tutor agent. 'Learning by Teaching' shows a scenario where the human 'tutee' agent in a 'Peer Tutoring' interaction is replaced by a computer agent. 'HCCL' (Human-Computer Collaborative Learning) [Dillenbourg, 1991] describes situations where a human agent and a computer agent collaborate on some problem.

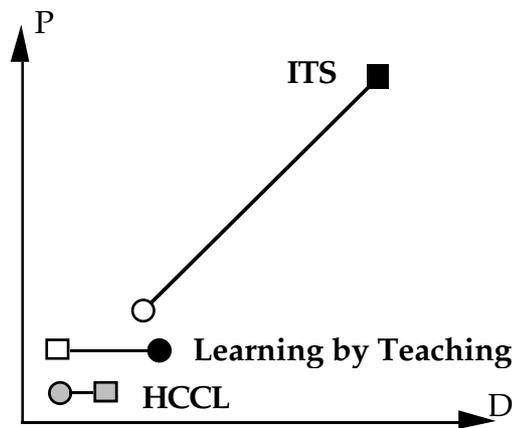
Figure 1.3c shows three types of computer-computer interaction. These scenarios do not fulfil the initial requirement of being of educational benefit to their human users. However as they contain the same types of agents as in Figure 1.3b they are useful in the development

¹ 'The literature ... can confuse in its uneven use of terminology. Peer tutoring if literally interpreted implies equality of status and merit, which is untrue ... for many peer tutoring initiatives refer to encounters between advanced and less advanced students' [Saunders, 1992]. This is the interpretation that will be used subsequently.

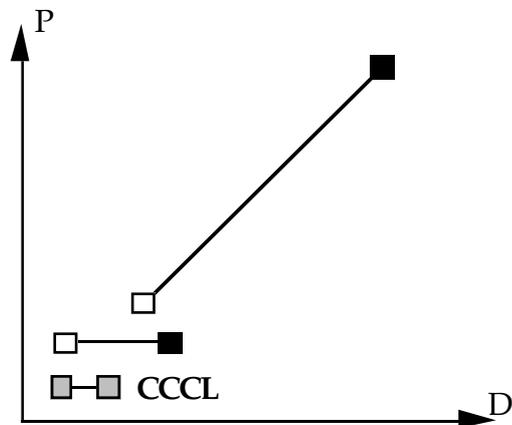
of ILEs and thus indirectly satisfy a relevancy criterion [Petrie-Brown, 1990]. CCCL (Computer-Computer Collaborative Learning), adapting [Dillenbourg, 1991]'s terminology slightly, describes two computer agents collaborating on some problem. The other two configurations, of an artificial tutor with an artificial student, do not appear to have accepted names although this type of interaction has been suggested to explore tutorial dialogues [Petrie-Brown, 1989]. We will refer to such interactions



a) Human-Human Interactions
(Conventional Education)



b) Human-Computer Interactions
(Intelligent Learning Environments)



c) Computer-Computer Interactions
(The Artificial Classroom)

Key:

	Learner	Tutor	Collaborator
P = Pedagogical Knowledge	○	●	◐
D = Domain Knowledge	□	■	◑

Figure 1.3 Educational Interactions

as taking place inside *The Artificial Classroom*.

The dominant analogy between the three classes of interaction shown in Figure 1.3 has been that between ITS and tutoring. Recently more attention has been paid to collaborative systems that depart from the classic ITS architecture [Chan & Baskin, 1990; Cumming & Self, 1989; Dillenbourg, 1991].

The Learning Companion System (LCS) [Chan & Baskin, 1990] is a three-agent framework, computer teacher, computer companion and human student, that emphasises the social context of learning. LCSs encompass a vast range of ILE configurations as the computer agents can take on different roles²; e.g. a computer companion can be a collaborator or a competitor [Chan et al, 1992]. The scope of LCSs is so vast that only a small proportion of scenarios have been examined.

[Palthepeu, Greer & McCalla, 1991] have suggested that learning by teaching may be a fruitful approach in replicating some of the effects of peer tutoring schemes [Berliner, 1989; Goodlad & Hirst, 1989; Goodlad & Hirst, 1990; Topping, 1988]. Learning by teaching is technically a subset of a teacher-less LCS where a computer companion collaborates with the human student by taking the role of a student. However the LCS paradigm appears to be more focused on collaboration on some given problem (e.g. integration problems [Chan & Baskin, 1988], game playing [Chan, et al., 1992]) than with an overtly metacognitive activity such as teaching. The Artificial Classroom remains almost totally unexplored with the only known example being some experiments with People-Power [Dillenbourg, 1991].

Of the three ILEs shown in Figure 1.3b a Learning by Teaching (LBT) system has received scant attention – just one system [Michie, Paterson & Hayes-Michie, 1989]. Is there any reason to believe that such a system may experience reduced communication failures? To answer this question we will examine the differences between an ITS dialogue and a LBT dialogue.

Although both dialogues have a tutor and a learner the learner performs a different role in an LBT scenario. The LBT learner is not really there to learn at all, but to act as a student in such a way that the human

² Or even disappear entirely. The computer teacher can be removed to leave the student and companion in a collaboration scenario.

tutor learns. An ILE that behaves in this way is called an *Intelligent Student System* (ISS).

This distinction also highlights the differences between LBT and machine learning systems. A definition of machine learning: ‘learning denotes changes in the system to do the same task or tasks drawn from the same population more effectively the next time’ [Simon, 1983]. A definition of an ISS: a system that acts as a student such that an agent acting as a teacher learns about the content, structure and limitations of its own knowledge.

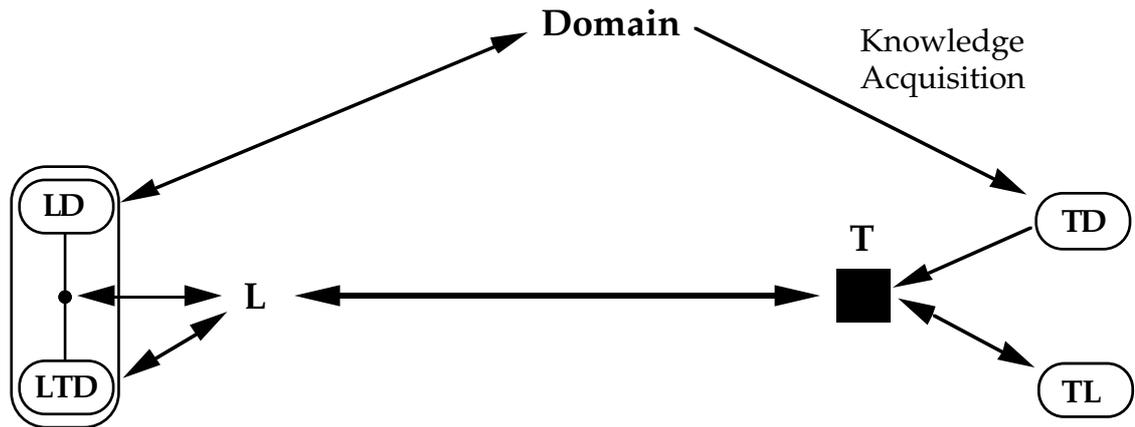
1.2.2 Dialogue Structure

Figure 1.4a shows an ITS dialogue about some distinct domain (D) between a tutoring system (T) and a human learner (L). T has a domain model, TD, acquired from domain experts and L also has a domain model, LD, acquired from past experiences. L also has some domain knowledge that has been taught by T, LTD. T has a student model, TL, containing knowledge about L’s capabilities.

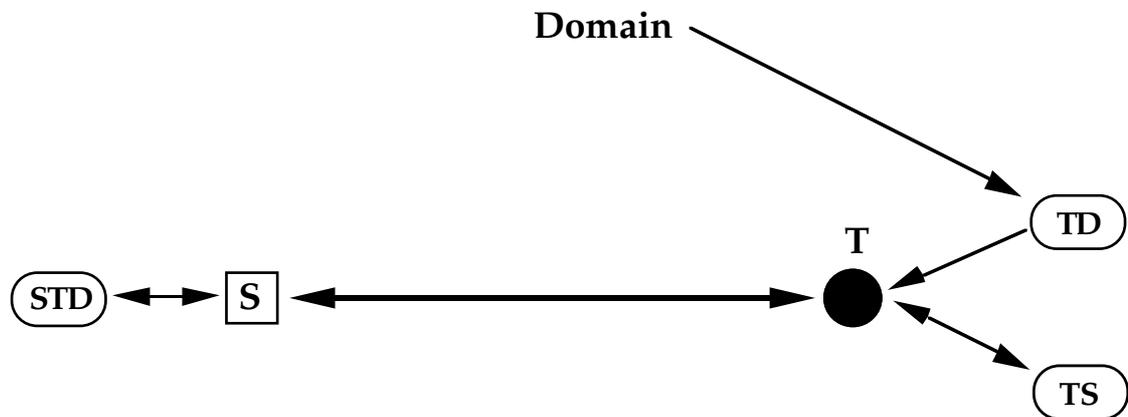
The potential for misinterpretation is clear, L has two sources of domain knowledge – the tutor and the real domain. There is no *a priori* reason why the learner’s interpretation, LD, and the expert interpretation, TD, should be the same. Indeed given that these models are created through radically different processes there is some reason to believe they will be significantly different. These misinterpretations are present even if we assume a perfect interface and that the learner doesn’t independently draw mistaken analogies with other domains. An ITS is fighting an uphill battle against misinterpretation even before it starts.

Figure 1.4b shows an LBT dialogue between a human tutor (T) and an ISS (S). The tutor has a domain model (TD) acquired from past experiences and a model of the student (TS). The ISS has a model of the domain (STD) *solely* drawn from the tutor so the tutor’s domain model and the domain model are the same model for S. The conflict between the domain and the tutor is no longer present. In addition if the interface is effective then after some time STD and TS will be similar and both will be closely related to TD.

An alternative approach is to consider the central object of the dialogue. In an ITS the central object is the difference between the domain model and the student model based on the student’s actions. In a LBT dialogue the central object is the human tutor’s domain model. As the existence of the ILE is based on the prospect of enhancing (in some way)



a) ITS dialogue



b) LBT dialogue

Key:

T	Tutor	a D	Domain Model of Agent a
L	Human Learner	a S	Model of Student/Learner held by Agent a
S	Artificial Student	a L	Model of Student/Learner held by Agent a
		a TD	Model of Tutor's Domain Model held by Agent a

the human agent's domain model it seems only natural that it is *this* model that should be the central object of the interaction.

Figure 1.4 Dialogue Structure in an ITS and a LBT system

There are theoretical reasons for believing that a LBT system may be subject to fewer communication failures arising from misinterpretation

than other ILEs. This does not mean we can say that a LBT system will deliver a better (or worse) educational experience than existing ILEs as the user of a LBT system will experience a radically different interaction from an ITS user.

1.2.3 The Rationale for Intelligent Student Systems

The previous section outlined a communication-based justification for investigating LBT systems; this section describes the other arguments.

As with many areas of artificial intelligence it has been easier to move from ideas to experimental systems than from the computing laboratory out into the real world. The idealised view of an intelligent tutor for every learner has had to be squared with the reality that *successful ITSs are difficult to build* [Burns & Capps, 1988; Woolf, 1988]. The absence of widespread ITS-usage in the educational system is consistent with this view – although it could be due to other factors such as teacher resistance.

In terms of Figure 1.1 it is proposed that the further an agent is from the origin the more knowledge is required and the more complex it is to build: intelligent students are easier to construct than intelligent tutors. There are two justifications to support this proposition:

- intelligent students do not need to model all of the tutor - only the domain
- intelligent students do not need a pre-defined domain model

The first justification states that a system with one model (see Figure 1.4) to reason about is simpler than one which has to combine two models [Johnson-Laird & Byrne, 1991]; the domain and agent models merge into one. Also, a student does not require the pedagogical knowledge necessary to coordinate the domain and student models of a tutor. Secondly, the knowledge acquisition effort required to generate a domain model is not needed.

The lack of any LBT work is a further strong argument for such an investigation. At the very least it would be valuable to know if LBT systems were totally useless. Not only to appropriately channel research activities but to identify human roles that computer-based agents can't adequately fill. We do not believe that this is the case but the informational returns justify the analysis irrespective of our beliefs.

The idea of evaluating ITSs through intelligent students is analogous to the concept of a programmable user model [Young, Green & Simon, 1989] for evaluating user interfaces.

A Programmable User Model (PUM) is a psychologically constrained architecture which an interface designer is invited to program to simulate a user performing a range of tasks with a proposed interface.

[Young, Green & Simon, 1989]

If the PUM is difficult to program to achieve some task on a system then it is suggested that a user may also find the task difficult. The PUM provides a mechanism for making the designer's assumptions about users explicit. For an ILE designer the user is a student and programming an intelligent student may make assumptions about students' behaviour explicit. PUMs are not artificial users but they force designers to think about their systems from the user's point of view.

Finally, intelligent students provide a consistent, modifiable, compliant, cheap, glass-box population of test subjects for ILE experiments. Before releasing ILEs on human students intelligent students allow simulations of possible interactions to evaluate an ILE. If an ITS cannot successfully tutor a student agent then it is unlikely to be successful in dealing with unpredictable and inconsistent human students. So intelligent students can function as a lower bound for evaluating the effectiveness of some ILEs. Also intelligent students could be used to train student teachers [Sandberg, Barnard & van-der-Hulst, 1992].

The arguments for investigating systems based on intelligent students can be summarised as:

- the potential replication of beneficial peer tutoring effects
- more effective dialogues through improved communication via reduced misinterpretation
- ease of construction advantages over ITSs
- providing an aid for ILE designers
- providing an evaluation environment for ITSs

The full arguments are presented in section 4.1.

In fact the rationale for exploring intelligent students rests on the disjunction of these five reasons – only one has to be true for there to be a valid justification for a LBT project. From the overall ILE perspective it

perhaps makes more sense to try to walk (build artificial students) before trying to run (build artificial tutors).

1.3 Knowledge Use in Intelligent Learning Environments

Any piece of software embodies the knowledge of its designer(s) and any experts they choose to call upon. In addition ILEs create and maintain a real-time model of the user (hopefully learner) which enables them to provide an individualised adaptive interaction. The ITS field (here referred to as *knowledge communication*) has converged on a four-way division of system components:

‘we follow a natural division of the task of knowledge communication into four distinct components, each corresponding to a distinct section: domain expertise, model of the student, communication strategies or pedagogical expertise, and interface with the student.’

[Wenger, 1987] pp 14

Informally this division can be regarded as relating to the questions:

- what are we communicating about? (Domain, section 1.3.1)
- who am I communicating with? (Learner model, section 1.3.2)
- how to decide what to do next ? (Communication strategy, section 1.3.3)
- in what ways can I communicate? (Interface, section 1.3.4)

The following sections will briefly outline the importance of viewpoints to these four components of an ILE.

1.3.1 Domain Knowledge

or what are we communicating about?

Most artificial intelligence (AI) software does not attempt to behave intelligently in all situations but instead restricts its coverage to specific contexts. ILEs follow the same path; SOPHIE [Brown, Burton & Kler, 1982] can conduct a dialogue about an electrical circuit but is less comfortable on insect biology. The domain knowledge embodied in SOPHIE does not extend beyond the limits determined by its designers. Almost all ILEs use this type of static pre-defined domain model. An exception was the self-improving integration tutor [Kimball, 1982] which added superior student solutions to its domain model.

Domain	Number	Percentage
Mathematics	17	35.4
Computing	14	29.2
Electrical	3	6.3
Languages	3	6.3
Medical	2	4.2
Meteorology	2	4.2
Others	7	14.6
All	48	100.0 ³

Table 1.2: Relative frequency of ILE domains

The domains covered by ILEs so far are not a representative sample of all possible domains. Work has concentrated on mathematics, physics, computer programming and has only occasionally ventured into non-formal domains [Goodyear & Stone, 1992]. Combining two surveys of the field forty-eight different ITSs and learning environments were identified [Nwana, 1990; Ross, 1987], see Table 1.2.

Table 1.2 shows that domains with a formal background dominate ILE research. This may be related to the experiences of ILE researchers but we believe this distribution is mainly a consequence of domain structure. It is easier to represent and reason with domains based on formal languages such as the rules of algebra or the syntax of Pascal.

Intelligent tutors are most easily specified for closed, formal domains such as constructing proofs in geometry and solving algebraic equations.

[Nathan et al, 1989]

These subjects exhibit the atypical characteristic of a single commonly agreed correct domain model. The majority of domains do not possess such a sound foundation.

Domains can be partially ordered along a formality spectrum, see Figure 1.5. Formal domains on the left of the spectrum are characterised by a commonly agreed core whereas the further to the right a domain is

³ Figures do not add to 100 because of rounding.

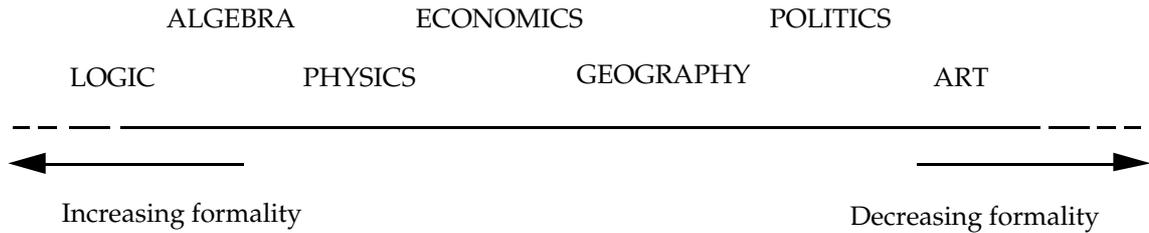


Figure 1.5 A Spectrum of Formality

positioned the more subjectivity and disagreement it is likely to possess. ILE research has been predominantly located on the left of this spectrum.

Domains in the social sciences and humanities do not have this formal framework to build on, they are concerned with the real world which does not behave in such predefined ways. The introduction of people into the domain means that a mathematical consensus is less likely and the domain consists of different experts with differing views [Goodyear & Stone, 1992].

The concentration of ILE research in formal domains has led to the implicit acceptance of the domain model as a single correct representation. The implication is that any learner that deviates from it is in error: leading to terminology such as bugs and mal-rules [Sleeman, 1983]. This approach is sustainable in closed formal domains but is insufficient for the majority of more complex domains where 'bugs' can arise from valid alternative interpretations.

The formality spectrum can only ever be an approximation as domains can exhibit differing degrees of formality at different grain sizes. The choice of the grain size for an ILE is dependent on the tasks the domain knowledge is designed to perform. The top-level description is that in order to aid a learner about some topic an agent must have some knowledge about that topic. However this general statement can be decomposed into sub-tasks.

Domain knowledge is used in several ways – to understand the learner's input, to aid diagnosis of the learner's state and to generate some appropriate reply. SOPHIE [Brown, Burton & Kler, 1982] provides a good example of the use of domain knowledge for input understanding. A *semantic grammar* [Burton & Brown, 1979] of domain concepts is used to parse the learner's statements in terms of semantic categories (quantities, variables, locations etc) rather than syntactic categories (noun phrase, verb phrase etc.).

Domain knowledge can be used in the diagnosis of the learner's state by relating the learner's behaviour to a domain expert's behaviour in the same circumstances. This is known as *differential modelling* and was used in the WEST system [Burton & Brown, 1982]. The diagnosis can then be used as an input to the communication strategy to inform the decision-making processes of the ILE. This process of comparing the domain knowledge with the student model is central to the ITS concept and is based on the notion that a teaching system must be knowledgeable about the domain being taught [Anderson, 1988].

Whenever an ILE introduces some new concept it is using domain knowledge in its responses. At its simplest this is just stating new facts, for example:

Q: Is it true that fog is a form of precipitation?

A: No, fog is an example of condensation.

[Brown, Burton & Zydel, 1973]

The formal domain model in ITSs has generally been regarded as a correct representation containing the knowledge to be communicated to the student. The importance of the domain model has allowed this view to permeate many of the knowledge-based activities of ILEs, input understanding, diagnosis, explanation etc. In order to support viewpoint reasoning in any of these tasks the domain model of an ILE must be expressed in a viewpoint-based manner. Previous ILEs have tended to avoid this conclusion by remaining in formal domains where these problems can be concealed in the noise from all the other problems in ILEs.

Although it is generally accepted, within the literature, that experts make use of multiple models of a domain, there have been few intelligent tutoring systems (ITSs) which address this issue.

[Sime & Leitch, 1992]

1.3.2 Learner Model

or who am I communicating with?

The learner model is the system's representation of what the learner believes about the domain based on the learner's behaviour [Dillenbourg & Self, 1990]. The process of inferring a learner model is called *diagnosis*. The fundamental difference between the learner model

and other knowledge in the ILE is that the learner model is more dynamic⁴ – it can change after each step in the interaction.

An *overlay model* [Brecht & Jones, 1988; Goldstein, 1982] represents the learner as missing some pieces of the domain model – the learner is implicitly diagnosed as being a subset of the expert. Alternatively, common student mistakes can be collected into a *bug library* [Burton, 1982].

This system diagnoses a student by finding bugs from the library that, when added to the expert model, yield a student model that fits the student's performance.

[VanLehn, 1988]

A further refinement is to create a library of more primitive operations (or bug parts) which can be used to generate bugs [Langley, Wogulis & Ohlsson, 1987]. These approaches rely on restricting possible diagnoses of the learner at the design stage rather than engaging in a clarifying dialogue at run-time. The individualised adaptive interactions are limited by the scope of the domain model or the system libraries – they adapt to the learner only as long as the learner remains within the scope envisaged by the designers. Given the inherent variability of human learners this is a significant restriction.

The approaches mentioned above are mainly behavioural approaches (bugs) to learner modelling rather than conceptual diagnoses (misconceptions). This is partly due to their procedural production system approaches. Misconceptions refer to the differences between the conceptual knowledge of the learner (as represented in the learner model) and that of the ILE.

The terminology employed in learner modelling reinforces the view that the ILE is always right and the learner is always wrong. A *misconception* may be the manifestation of an alternative interpretation of the domain. Columbus was thought to have a major misconception in trying to reach China by sailing west but in a viewpoint with a spherical earth it was a reasonable approach. If the diagnosis of the learner model is reliant on a domain model with a different interpretation then the ILE's

⁴ The domain model doesn't change unless the student produces a better solution than the system. Teaching knowledge doesn't change although the particular instantiation of a dialogue plan will follow changes in the learner model.

decisions will be misinformed – and communication failure is to be expected.

Any diagnosis that uses the domain model is implicitly assuming that the domain interpretation of the learner and the system are similar – they share a common viewpoint. The learner model is constructed by processing information from the domain model, the initial assumptions about communication and past learner behaviour. The assumptions may be implicit and distributed throughout the system, for example, all numbers are in base 10.

Learners frequently use several models to solve problems [Collins, 1985; Collins & Gentner, 1983; diSessa, 1986; Williams, Hollan & Stevens, 1983]. [Minsky, 1985] proposes that diagnosing an electrical motor requires at least two models: an electrical and a mechanical. So the learner model must be prepared to change viewpoints to accurately reflect the learner.

The learner model is the basis for all intelligent interaction in an ILE.

We regard it as axiomatic that any intelligent tutoring system needs a student model

[Self, 1988a]

Diagnosis that assumes that domain model and learner share the same viewpoint, or that a learner uses the same viewpoint at all times, is likely to generate faulty learner models. This implies that:

- learner modelling should support several viewpoints
- diagnostic procedures should be expressed in viewpoint-based terms
- domain models should be expressed in viewpoint-based terms

1.3.3 Communication Strategy/Pedagogical Knowledge

or how to decide what to do next ?

The communication strategy of an ILE determines the choice and sequence of actions it takes in order to achieve its goals. The inputs to this process are typically the current user action, the learner model, the domain model, previous action history etc.

The communication strategy can be expressed as a network of states, such as *Explore Competency* and *Repair Misconception* [Woolf & McDonald, 1984], or as a series of teaching operators with preconditions and expected effects [Peachey & McCalla, 1986]. The strategy may realise a

particular educational technique such as cognitive apprenticeship [Collins & Brown, 1988] or Socratic tutoring [Stevens, Collins & Goldin, 1982]. However it is generally acknowledged that a system must have several revisable plans in order to provide adaptive interaction [Ohlsson, 1986], e.g. the DOMINIE system [Elsom-Cook & Spensley, 1990].

The use of different communication strategies reflect different assessments of the learner relative to the system's goals. These assessments can only ever be as good as the learner model on which they are based, so any strategy is dependent on accurate learner modelling for its effectiveness. This dependence exhibits itself in the accessibility of the learner and domain models which will answer queries in terms of their knowledge representation language.

The main implication of this thesis for communication strategies is that the knowledge representation used in their input and decision taking should be in terms of viewpoints irrespective of the particular strategy represented.

1.3.4 Interface

or in what ways can I communicate?

The interface is where the learner perceives the output of the ILE.: ranging from simple text to multimedia systems using video and sound.

The interface also mediates the learner's communication with the ILE. Voice recognition is not yet a robust enough technology and so almost all input to the ILE is via keyboard and mouse. In addition natural language understanding is not yet advanced enough to allow learners to communicate freely in dialogue systems.

Despite (or perhaps because of) these restrictions the interface is so important that successful systems have been built that embody little intelligence but have supportive problem-solving environments, e.g. EPIC [Twidale, 1989], STEAMER [Hollan, Hutchins & Weitzman, 1984], ALGEBRALAND [Foss, 1987]. The interface can make different conceptual interpretations concrete, for example, STEAMER allows learners to view a graphic propulsion plant at different levels of granularity.

In a domain where an ILE can display different interpretations to the learner the interface becomes an important part of the communication strategy. In other, primarily textual, domains the interface is less of an active entity in the dialogue and more of a messenger between ILE and learner: its central role is not to corrupt the messages.

Where the presentation of the domain is non-textual the form of the interface has a significant influence on the interaction – interface choices are really pedagogical choices and so should be explicitly represented in the communication strategy (section 1.3.3).

1.3.5 Summary

Viewpoints affect all knowledge-based activities, which includes the whole of an ILE: learner modelling, domain modelling, communication strategy and interface design. However the importance of viewpoints is not evenly distributed between the components of the ILE.

The domain model of an ILE has to be represented in terms of viewpoints for many, if not most, domains. However it is precisely these domains which ILE research has failed to investigate. Agent models also need to be represented in terms of viewpoints both to model agents in the domain and to allow agents to have valid alternative interpretations of the domain.

The goal of interface design is to make the interface transparent [Burns & Capps, 1988]; to have no corrupting influence on the communication. The pedagogical strategy is dependent on other components to deliver information so that decisions can be made. Together these statements imply that knowledge, and hence viewpoints, are of maximum importance in the domain and agent modelling components of an ILE.

Although the display technology and the agent communication strategies may change the domain and agent modelling components will be required by virtually all ILEs. Consequently, work in these areas will have the widest range of future applications.

1.4 Chapter Summary

The majority of research on ILEs has been based on representing knowledge using a single interpretation. It has been recognised that this is an untenable position for many non-formal non-closed domains that ILEs have previously avoided. This is partially a result of the structure of ITS dialogues and the predominance of ITSs within the ILE field. It has also led to the adoption of a knowledge communication philosophy:

The 'knowledge communication' approach reflects a remarkably Platonic view of the nature of knowledge – that there is some objective knowledge 'out there', absolute and true, which it is the ITS designer's job to describe and communicate to

a learner. Epistemologists would concede this only for the purest of pure mathematics. ... teachers of the social sciences and humanities would blanch at the suggestion that they should merely 'communicate knowledge.'

[Self, 1988b]

There are two complementary approaches to dealing with this problem: redesigning the concept of an ILE and improving its reasoning abilities. The educational technique of peer tutoring provides an alternative conception of an ILE that may reduce the problems caused by differing viewpoints. Within the ILE viewpoint problems occur in all of the components although the domain and agent modelling sections are the most important.

Peer tutoring can be implemented in a learning by teaching system based on an artificial student – which should prove easier to build than an ITS. Although such a system has been proposed [Palthepeu, Greer & McCalla, 1991] only one system has been implemented [Michie, Paterson & Hayes-Michie, 1989].

The central themes of this thesis are therefore:

- learning by teaching is an alternative form of ILE that may have fewer viewpoint-related problems than an ITS
- all of the components of an ILE should be designed and implemented in terms of viewpoints

This thesis therefore adopts a research insight in its methodology – 'one problem space is not usually enough for hard problems' [Minsky, 1985]. When diagnosing the failure of a car the engine can be seen from two different viewpoints: a mechanical system and an electrical system. Working in one viewpoint at a time reduces the problem complexity and enables the problem solver to concentrate on specific aspects, e.g. electrical conductivity. The final solution may well involve integrating results from several problem spaces, e.g. the mechanical chassis of a car has electrical properties.

This thesis will deal with the viewpoint problem space, then the learning by teaching problem space and finally will attempt to bring the separate results together.

1.5 Thesis Structure

Chapters 2 and 3 review the use of viewpoints in the literature in both implemented systems and in less tangible fields. Chapter 4 examines learning by teaching starting with the educational theory of peer tutoring

and discusses the issues in building an ILE to act as an intelligent student; an Intelligent Student System (ISS).

Chapter 5 describes the DENISE (Demonstration Environment for an Intelligent Student in Economics) system and the experimental domain of economics. Chapter 6 describes the experimental considerations in evaluating DENISE as an ISS.

Chapter 7 describes and analyses the results of interactions with DENISE and Chapter 8 concludes the thesis with a synthesis of the research and proposes future work in the area.

Chapter 2

Implemented Viewpoints

2.1 Introduction

In order to properly discuss the application of viewpoints it is necessary to identify both implemented and non-implemented viewpoints. This chapter reviews the use of viewpoints in implemented knowledge-based systems. Chapter 3 contains an examination of viewpoints that have not been implemented or are from fields not associated with computer implementation.

As we are concerned specifically with the application of viewpoints to Intelligent Learning Environments the literature review in this chapter is divided into educational and non-educational systems. Section 2.2 reviews educational systems, section 2.3 other systems, such as knowledge representation languages, and section 2.4 compares the two groups.

As the terminology in this field is inconsistent italics will be used to signal terms⁵ specific to each system, e.g. *context, world, space, environment* etc.; from section 2.2.2 to section 2.3.4 (inclusive).

2.2 Implemented Viewpoints in Educational Systems

In the opening chapter viewpoints were identified as being relevant to all four of the conventional components of an ILE: domain model, learner model, interface and communication strategy. In this section we will extend this decomposition to cover all types of implemented educational systems.

2.2.1 Viewpoints in Domain Models

This section reviews the use of viewpoints in the domain models of implemented educational systems. In addition, the WHY system is included as the first study to recognise the importance of viewpoints; although the conclusions were not implemented they greatly influenced later work.

⁵ Viewpoints are a subset of the terms used in describing the systems; not all terms refer to viewpoints.

WHY

The WHY tutor is particularly interesting as in evaluating the original system the developers came to the conclusion that 'representing knowledge about physical processes requires multiple representational viewpoints'⁶ [Stevens, Collins & Goldin, 1982]. WHY performed simple dialogues to aid the student to acquire a model of the causes of rainfall and in doing so provided an eloquent demonstration of the limitations of a single-viewpoint approach.

The domain knowledge in WHY was originally represented as a number of hierarchical scripts. These scripts viewed causes of rainfall as a 'temporally-ordered linear sequence of events' [Stevens, Collins & Goldin, 1982]. This was discovered to be only a partial explanation of the domain and a 'functional viewpoint' was proposed to explain the relationships amongst object attributes. Other viewpoints were proposed to explain the presence of bug patterns in their studies of human reasoning: the energy viewpoint, the change-of-state viewpoint and spatial relationships.

The evaluation of the WHY system concluded that although the move to multiple viewpoints is complicated it is necessary for effective tutoring.

SOPHIE

One of the most significant projects in the ITS field was that of SOPHIE (I, II and III) in the domain of electronic troubleshooting. SOPHIE I [Brown & Burton, 1975; Brown, Burton & Bell, 1975] used four representations of knowledge to function as a reactive learning environment: a simulator, a world state (or database), several inference specialists and a module of qualitative knowledge about electronics.

The simulator generated the database which was acted on by the inference specialists. For example, one of these specialists is called the *Proposer*, it takes an observed circuit measurement and deduces a list of faults that could explain it. The Proposer does this using a set of procedures which encode relevant knowledge about electrical circuits. Thus the electronic knowledge in SOPHIE is distributed between a quantitative simulation and qualitative reasoners.

⁶ [Moyses, 1988] describes this as a 'pleasing confusion of terms' as they do not correspond to his definitions of representation and viewpoint.

This distinction was made clearer in SOPHIE III which was constructed basing 'its inference techniques on those that we observed experts and students using' [Brown, Burton & Kler, 1982]. The electronics expert consisted of three levels, quantitative voltages and currents, qualitative assertions and a behaviour tree of components and circuit modules. Communication between different reasoners is made through a 'common language of justifications and assumptions' [Brown, Burton & Kler, 1982].

SOPHIE III illustrated the integration of two alternative viewpoints in the domain of electronics, a low-level quantitative view and a qualitative view.

NEOMYCIN

NEOMYCIN [Clancey, 1983] is a reconfiguration of the rule-based expert system MYCIN so that it can function effectively as a knowledge base for the tutor GUIDON [Clancey, 1987].

The motivation behind NEOMYCIN was that the knowledge in MYCIN was compiled into the rules thus making it impossible to extract and utilise. The result was that students could not follow the explanations given by MYCIN. Although the diagnostic performance of MYCIN was excellent the knowledge was not organised in an appropriate manner for teaching. 'In order to make contact with the knowledge of the domain a level of *structural knowledge* is necessary' [Clancey, 1983].

The structural knowledge required grouped rules into semantic categories closer to human-like reasoning: causal relations, taxonomic relations and diagnostic strategies. [Wenger, 1987] pp277 describes these structural indices as 'orthogonal viewpoints'.

The importance of NEOMYCIN is that it demonstrated that performance in a domain is insufficient for educational applications. Although MYCIN was effective at problem-solving its knowledge was inappropriate for tutoring. The *structural knowledge* of NEOMYCIN was crucial for effective communication and was at least equally important as task-related success in the domain.

QUEST

The QUEST (Qualitative Understanding of Electrical System Trouble-shooting) system is a learning environment for locating faults in electrical circuits and is based on a progression of causal models [White &

Fredericksen, 1987]. The electrical expertise is 'captured by a small set of mental models that embody alternative, but coordinated, conceptualisations of system operation' [White & Fredericksen, 1990].

Progressions can vary along three dimensions: *perspective*, *order* and *degree of elaboration*. The *perspective* of a model 'refers to the nature of the model's reasoning in explaining a circuit's operation' [White & Fredericksen, 1990]. Three examples are given of different perspectives:

- high level functionality of the circuit (functional models)
- behaviour of the circuit components (behavioural models)
- micro-behaviour of the circuit (reductionistic physical models)

Different perspectives appear to have both different grain sizes and model types – although all models are qualitative and causal. It is not clear whether two functional models at different grain sizes qualify as different perspectives.

The *order* of the models refers to a subdivision in behavioural models - between zero-order models and first-order models. Zero-order models reason on the presence or absence of voltage, current etc. (is the light on?). First-order models reason on changes (qualitative derivatives) in voltage, current etc. (is there a voltage increase when we change the light's resistance?). These are distinct from quantitative models that return measures of actual voltages.

The *degree of elaboration* of a model 'is determined by the number of qualitative rules used in propagating the effects of changes in states of circuit components on the behaviour of other components' [White & Fredericksen, 1990]. Both the order and the degree of elaboration appear to refer to a notion of complexity – the form of rules and the number of rules, respectively.

The three model dimensions allow different model evolutions to be defined so learners can choose alternative model progressions according to their goals. To ensure learnability along a progression of models the models have to be *causally consistent* – that is there are no contradictions between models whatever their perspective, order or degree of elaboration.

QUEST is an important project as it demonstrates that even in domains that appear to be formally defined the issue of viewpoints is still relevant for domain knowledge to be used effectively in an ILE.

The VIPER (Viewpoint-Based Instruction for Prolog Error Recognition) system tutors the localisation of errors made by Prolog students in terms of three pre-defined *viewpoints* [Moyse, 1992].

A *viewpoint* is defined as consisting of three parts, see Figure 2.1: a model, a set of inference procedures (or operators) and a set of heuristics to specify the contexts and goals for applying the viewpoint (although VIPER itself does not use application heuristics). Viewpoints are not proposed as psychological structures but as ‘a description of the application of a mental model’ [Moyse, 1991].

VIPER embodies three viewpoints of a Prolog interpreter: a search space, a resolution process and a search strategy. These viewpoints are explicitly represented in the interface and govern knowledge representation throughout the system, e.g. ‘the [activity] history must contain all the information required for an analysis in terms of any one of the implemented viewpoints’ [Moyse, 1992].

The student can choose which of the three viewpoints to work in when attempting some problem although one user ‘maintained that there were only two viewpoints in his opinion, Resolution and Search’ [Moyse, 1992]. However the learner cannot remove any of the pre-defined viewpoints or define new viewpoints.

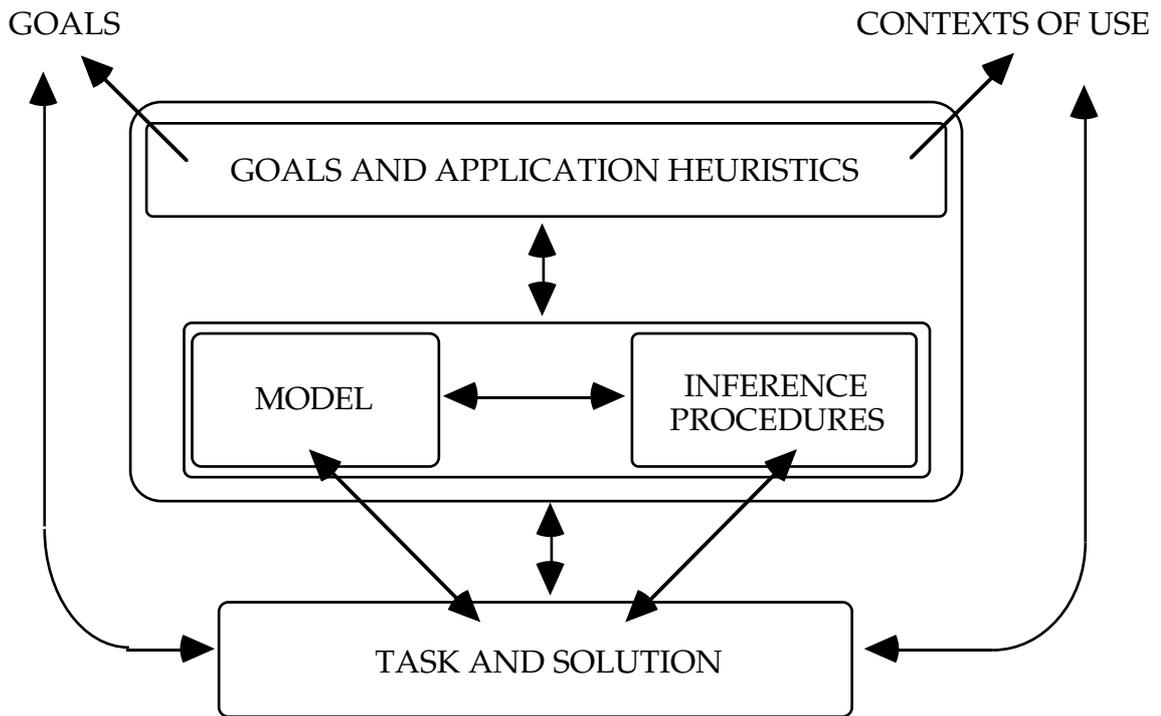


Figure 2.1 An outline structure of a viewpoint from [Moyse, 1990]

Viewpoints are present in all the components of VIPER illustrating their relevance to all aspects of an ILE, as noted in section 1.3. However VIPER illustrates both good and bad points about a viewpoint-based system. On the positive side VIPER shows that viewpoints can be implemented in a working system and they are useful in decomposing a domain. On the other hand, for pragmatic reasons, the viewpoints in VIPER are pre-defined (and 'hard-wired' into the interface) and static which limits any adaptation of the system to those circumstances predicted at design-time. VIPER does not contain a student model so adaptation does not in fact occur 'although [Moyse, 1990] indicates that suitable models could be implemented using well-researched techniques' [Moyse, 1992].

SCENT

SCENT [Greer & McCalla, 1989] is an advice-giving system to novice LISP programmers that uses granularity to recognise student programming strategies.

Granularity is represented through an *abstraction* hierarchy and an *aggregation* hierarchy. The abstraction hierarchy describes general-specific focus shifts whereas the aggregation hierarchy is based on part-whole relationships. For example, a path through the abstraction hierarchy could

be, <Lisp Program, Function Definition, Recursion, Cdr Recursion>, from most general to most specific. In the aggregation hierarchy Cdr Recursion consists of <Cond Well Formed, Null base case, Recursive cdr reduction case>.

However it is recognised that granularity is necessary but not sufficient: 'frequently an object can be characterised in a number of ways, even at a specific grain size' [Greer & McCalla, 1989]. To represent this each characterisation is called a '*K-cluster*'. K-clusters occur in the aggregation hierarchy and collect relevant predicates into relevance groups. They represent alternative sets of components that are equally valid ways of decomposing/generating their common ancestor object.

The explicit representation of granularity in SCENT enables it to simultaneously monitor a student's activity at several levels of granularity. So even if one action cannot be understood at one level it is likely it will be understood somewhere in the two hierarchies and so provide a sound basis for action.

SAMPLE

SAMPLE [Micarelli et al, 1991; Micarelli et al, 1992] is an ITS for teaching students about electrical circuits. SAMPLE is realised in KEE^{TM7} [Filman, 1988] which provides a context mechanism of *worlds* and a Truth Maintenance System (TMS).

A world is a representation of a set of related facts, i.e., a situation, a belief set, or a hypothetical state of a problem solver. In the world system the declarative knowledge sets up the *Background*. The Background is the set of facts that are true in every situation...

[Micarelli, et al., 1991]

Worlds are connected in an acyclic directed graph with child worlds produced by incremental modification of existing worlds. Child worlds inherit the facts contained in consistent ancestor worlds – inconsistent worlds cannot generate child worlds. Worlds can also be *merged* to allow the re-combination of results from a problem that has been split into sub-goals.

⁷ KEE (Knowledge Engineering Environment) is a trademark of IntelliCorp.

The world mechanism in SAMPLE appears to be entirely hidden within the system and is not observable by the learner. The worlds are not used in a directly educational manner but solely to improve the domain reasoning of SAMPLE .

Others

CIRCSIM-TUTOR [Khuwaja et al, 1992] is an ITS in the domain of the baroreceptor reflex of the human cardiovascular system. Experience with a single qualitative causal model in version 1 showed that this was not adequate for a significant minority of students. This led to version 2 including a three level causal model which reflected a categorisation of students into three groups depending on their reasoning ability. As in QUEST the model was causally consistent across all levels.

2.2.2 Viewpoints in Learner Models

When domains consist of viewpoint-oriented knowledge it is not surprising that modelling learners in those domains requires a similar viewpoint-based approach. However, learner modelling is considerably more difficult than domain modelling for several reasons:

- change – learner models must be dynamic whereas domain models are static. In addition human learning itself is non-monotonic, learners can forget skills they previously knew.
- bandwidth problem – learner modelling only has learner behaviour on a particular problem(s) as permitted by the interface as input compared with the variety of knowledge acquisition techniques [Wenger, 1987] pp388.
- restricted time – learner modelling decisions are made in an interactive environment with timing constraints which limit processing.
- behaviour discrimination/credit assignment – a learner modeller usually does not know whether a particular action is a true reflection of the learner's beliefs or a mistake due to misconceptions about the interface, cognitive overload [Wenger, 1987] pp386 - or even the learner just experimenting with the system [Twidale, 1991].
- domain boundary problem – a learner modeller cannot know whether some action is a misconception relative to its domain model or a manifestation of the learner using knowledge from outside its

domain coverage, i.e. alternative viewpoints. Any knowledge an expert uses is necessarily relevant – because they're an expert!

- learner inarticulation – learners are less likely to be able to articulate their beliefs, plans and goals than domain experts.

Consequently learner models are more likely to rely on simpler (non-viewpoint) techniques than domain models as solving these problems is considered difficult [Moyse, 1990] pp 2.

IMAGE

IMAGE [London, 1992] is the student modeller of the GUIDON2 ICAI system - the domain model is NEOMYCIN, see above.

IMAGE uses a *multiple anticipation* approach to plan recognition to model the learner. Common assumptions such as the closed-world assumption, user correctness, a cooperative user, no real-time constraint and a single unified plan are dropped and replaced by two other assumptions of *relevance* and *ease*.

Relevance: If there are several possible explanations of a student's action, provisionally assume the one that is most closely relevant to previous actions.

Ease: Of several possible explanations, assume the one that is simplest in terms of domain concepts.

[London, 1992]

Students are assumed to prefer cognitive economy – they will tend to maintain continuity in their behaviour and use simpler rather than complex explanations. Possible plans are constructed from partial plans and are ranked according to the two cognitive economy assumptions. However IMAGE also maintains plans other than the most likely and continually revises all plans in response to new input.

In order to restrict the plan recognition search space IMAGE uses explicit contexts (intentional, temporal, conceptual) and gradually relaxes them to expand its search in a controlled manner. The contexts group together situations of the same kind leaving three distinct types of viewpoints:

1. *Layers*: levels of abstraction, granularity, complexity or articulation
2. *Variations*: alternative possibilities; permutations with differing assumptions

3. *Angles*: specializing filters; partial semantics in a consistent body of beliefs

[London, 1991]

IMAGE attempts to advise students using viewpoints only in the third sense of complementary models or *angles*.

2.2.3 Viewpoints in the Communication Strategy

In section 1.3.3 it was noted that the communication strategy is reliant on the other components of an ILE to provide information to decide on the next action. However the interface between the communication module and the other modules will probably be jointly specified by an ILE's designer.

Within the communication module the role of viewpoints is unclear. One possible question is: do multiple teaching strategies (as in DOMINIE [Elsom-Cook & Spensley, 1990]) qualify as viewpoints? DOMINIE has several different styles including: cognitive apprenticeship, discovery learning, Socratic diagnosis and abstraction. The decision to use one style over another is based on a belief that it will be more effective with the learner. This decision is informed by the interaction history, student preferences and constraints between the styles (some styles are more suited to frequent assessment).

The lack of explicit knowledge in DOMINIE hinders the consideration of the styles as viewpoints although work is being done to rectify this:

To some extent this [style selection] extension of the system requires a more meaningful student model, but it also requires a set of styles which are specifically constructed from explicitly represented goals and beliefs.

[Elsom-Cook & Spensley, 1990]

Although the use of multiple teaching strategies has been recommended, the 'Principle of Versatile Output' [Ohlsson, 1986], it has not been applied in implemented systems [Elsom-Cook & Spensley, 1990]. A possible explanation (of this) is that viewpoints are of greater relevance to the domain and learner models of an ILE (see section 1.3.3) and that viewpoint-related reasoning is done before information is passed to the communication component. This means that discussion of viewpoints in the communication component is really only a discussion of the effects of maintaining viewpoints in the other parts of the ILE.

2.2.4 Viewpoints at the Interface

The interface is in the position of having the greatest influence on the usability of an ILE [Dillenbourg, 1991; Twidale, 1991]. It provides the user's only view of the system and however good the communication strategy it cannot compensate for the inability of an interface to effectively convey the system's intentions to the user.

STEAMER

STEAMER is an instructional system based on an interactive inspectable simulation of the steam propulsion unit of a large ship [Hollan, Hutchins & Weitzman, 1984]. The graphical models STEAMER uses approximate the models that experts use to reason about the propulsion system. An important aspect of this *conceptual fidelity* is that the student can 'view and manipulate the plant at a number of different hierarchical levels' [Hollan, Hutchins & Weitzman, 1984].

The views available in STEAMER vary from actual subsystems in the plant to abstract representations, for example, icons are used to represent rates of change - information not easily derived from traditional gauges.

Such a qualitative graphical interface can operate as a *continuous explanation* of the behaviour of the system being modelled by allowing a user to more directly apprehend the relationships that are typically described by experts.

[Hollan, Hutchins & Weitzman, 1984].

The large number of views (one hundred) required in STEAMER demonstrates that although complex devices can be made more comprehensible through multiple representations they may require considerable effort to encode.

BRIDGE

BRIDGE [Bonar & Cunningham, 1988] is a system that enables students to learn programming through the successive refinement of goals and plans into Pascal code. Further work on BRIDGE has led to the emphasis on representations that display a domain and how to link and move between them. The methodology adopted is that interface should be based on the domain expert's view of the domain.

Three kinds of expertise are delineated (concepts, representations and rules for referring to situations in the world) and it is the links between these kinds of expertise that are important. 'There must be a way for the student to see how actions in one representation influence a second representation' [Bonar, 1991]. These connections are called *yoking* between representations (of expertise), for example, moving from an informal natural language plan through an iconic plan to Pascal code.

Through the introduction of an iconic plan representation BRIDGE provides an intermediate viewpoint that provides a path from natural language to Pascal code. This demonstrates a further use of viewpoints – as linking mechanisms that provide representational stepping stones that segment the conceptual distance learners have to travel.

ANALYZER

Finally we note that a concern with viewpoints is not restricted to the AI and Education community – CAI software can also present students with multiple representations of a domain. ANALYZER [Yerushalmy, 1991] is a mathematics package that presents algebraic functions and graphs. The findings indicated that the software promoted learner inventiveness and enhanced understanding but that the connections between different representations were not implicitly acquired by the pupils. Considering other related CAI work:

The picture which unfolds from these and other studies is that multiple representation tools have the potential to adequately and successfully serve algebra students and help them to understand major concepts. Effective methods for doing this are yet to be developed, however.

[Yerushalmy, 1991]

2.3 Non-Educational Implemented Viewpoints

These are divided into three groups, knowledge representation systems, databases and others.

2.3.1 Knowledge Representation Systems

This section reviews implemented viewpoints in knowledge representation systems. Knowledge representation is almost synonymous with AI and the systems discussed come from widely differing areas although they all share a common trait - viewpoints.

Viewgen

Viewgen is a 'belief engine' built to demonstrate the first steps towards an artificial believer by embodying an algorithm for belief ascription [Ballim & Wilks, 1991].

Any system that is designed to engage in dialogue with other agents must reason about them and their beliefs [Ballim, Wilks & Barnden, 1991; Wilks & Ballim, 1987; Wilks & Bien, 1983] In Viewgen these requirements

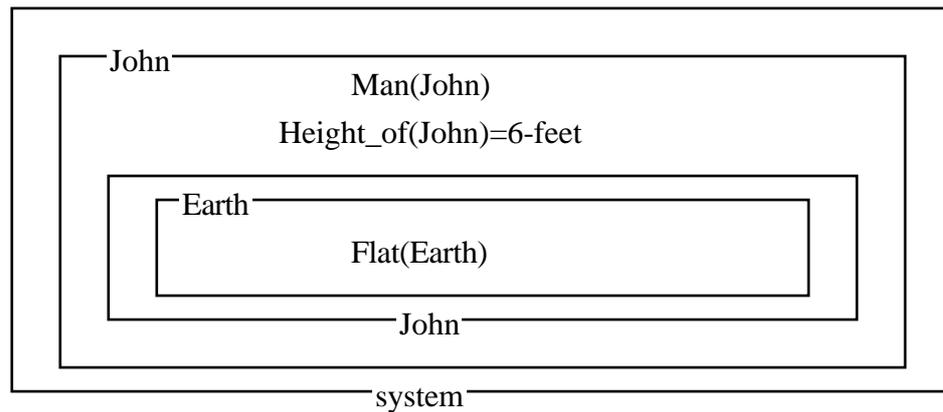


Figure 2.2 Beliefs about and of John from [Ballim, Wilks & Barnden, 1991]

are satisfied through the use of *environments* which partition the beliefs the system holds. Environments are groups of propositions and exist in two forms: *viewpoints*, representing a particular agent's point of view, and *topics*, containing beliefs relevant to a given subject [Ballim & Wilks, 1991]. Viewpoints are constructed according to a rule of belief ascription:

The default ascriptional rule is to assume that another person's view is the same as one's own *except where is explicit evidence to the contrary*.

[Ballim & Wilks, 1991]

Figure 2.2 shows a pictorial representation of beliefs about and of John [Ballim, Wilks & Barnden, 1991]. Boxes with labels at the bottom are viewpoints or believer environments. All beliefs lie within the system viewpoint. Viewpoints contain topic environments, e.g. 'John' and 'Earth', shown as boxes with labels at the top left. So the system believes that John is a man and that he is six feet tall. As the topic environment 'John' is a person it can contain beliefs held by that person as well as beliefs about John. So within the topic environment is the (nested) viewpoint 'John' which contains the topic environment 'Earth' and the belief that the earth is flat. In other words, the system has beliefs about a six foot tall man called John who believes the earth is flat.

Viewpoints are generated on demand, rather than pre-stored, by ascribing beliefs from one viewpoint to another using the default rule mentioned above. Viewgen is at present not integrated with a natural language interface, although this is a long term aim.

Spaceprobe

Spaceprobe is a computational knowledge representation system that embodies the theory of *Partitioned Representations* [Dinsmore, 1991].

In partitioned representations the information communicated in a discourse or acquired by other means is distributed in a principled way over a large set of *spaces*, each of which defines a local, or *parochial*, domain of reasoning.

[Dinsmore, 1991] pp 45

Partitioned Representations are intended to be both a system of mental representation and a means of understanding natural language. Processing is performed in *spaces* that are created in response to 'certain morphemes and syntactic constructions' [Dinsmore, 1991] pp119 such as: 'Warren believes', 'If Bush has a dog', 'In the film', 'In Spain', 'In 1935' etc. These cues are similar to the *space builders* of the *Mental Spaces* theory presented in [Fauconnier, 1985], see section 3.4.

Spaces are organised into a hierarchy and accessed via a *context* such as 'Warren believes' in a similar way to the viewpoints in *Viewgen* (see above). Reasoning is local to a space although the results in child spaces can be raised into parent spaces using the context: this is called *context climbing*. The context can be thought of as similar to an 'application heuristic' [Moyse, 1992] in that it determines when a space should be used. Spaceprobe also uses various activations for its spaces depending on how recently they have been used in discourse processing.

Partitioned Representations are basically a structured set of small knowledge bases with information distributed between the content and organisation of the spaces. The *parochial* reasoning used is the same in all spaces. Although the motivation behind Spaceprobe and Partitioned Representations seems to be linguistic they illustrate the power that can be derived from moving away from an unstructured knowledge base.

MULTILOG, OMEGA and IM2

MULTILOG is a logic programming language for knowledge representation based on *worlds* [Kaufmann & Grumbach, 1986]. Worlds consist of a set of clauses and an inference mechanism and can be linked together via inheritance relations.

MULTILOG extends the logic-based OMEGA language [Attardi & Simi, 1984] and partitioned semantic networks [Hendrix, 1979] by providing local inference mechanisms for each world. [Kaufmann & Grumbach, 1986] provide an example involving diagnosing a logic circuit which shows that MULTILOG is capable of hypothetical reasoning and of using different strategies (forward and backward chaining) in different worlds.

In OMEGA a *viewpoint* is defined as being a set of assumptions and as in MULTILog viewpoints can be related to other viewpoints, e.g. viewpoint *vp1* is a subset of viewpoint *vp2*. [Attardi & Simi, 1984] outline an example in which OMEGA solves the Wise Men Puzzle⁸ by assigning a single viewpoint to each wise man. In addition there is a viewpoint of the real world which acts as an oracle, or truth source⁹. [Attardi & Simi, 1984] claims that the OMEGA framework allows different situations at different times to be represented without resorting to temporal logic.

IM2 [Emde, 1987] is a knowledge representation system based on *worlds* which are similar to the *viewpoints* of OMEGA.

‘The most notable difference between worlds and [OMEGA] viewpoints is that information can be only added but not changed within viewpoints. If a description in OMEGA must be changed, then a new viewpoint has to be created.’

[Emde, 1987]

IM2 *worlds* are used to separate knowledge of different generality, declarative and procedural knowledge and also to allow the system to represent competing models of a domain and to run ‘contests’ between them. IM2 provides *world* inheritance in a similar way to OMEGA and MULTILog but also uses *evidence points* to represent uncertainty and so allows a *world* to contain contradictory information.

SNeBR

SNeBR (SNePS¹⁰ with Belief Revision) [Martins & Shapiro, 1988] is an implementation of MBR (Multiple Belief Reasoner) [Martins & Shapiro, 1983], an abstract belief revision system.

⁸ [Attardi & Simi, 1984] state the puzzle as: ‘A king wishing to know which of his men is the wisest puts a white hat on each of their heads, tells them that at least one hat is white, and asks the first to tell the colour of his hat. The man answers that he does not know. The second man gives the same answer to the same question. The third man answers that his hat is white. The puzzle is: how did the third man know his hat was white?’

⁹ [Ballim, 1991, pp 208] criticise this element of the OMEGA system as being unrealistic and present an alternative solution to the Wise Men Puzzle.

¹⁰ Semantic Network Processing System [Shapiro, 1979].

MBR works with *contexts*, sets of hypotheses, which determine *belief spaces*, the hypotheses plus all propositions derived exclusively from them. Each operation is performed relative to a *current context* and a *current belief space*. There are no restrictions on the content or subject of hypotheses in a *context* other than that they are not known to be inconsistent.

SNeBR integrates both forward and backward inference in a context although all contexts use the reasoning of the logic SWM¹¹ [Martins & Shapiro, 1988]. Belief revision is done interactively by the user by defining which hypotheses should be present in a context.

IDM

IDM (Integrated Diagnostic Model) [Fink & Lusth, 1987] is an expert system that combines two forms of knowledge, *experiential* and *fundamental*, in the domain of diagnosis and repair of electrical devices.

Experiential knowledge is the shallow empirical knowledge that an expert acquires over a period of time based on experience. ... Fundamental knowledge is the deeper model-oriented knowledge one usually acquires early on in training. It often comes from books and tends to be based on the structure and function of the device.

[Fink & Lusth, 1987]

The two knowledge bases have independent inference engines and their results are combined via an *executor*. The executor has a further knowledge base which provides the means of integrating the two experiential and fundamental knowledge bases. So IDM can function in situations where it is lacking device knowledge or experience.

Compositional Modeling

Compositional Modeling [Falkenhainer & Forbus, 1991] is an implemented technique for organising domain knowledge into *model fragments* to support large-scale multi-grain multi-perspective models.

A *model fragment* consists of four parts: structural configuration, relevance assumptions, operating conditions and relations imposed on parts of the model. These parts allow the re-usable model fragments to be

¹¹ After Shapiro, Wand and Martins.

composed into larger models to solve specific problems, e.g. only model fragments with appropriate operating conditions can be used in a situation.

The relevance assumptions include grain-size, ontology (e.g. contained stuff, energy flow, molecular collection), approximations and abstractions. [Falkenhainer & Forbus, 1991] describes compositional modeling in terms of mechanical devices but considers the algorithms to be domain independent. The project's ultimate aim is to produce a generic ITS and they implicitly consider the multi-grain multi-perspective problems to be endemic to all domains.

The essence of compositional modeling is selecting and combining elements of a large library of partial models in the most efficient and relevant way possible given the constraints imposed by the problem. The significance of the work is how much prominence is given to viewpoints (grain size and perspective) as a means of focusing computational effort.

Others

Cyc [Guha & Lenat, 1990] is an AI project to develop a system with common sense by representing a very large body of knowledge.

Cycl, the representation language of Cyc, allows a set of sentences to be declared a *microtheory* and have associated with it a description related to their scope and use. Microtheories allow Cyc to reason about grain size, multiple representations and, by restricting their context, simplify the axioms used.

KRL [Bobrow & Winograd, 1977] is a frame-based knowledge representation language that integrates procedural and declarative knowledge. One of the major intuitions that form the basis of KRL is:

A description [of a conceptual entity] must be able to represent partial knowledge about an entity and accommodate multiple descriptors which can describe the associated entity from different viewpoints.

[Bobrow & Winograd, 1977]

Other knowledge representation languages, such as KL-ONE (Brachman), ART (Williams), have provided similar capabilities for describing objects from different viewpoints 'but no particular approach for applying them within a domain or using them for appropriate explanation' [London, 1991] pp 42.

An alternative approach is that of representing *Graphs of Models* [Addanki, Cremonini & Penberthy, 1991] – models are treated as nodes and edges represent the assumptions that have to be changed in moving between the models. When a model conflicts with a problem the system switches to another model that is based on assumptions that tend to alleviate the conflict. Graphs of Models (or GoMs) have been implemented in physical domains where the typical assumptions may involve a frictionless system, rigidity or mass distribution. This approach highlights the problem of model choice – how to choose a new model when the present model is inadequate – and makes the relationships between the different models (viewpoints) particularly explicit.

2.3.2 Databases

Databases can provide users with different *views* of their contents:

A view may be a subset of the database or it may contain **virtual** data that is derived from the database files but is not explicitly stored.

[Elmasri & Navathe, 1989] pp9

The view facility allows users to restrict the information presented by the database so they can concentrate on specific subsets. In relational databases, e.g. SQL based, a view means a '*named derived virtual table*' [Date, 1990] pp187. Views are not based on physical data but are generated by a definition in terms of other tables. [Date, 1990] lists four advantages of views:

- logical data independence when restructuring the database
- allowing the same data to be seen differently by different users
- simplifying the user's perception
- automatic security is provided for *hidden*¹² data

Views are also being developed for object-oriented databases - to provide similar, and commonly understood, features found in relational systems, pp215 [Kim, 1990; Mariani, 1992]. There are many examples of relational systems with views, see [Date, 1990], but the Botany Knowledge Base has been realised with a more sophisticated concept of views than those outlined above.

Botany Knowledge Base

¹² Hidden data is data not visible through a given view.

View Type	Context
Class-dependent	class hierarchy
Structural	physical or temporal structure of an object or process
Functional	the role of an object in a process
Modulatory	how one object or process affects another
Attributional	the properties of a concept
Comparative	comparing or contrasting two concepts

Table 2.1 View Types identified in [Acker, et al., 1991]

The Botany Knowledge Base (BKB) is a representation of *fundamental* domain knowledge containing multiple highly integrated *viewpoints* [Acker et al, 1991]. Fundamental knowledge is characterised as that covered in introductory college, broadly surveys the domain and 'is not reducible to a small number of principles or axioms' [Acker, et al., 1991].

The principle behind the BKB is that representing all possible viewpoints is not practical for a large knowledge base and instead the required viewpoint is dynamically generated according to a number of *view types*.

A view type defines the relations and properties of a concept that are relevant when considering the concept from a viewpoint belonging to that view type. It specifies *necessary relations*, which must be included, and *permissible relations*, which may be included but are not required.

[Acker, et al., 1991]

It is suggested that a 'small number' of these view types is all that is required to categorise all of the viewpoints within physical domains. The view types proposed so far are shown in Table 2.1. A *viewpoint* is specified by applying a view type to a concept of interest (the main topic) and a reference concept (to be related to the main topic). A functional view type applied to a concept of interest, e.g. pollen, and a reference concept, e.g. plant reproduction, generates the viewpoint 'the functional role of pollen in plant reproduction' [Acker, et al., 1991].

A question to the BKB determines the concept of interest whereas the view type and the reference concept are inferred from the student

model, history, teaching plan etc. The view type is effectively a selector from the knowledge base that restricts the system to a particular set of relations to be used in answering a question. The concept of interest and the reference concept select on the elements rather than the relations.

2.3.4 Others

The production of explanations for users is an example of an area where communication is crucial and has consequently seen some work on viewpoints. BLAH [Weiner, 1980] is an example of the simplest – maintaining a *system* and a *user* view. Elements in the user view were deleted from explanations on the assumption that the user did not want explanations cluttered with obvious or repetitive information.

XPLAIN [Swartout, 1983] attached viewpoints to steps in the methods for generating explanations. When a step was used in creating an explanation the viewpoint is compared against a filter list and the result determined whether the step would be explained. The filter list could be changed to respond to the user's behaviour although 'the problem of deciding which viewpoint to present to a particular user remains open' [Swartout, 1983].

2.4 A Comparison of Implemented Viewpoints

The common link between all the systems discussed above is that a monolithic domain model has been rejected as a possible solution in favour of some notion of viewpoints. Although these solutions are varied there are some common elements – the most important being shown in Table 2.2.

There is a clear distinction between those systems which have separate inference mechanisms for each space (e.g. MULTILOG, VIPER) and those which attempt a global solution (e.g. Viewgen, OMEGA). In addition systems vary in the creation of new spaces, from user defined (e.g. SNeBR) to those triggered by linguistic cues (e.g. Spaceprobe); whereas other systems are restricted to the viewpoints defined at design/compile time (e.g. STEAMER, VIPER) Also some systems have spaces that are necessarily associated with agents (e.g. Viewgen) whereas others allow any beliefs in a space (e.g. SNeBR, MULTILOG). A further difference is evident in the relationships between viewpoints, are these made explicit (e.g. Graphs of Models) in the system, are they present in a hierarchical system (e.g. Viewgen, Spaceprobe) or are they implicit in the user's mind (e.g. VIPER)?

Table 2.2 shows a summary of the major differences in the approaches of most of the systems reviewed in this chapter. The systems are compared along several dimensions – possible answers are listed in parentheses.

Global/local inference: is inference the same in all spaces or can different inference mechanisms work in different spaces (global, local)

Agent Spaces: are spaces necessarily associated with agents? (Yes, P(Permitted), No)

Dynamic Space Creation: can new spaces be created dynamically? (Yes, Library, No)

Space/User Transparency: are the different spaces made available to users? (Yes, No)

Space Relationships: can spaces inherit from other spaces? (Yes, No)

As this comparison is based on reports of systems there exists some ambiguity as to some of the attributes for some of the systems. In Table 2.2 this ambiguity is represented by question marks.

System/ Language	Global/ Local Inference	Agent Spaces	Dynamic Space Creation	Space/User Transparency	Space Relationships (inheritance?)
WHY	-	-	-	-	-
SOPHIE	Global	No	No	No	No
QUEST	Global	No	No	Yes?	Yes?
NEOMYCIN	Global?	No	No	No	?
VIPER	Local	No	No	Yes	No
CIRCSIM	Global	No	No	?	Yes
SAMPLE	Global	No	Yes	No	Yes
SCENT	Global	No	Lib?	No	Yes
IMAGE	Global	No	Lib	No	No
Viewgen	Global	Yes	Yes	No	Yes
Spaceprobe	Global	Yes	Yes	No	Yes
MULTILOG	Local	P	Yes	No	Yes
OMEGA	Global	P	Yes	No	Yes
IM2	Global	P	Yes?	?	?
SNeBR	Global	P	Yes	Yes	Yes
IDM	Global	No	No?	No	No
Cycl	Global	Yes	Yes	?	Yes
Comp. Modlg.	Local	No	Lib	?	Yes
KL-ONE	Global	Yes	Yes	?	Yes
GoMs	Local	No	No	?	No
BKB	Global	No	Yes	Yes?	No

Table 2.2 A Comparison of Implemented Viewpoints

In addition to computer implementation there is a considerable body of viewpoint-relevant work in other fields – it is reviewed in Chapter 3.

2.5 Chapter Summary

The chapter has reviewed the examples of viewpoint implementation in a variety of fields. The most important differences between the implemented systems have been isolated to:

- user transparency: is the user aware of the viewpoints?
- dynamic spaces: can viewpoints be added/deleted during execution?
- inference: is the inference method space-local or space-global?
- agent spaces: are agents' beliefs necessarily associated with viewpoints?

- model relationships: are the relationships between models explicit?

Chapter 3

Non-Implemented Viewpoints

3.1 Introduction

The previous chapter examined viewpoints in implemented systems. In this chapter the review is extended to examine viewpoints that have not been realised in a computer implementation. This includes work in the fields of mental models, linguistics and ILEs.

As before, *italics* indicate the terms used by the authors in referring to their own work, from section 3.2 to section 3.5 (inclusive).

3.2 Intelligent Learning Environments

This section reviews ILE work on viewpoints which has not been implemented. The intangibility of this research arises from several sources. Some work is in the form of a review, other authors provide suggestions or speculations and others simply fail to indicate whether any implementation has been attempted.

3.2.1 Wenger: Interpretative Contexts

[Wenger, 1987] considers *viewpoints* to be necessary 'to place misconceptions in a broader conceptual context.' A *viewpoint* is defined as an *interpretative context* determined by its *kernel* and its *scope*.

The kernel of a viewpoint consists of a variable number of keys, which together define the interpretative context referred to as a viewpoint. Keys can be prior decisions, correct or incorrect beliefs, analogies, or assumptions that either explicitly belong to the model or must be inferred as underlying it. The scope of a viewpoint delineates its foreseeable area of relevance. Note that the exact scope of a viewpoint is rarely precisely defined a priori since viewpoints are likely to have obscure ramifications. "Foreseeable implies that the scope of a viewpoint is some minimal area of probable applicability." In sum, a viewpoint is an interpretative context whose kernel contains critical keys to the proper understanding of entities within its scope.

[Wenger, 1987] pp 355

The interpretative context has three levels of application:

- situation specific: problems can be viewed and solved in different ways.
- domain specific: domains can be viewed in different ways.
- background: the assumptions of a learner's worldview.

In addition [Wenger, 1987] briefly outlines composite viewpoints formed by combining the kernels of two viewpoints. He also states that viewpoints can be *compatible* or *competing* but gives no examples or justification.

The usefulness of viewpoints in an instructional setting is decomposed into:

- optimisation of diagnosis: understanding the student's input
- optimisation of didactic steps: acting in line with the student's viewpoint
- instructional tools: introducing new viewpoints on a subject
- instructional targets: using the attainment of a new viewpoint as a goal

Wenger gives no indications of the possible links between the three levels of viewpoint application – or indeed how/if reasoning might differ between the levels. The reason for this is that the levels lack clear boundaries – quite where a situation becomes a domain or a domain becomes a worldview is unclear. The differences between seeing ‘a problem in multiple ways’, using ‘different primitives’ and different ‘worldviews’ are linguistic rather than semantic. The division is equivalent to picking out three levels of an inheritance hierarchy in preference to the others. Wenger's proto-typology of viewpoints merely serves to illustrate that viewpoints are relevant at all levels of knowledge.

The remarks on composite viewpoints are too vague to be useful. The uses of viewpoints, above, are orthogonal to those noted in [Moyses & Elsom-Cook, 1992] (see 2.2.1, pg 24) but address the same underlying issues.

3.2.2 Self: Belief Sets

[Self, 1992] regards a viewpoint as a set of beliefs held by some agent.

A belief is a dispositional state. A belief (and hence a viewpoint) is held by some agent. ... dispositional is intended to indicate that possession of the belief disposes the agent to behave in an certain manner but does not guarantee it.

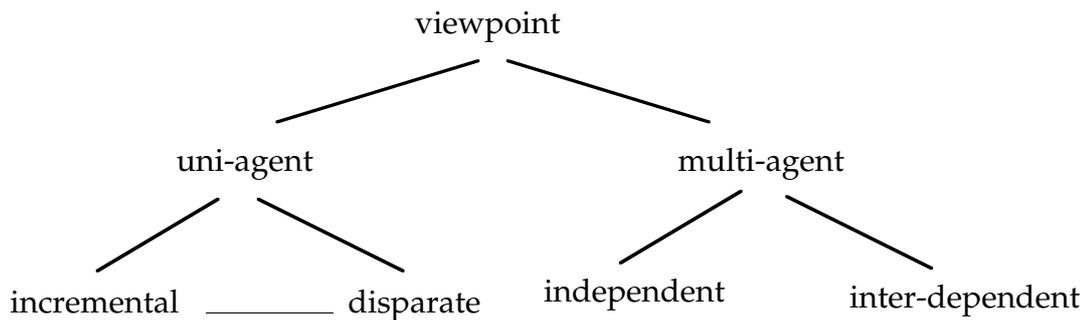


Figure 3.1 Classification of Viewpoints from [Self, 1992]

[Self, 1992]

Applying a viewpoint yields a view. The discussion of viewpoints takes place in the context of the classification shown in Figure 3.1. The major distinction is between viewpoints held by one agent and those held by two or more agents.

Viewpoints held by one agent vary from each other by differing amounts, those close to each are called incremental viewpoints, those significantly different, disparate. These distinctions represent fuzzy areas on a viewpoint spectrum rather than discrete alternatives. An example of incremental viewpoints are the successive states of solving a state-space problem such as the missionaries and cannibals. In contrast disparate viewpoints involve changes in perspective that hopefully make the problem easier to solve. Different representations of the problem can have significant influences on the ease of solution, e.g. the mutilated chessboard.

The main complication that arises in multi-agent scenarios is that agents can maintain viewpoints about other agents and consequently viewpoints about the viewpoints of other agents, etc. Blackboard systems [Hayes-Roth, 1985] are offered as an example of independent viewpoints where each knowledge source acts without reference to the others in order to solve the problem. In contrast distributed problem-solving also occurs when agents do reason about each other, thus creating inter-dependent viewpoints (e.g. see *Viewgen* in section 2.3.1).

3.2.3 Other ILE Work

Using resolution in different viewpoints has been discussed in the context of diagnosing a student's misconceptions in an ITS [Costa,

Duchenoey & Kodratoff, 1988]. It is based on the idea that misunderstandings can be caused by 'teacher and learner working with different *contexts*.' The system must therefore have partitions in its knowledge base and establish which partition, or context, the interaction takes place in *and* which the student thinks it takes place in. Costa's example shows that by locating the correct *context*, through resolution, the student can be more accurately diagnosed.

[Finch, 1988] regards *viewpoints* as *filters* over a domain model which are selected to match an explanation of a domain concept to the student's current pedagogical needs. This is similar to the *overlay* learner model (section 1.3.2). Multi-expert knowledge acquisition is recommended as a method of obtaining the different viewpoints to be drawn over the domain.

3.3 Mental Models

Mental models research is fundamentally concerned with understanding human knowledge about the world

[Stevens & Gentner, 1983]

Mental models are the constructs that the human mind builds about the physical world; the model that you use to predict the behaviour of a calculator, moving objects and rainfall.

When attempting to instruct students in understanding electrical circuits tutors often use conceptual models of other systems, or analogies, to convey aspects of the domain. The rationale for this is that the students understand the principles of the analogical conceptual model and can apply some of these to the target system.

Electricity

Two such analogical conceptual models of electricity are that of 'flowing waters' and 'teeming crowds' [Gentner & Gentner, 1983]. This study of the effects of these analogies on subjects' reasoning about electricity showed that their predictions varied according to the structure of the analogical model they held. Their conclusions show the importance of the selection of the conceptual models presented (directly, or through analogies) to students.

In another study of mental models of electric circuits [Collins, 1985] identified six 'incorrect' models in addition to 'a more or less correct model of the way a battery circuit works.' In addition subjects were

observed to have different views on the components of the circuit, 'four different views of the battery' and 'three prevailing views of the light bulb.' Collins argues that the two or three global views of systems usually observed are the product of different views of components at a finer grain level. These lower level views produce many combinations but 'frequently two or three combinations predominate giving rise to the global mental models' [Collins, 1985].

Physical Systems

The use of multiple models has been observed in subjects reasoning about a heat exchanger system.

One thing that surprised us about his use of multiple models was the extent to which he seemed to switch between models in the midst of a single chain of reasoning. ... We consider the use of multiple models to be one of the crucial features of human reasoning.

[Williams, Hollan & Stevens, 1983]

The subject successfully integrated two incomplete system models in order to solve a problem. This use of multiple models has been observed with subjects shifting between the various models when bugs or ambiguities arose [deKleer & Brown, 1983]. [Collins, 1985] comments on two mental models of how a thermostat works, the feedback and the valve views, and at a finer-grain level suggests the existence of many component models as for electricity.

The benefits of using a conceptual model in an instructional situation have been demonstrated when a group with a device model was compared with a control group [Kieras & Bovair, 1984]. Although both subject groups were taught the same procedures the group with a model of the device performed significantly better. These benefits are reliant on the models being perceived in the same way by all the participants in the interaction.

Mental models research shows that humans construct personal models about the world which are at variance with reality. Some of the 'misconception' models are generated (or possibly propagated) by many subjects. The 'remediation' of these models is dependent on the tutor (human or computer) recognising the existence of the models and acting appropriately.

3.4 Linguistics

This thesis started by saying that communication between agents results in misunderstandings due to the different beliefs of the dialogue participants [Black, Turner & Bower, 1979; Grice, 1968]. Linguistics has addressed the problem when the communication takes place in natural language.

Mental Spaces

One account of how to construct meaning in natural language is given in the *Mental Spaces* theory of [Fauconnier, 1985]. Mental Spaces are typically established through linguistic cues that separate the subsequent beliefs from current beliefs. Any expression that generates a new space, or refers back to previously created space, is called a *space-builder*. Examples of space builders include:

- in Len's picture
- in John's mind
- in 1929
- from her point of view
- possibly
- theoretically
- Max believes
- Gertrude claims

[Fauconnier, 1985] claims that in order to understand natural language humans continually create, modify, delete and move between these mental spaces. Many of the space-builders are analogous to belief logics and systems like *Viewgen* (section 2.3.1) in using agent-specific viewpoints. However the space structure is augmented by *connectors* which link together items in different spaces. For example, a drama *connector* links a *trigger* of Henry V in a play with a *target* of Henry V in the real world [Fauconnier, 1985] pp19. This approach means that although spaces are structured and hierarchical (in that Len's beliefs about Bill are 'deeper' in the hierarchy than Len's beliefs in general) the majority of inter-space relationships are between elements. These links provide a heuristic for locating a new space:

- plausible new spaces are those with elements with connectors to recently processed elements in the current space

Relevance

[Sperber & Wilson, 1986] contend that human cognition revolves around one property – *relevance*. They claim that ostensive communication, in addition to the message itself, ‘communicates the presumption of its own optimal relevance’ [Sperber & Wilson, 1986] pp158. That is, the message should contain something useful and it is the most relevant of the possible messages the communicator could have chosen.

The implication of this is that given a multiple-space belief representation a plausible metric for selecting spaces is how relevant the belief is. Relevance is defined in terms of *contextual effects* and *contextual effects* are the consequences of adding new information to a context (or space), specifically:

- erasing assumptions from the context [space]
- modifying the strength of assumptions in the context [space]
- deriving implications in the context [space]

Trivial effects such as just adding new information without linking it to existing information do not count as *contextual effects*. This theory also suggests a plausible space selection heuristic:

- select a space where the contextual effects are greatest

A similar heuristic is indeed one of the two methods employed in the IMAGE system, see section 2.2.2.

3.5 Artificial Intelligence

A consequence of approaching problems from different points of view is that one view’s beliefs are likely to be inconsistent with those in another view. [Fagin & Halpern, 1988] have formalized this into a logic of *local reasoning* that permits agents to hold inconsistent beliefs without incoherent situations.

One reason that ‘people hold inconsistent beliefs is that beliefs tend to come in non-interacting clusters’ [Fagin & Halpern, 1988]. As each frame of mind can contain different beliefs about the same objects this amounts to allowing different viewpoints about some set of objects. This logic-based approach is similar to the SNePS system described in section 2.3.1. A formulation in terms of non-monotonic logic is described in [Kumata & Atsumi, 1988].

[Self, 1992] highlights three other logic-based ideas: implicit and explicit belief [Levesque, 1984], awareness [Fagin & Halpern, 1988] and

resource bounds. An agent's implicit beliefs include all the consequences of its explicit beliefs. Awareness allows agents to reason with just those beliefs that they are 'aware' of, i.e., those they consider to be relevant. The limited resources available to an agent may also restrict its reasoning, for example to achieve real-time constraints on response times.

3.6 Synthesis

The two preceding Chapters have shown that the informal term viewpoints covers a wide variety of interpretations and domains. The review has covered the main areas of viewpoint-related but a comprehensive review of *all* relevant topics is unrealistic within the scope of this research. [London, 1991] and [Moyse, 1990] contain reviews of viewpoints and they both conclude that a restricted examination of the topic is the only practicable approach. Moyse restricts the VIPER system to pre-defined viewpoints and London only considers viewpoints in his sense of *angles* (see section 2.2.2) in the IMAGE system.

The two complementary senses of viewpoints described in section 1.1 come together in the mechanism the system uses to represent the viewpoints of learners. In ITSs the general problem of student modelling is made considerably harder by the possibility of both the system's domain model and the learner's conception of the domain reflecting alternative viewpoints.

Placing the computer in the role of a learner simplifies the problems but does not remove the need for the system to represent learner's viewpoints on the domain. Given the large number of approaches to this task outlined in the Chapters 2 and 3 some conclusions need to be abstracted from the literature to proceed with the development of the proposed Intelligent Student System.

Table 2.2 listed the important elements of the implemented systems considered in Chapter 2. These were:

- user space transparency: are the viewpoints visible at the interface?
- dynamic space creation: are the viewpoints dynamic?
- space relationship: how are the viewpoints related to each other?
- agent spaces: are spaces associated with agents?
- global/local inference: do spaces have separate inference mechanisms?

The form of the proposed ISS necessitates that the viewpoints and their contents should be dynamic – otherwise the system will be unable to represent revision of users’ beliefs during the interaction. As the ISS is designed to represent a single learner’s beliefs there is only one agent to be modelled and so agent spaces are not required. The locality of inference and the relationships between viewpoints are determined by the choice of domain for the system (see section 5.2).

The emphasis on communication in section 1.1 implies that viewpoints should be visible at the interface – restricting the viewpoints to internal models only increases the probability of misunderstandings. This may seem obvious but many systems appear to treat viewpoints as an aide to reasoning and not as an aide to communication (see Table 2.2).

A corollary of *visible viewpoints* is that they should be explicit elements of the dialogue and consequently have unique identifiers – *names*. Viewpoints cannot be manipulated or discussed by the user without explicit identifiers.

In summary, the desire to explore the learner’s viewpoint in a domain with inherent viewpoints¹³ implies that the viewpoints themselves should become explicit and manipulable by the learner. These ideas are implemented in section 5.3.1.

¹³ A domain where a dialogue can be reasonably expected to include explicit viewpoints, e.g. economics, politics, psychology, religion, art etc.

Chapter 4

Intelligent Student Systems

4.1 Introduction

The concept of an educational system that behaves as a student, rather than a teacher, was outlined in Chapter 1. This Chapter considers these Intelligent Student Systems (ISSs) in more detail.

ISSs have several potential uses :-

1) **to support learning through being taught by a student.** As a teacher the human learner is forced to explain themselves and make their implicit reasoning explicit.

Explanation will bring out possible flaws in arguments, since through attempts to justify the reasoning, the person must examine previously unconsidered (and possibly unconsidered) justifications (VanLehn, 1985). If an explanation is flawed, then the process of having constructed the explanation will characterise the cause of the flawed explanation either as an error in logic (or argument) or as a wrong assumption.

[Palthepeu, Greer & McCalla, 1991]

2) **as a learning aid for trainee teachers** – VanLehn in [Sandberg, Barnard & van-der-Hulst, 1992]. The training of teachers involves a combination of theory and practice. Currently the practice takes place in real classrooms with real students. Although this provides real-world experience for the teacher it can be harmful for the students who run the risk of sub-standard teaching. Also there is no guarantee that the experience gained by the teacher is representative of possible student behaviour. A computerised version of teacher training could be tailored to prepare the novice teacher for a representative variety of situations with a consequent reduction in the interference in students' education [Berliner, 1985]. This would be beneficial for trainee teachers, current students and future students (whose teachers would be better trained).

For example, [Corte, Verschaffel & Schrooten, 1991] reports on a system which tests the diagnostic skills of student-teachers through identifying procedural bugs in arithmetic skills. The system is restricted to bug identification though and there is no element of remediation (or learner improvement) as the bugs are not generated from any form of learner model. On a larger scale STDM (Simulation

on Teacher Decision Making) [Shelley & Sibert, 1991] allows student-teachers to practice on a simulated class of learners. The student-teachers 'received research-based feedback on the probable effects of these decisions' [Shelley & Sibert, 1991] from the simulation. However the pupil simulation (and the student-teacher's decision-making) was based on variables like test results, attendance, health, family background and psychological reports rather than beliefs about a specific domain.

An ISS could also be used in the training of tutors for a peer tutoring project (see section 4.2.1).

3) as an aid for ITS designers. The design of an ITS implicitly includes a great deal of information about its potential users – learners. An ISS could be used in an analogous manner to a programmable user model:

A PUM [programmable user model] is a constrained cognitive architecture that can be programmed (e.g. by an interface designer) to simulate an hypothetical user performing some range of tasks with a proposed interface. A PUM therefore acts as an analytical model of a computer user, cast in a form in which the interface designer has to "program" certain aspects of the user's intended behaviour.

[Young, Green & Simon, 1989]

Although PUMs were conceived of as a tool for the evaluation of interface designs they can be applied to ITSs.¹⁴ A PUM is a subset of an ISS as it is only intended to interact with an ITS over a small range of behaviour. So the partial construction of an ISS by an ITS designer can highlight the implicit assumptions made about learners in an ITS.

4) as an alternative assessment metric for learners. Assessment of learners in most procedural domains is straightforward, e.g. after being taught a subtraction algorithm learners are given subtraction problems to perform. In declarative domains task performance is not so clearly identified. Conventional methods include multiple-choice, data-response and essay-style questions. An alternative is to measure how

¹⁴ [Young, 1989] considers that, in general, 'artificial users' are impossible to build. However constraining the 'users' to be simply artificial *learners* renders the approach tractable. The user of an interface may well be undertaking a complex loosely-bounded activity in an unspecified domain (e.g. writing a thesis, designing a car, etc.) whereas a learner learning a new concept is only involved in one human-like activity in a tightly constrained domain.

successfully a learner can communicate their knowledge about the domain to an ISS. How successfully the ISS acquires knowledge, or can answer questions about the domain, can be used as the basis of a measure of how well the learner has comprehended the domain.

Human-human communicability is unreliable as a metric as those being taught cannot be standardised between learners – whereas an ISS can be consistent. The teaching of knowledge provides a concrete interactive task for learners compared with, for instance, the difficulties of predicting the beliefs of somebody who will read an essay at some future time.

5) **as an evaluation tool for ITSs.** The evaluation of ITSs is ‘costly, frustrating, and time-consuming’ [Littman & Soloway, 1988]. The automation of the evaluation process is therefore a desirable goal for ILE research. ITSs are evaluated with respect to their effects on human learners (who can be difficult to obtain) and a significant part of automating their evaluation is to provide a population of artificial learners (ISSs). Using ISSs ITSs could undergo many more trials before moving into a real-world teaching situation. If an ITS fails to teach a cooperative ISS then it will probably have considerable difficulty with human subjects. As such an ISS can provide a lower bound for the effectiveness of an ITS.

6) **as a method of evaluating different methods of instruction.** Alternative theories of instruction are compared with respect to the resultant complexity of the ISSs behaviour as it learns. [Ohlsson, 1992; Ohlsson, Ernst & Rees, 1992] uses this ‘method of teachable simulation models’ implemented as a production system with constraints to compare the complexity of the *regrouping* and *augmenting* subtraction algorithms.

This Chapter will concentrate on the first of these uses: a one-to-one learning by teaching system – a computerised analogue for the educational practice of peer tutoring (although this could be said to include an element of self-assessment as well). The objective of such a system is not to introduce the learner to new material but to encourage greater understanding of previously acquired domain knowledge. A learning by teaching ISS is thus a complement to an ITS rather than a substitute.

First, peer tutoring and tutorials are examined as they provide the educational rationale for a learning by teaching ISS. Other related systems, such as automated knowledge acquisition tools and machine learning programs, are compared with ISSs. The knowledge requirements of an ISS are discussed and an architecture outlined.

4.2 Related Educational Research

The utility of the ISS approach can be considered from two perspectives: the analogy with peer tutoring and its relationship to other conventional educational practices.

4.2.1 Peer Tutoring

Peer Tutoring is 'the system of instruction in which learners help each other and learn by teaching' [Goodlad & Hirst, 1989] pp13. Collaborative behaviour has always been an important part of learning but peer tutoring represents a structured approach with definite educational objectives.

The simplest peer tutoring scenario consists of two subjects: a tutor and a tutee – the tutor being more able in the domain than the tutee. The tutee is usually given some tasks which are at the limit of their ability and when they encounter difficulties the tutor provides help. At this point the tutors take on the role of a teacher in understanding the tutee's difficulties and providing a tailored explanation. As tutees are not usually in continued simultaneous difficulty it is common for tutors to be allocated a small group of tutees although most interaction is usually one-to-one.

The basic idea is that tutees benefit through increased teacher-like personal attention and tutors benefit through experiencing the cognitive demands of teaching. Studies of peer tutoring schemes indicate that both tutors and tutees make cognitive gains. [Goodlad & Hirst, 1989] pp 61 classify the benefits to tutors as:

- development of their sense of personal adequacy
- finding a meaningful use of the subject-matter for their studies
- reinforcing their knowledge of fundamentals
- in the adult role, and with status of teacher, the experience of being part of a productive society
- developing insight into the teaching/learning process and cooperating better with their own teachers

The studies show that tutors can learn by teaching, though the learning is not a simple acquisition of new facts. Tutors learn by restructuring their existing knowledge via experiencing the alternative cognitive demands required by the role of a teacher rather than that of student. A peer tutor is unlikely to acquire new knowledge of the domain in question but may well gain new insight into previously studied topics.

Although benefits to the tutees (individualised instruction, more teaching, companionship and better response to peers than teachers

[Goodlad & Hirst, 1989]) are used as justifications for classroom peer tutoring schemes, as we are considering systems to aid human learners, the tutee benefits are not relevant¹⁵.

Tutors on peer tutoring schemes report positively on reinforcing subject knowledge, practice in communication of ideas and gaining insight into other perceptions of the domain. For example, Table 4.1 shows some questionnaire results from the 'Pimlico Connection'¹⁶ peer tutoring scheme [Goodlad & Hirst, 1989] pp104. Evaluation of this scheme's benefits to tutors were confined to qualitative mechanisms such as interviews and questionnaires.

Studies of the academic achievements of ex-tutors have not been undertaken although some questionnaire results suggest that the experience improves the perception of teaching as a career [Goodlad & Hirst, 1989] pp 108. The schemes do not work well for all participants and some tutors do not benefit as much as others. However, a general result is that significant (not necessarily a majority) numbers of participants do benefit from the experience (see section 'Benefits to Tutors' on page 57) [Goodlad & Hirst, 1989] pp 61.

Studies in the matching of tutors and tutees tend to concentrate on the age and gender of the participants, the structure of the scheme and whether the tutors received any pre-scheme training. Although tutor differences have been compared with respect to tutee benefits there appear to be no guidelines in the literature as to which tutee characteristics are beneficial to tutors beyond broad statements such as 'an age/experience gap of three years seems quite adequate' [Goodlad & Hirst, 1989] pp 85.

Peer tutoring results are robust and wide-ranging; they have been repeated with subjects of different ages, genders, domains, educational levels, cultures, social backgrounds and mental and physical abilities

¹⁵ In the same way that any knowledge gained by ITSs is not used as a justification for adopting their use. The knowledge an ITSs acquires about a learner may (or may not) be lost at the end of a session – it doesn't affect a judgement about how effective the session was in aiding the learner.

¹⁶ A large peer tutoring scheme using students from Imperial College of Science, Technology and Medicine, University of London, as tutors in local secondary schools. The scheme began in 1975 and has used tutors in science, mathematics and craft design technology.

Percentage of students indicating they had benefited from being a tutor by:

	Greatly	Somewh at	Not at all	Not sure
Reinforcing your knowledge of some aspect of your subject?	2	38	56	4
Getting practice in the simple communication of scientific ideas?	55	41	2	1
Gaining insight into how other people perceive your subject?	35	52	10	3
Increasing your self-confidence?	15	59	19	7

Table 4.1 Selected questionnaire responses from the 'Pimlico Connection' peer tutoring scheme from [Goodlad & Hirst, 1989]

[Goodlad & Hirst, 1989; Goodlad & Hirst, 1990; Saunders, 1992]. There are studies which fail to replicate these common findings (e.g. [Kennedy, 1990]) which suggest that the full range of necessary factors for successful peer tutoring have not been identified.

Related to the peer tutoring studies are the informal reports of teachers who find that teaching students is effective in improving their own understanding of the subject [Berliner, 1989; Dillenbourg, 1991; Michie, Paterson & Hayes-Michie, 1989; Palthepe, Greer & McCalla, 1991]. A central mechanism is that teachers find they have to be able to explain a concept to themselves as a pre-requisite to explaining it to their students¹⁷ [Palthepe, Greer & McCalla, 1991]. Good self-explanation has been correlated with good student understanding [Chi et al, 1989; Ferguson-Hessler & Jong, 1990; Pirollo & Bielaczyc, 1989]. Indeed sometimes it is the act of explanation itself rather than any attribute of the other agent(s) that is important:

¹⁷ However the teacher's self-explanations are likely to involve a greater depth of knowledge than the explanations given to students.

Put it another way. Sometimes I've been stuck with a problem, and I've gone into a colleague's room and I've said "Can you spare me a minute?", right. And I've stopped halfway through what I've wanted to say because in exposing the argument I've understood where I've gone wrong. All he's done by way of help is to sit there. And he hasn't understood what I've been trying to say to him because I haven't finished what I was saying.

Economics lecturer¹⁸, interview 1, (lines 1147-1155)

There is, however, a difference between peer tutoring schemes and teachers' reports in that peer tutors do not attempt to teach an integrated course module but rather to opportunistically repair the tutee's local failures on some task.

A third evidential source comes from experiments in social cognition. One study of paired peers engaged in problem-solving showed that:

the initially more able members of each pair made more progress than children of comparable initial competence who worked alone.

[Light & Glachan, 1985]

This outcome was not universal amongst the subjects but, together with similar studies [Glachan & Light, 1982; Mugny & Doise, 1978], shows that these effects can be replicated under controlled experimental conditions as well as in the more informal situations found in peer tutoring schemes. Unfortunately, social cognition research concentrates on the tutees and equally matched pairs, with relatively little work investigating the effects on tutors [Rogoff, 1990]. The role of the less able peer in these experiments is to act as a Piagetian source of cognitive and conceptual conflict [Hewson & Hewson, 1984; Nussbaum & Novick, 1982].

These reports indicate that there may be cognitive benefits for learners in teaching a computer about the domain in which they are studying. These benefits include an increased understanding of the learning process, appreciation of the extent of their domain coverage, increased reflection, reinforcement of existing knowledge and making

¹⁸ As part of the examination of the domain of economics (see section 5.2) an economics lecturer was interviewed several times. Some interviews were informal and unrecorded; two interviews were recorded and transcribed. In interview 1 the lecturer described this episode which is relevant to this separate argument.

their implicit reasoning explicit. These effects should be independent of whether the learners are being initially taught by an ITS or a conventional human teacher.

Benefits to Tutors

Tutees on peer tutoring projects do not always behave in a manner conducive to the learning of the tutors. When tutors report what they 'liked least' about their experiences the most frequent answers were 'inattentiveness and indiscipline among pupils and not being fully used by teachers', also 'tutors could feel under-used or confused if lessons were not designed to make use of them' [Goodlad, 1985]. These drawbacks to peer tutoring are exactly the type of problems that a computerised alternative could be expected to address.

Peer tutoring schemes are often cited as providing affective, as well as cognitive, gains for their participants, e.g. [Horan et al, 1974; Yogeve & Ronen, 1982]. These include improvements in altruism, self-esteem, attitudes to teachers and motivation [Fresko & Chen, 1989; Goodlad & Hirst, 1989].

There is clearly a link between learners' attitudes and their achievements and the relationship is probably simultaneous (a positive feedback loop). The problems of accurately assessing affective changes [Goodlad & Hirst, 1989] pp76 makes it difficult to isolate whether cognitive gains are dependent on the social context of peer tutoring or are largely a consequence of the teaching activities. The possibility of removing the human social aspect (section 4.2.2) by replacing the tutee with an ISS should enable the cognitive effects of peer tutoring to be more accurately assessed.

Some preliminary evidence to suggest that the cognitive effects are related to the teaching activities, and independent of a human tutee, comes from experiments run on an algebra learning by teaching system [Michie, Paterson & Hayes-Michie, 1989]. Subjects taught (by example) a system centred on the inductive machine learning algorithm ID3 [Quinlan, 1986] in the domain of simple linear equations. Two versions of the system were used – in one the learning component was disabled. The subjects that used the learning version recorded the larger pre-test to post-test gain. A control group and a group using a commercial maths-education package performed less well than the learning by teaching subjects.

The conjectured motivational improvements of the [Michie, Paterson & Hayes-Michie, 1989] subjects, as measured by questionnaire responses, were not observed. This result is consistent with the intuitive assumption that affective gains are less likely from human-computer interactions than human-human ones. It is probable that ISSs will have to rely solely on cognitive outcomes to justify their use. However any conclusions drawn from just a single study should be treated cautiously. It is still the case that the [Michie, Paterson & Hayes-Michie, 1989] results are the *only* reported instance of a learning by teaching system in operation. Whether the outcomes are generalisable, for example across age-ranges or to declarative domains, remains an open question.

Research Methodology in Peer Tutoring

The evaluation method for the [Michie, Paterson & Hayes-Michie, 1989] experiment is typical of human-human peer tutoring schemes; pre and post-tests together with questionnaires. The interactions themselves are treated as black-boxes with few reports of actual tutee-tutor communications. Qualitative research is restricted to questionnaires and post-session interviews. Possible reasons for this include: the large scale of studies¹⁹, administrative and organisational restrictions, intrusion into and contamination of the tutor-tutee interactions.

This approach has led to the inappropriate application of psychological theories to peer tutoring:

Loose general references to complete theoretical frameworks within psychology (e.g. Gestalt theory) are unlikely to aid the production of a cohesive body of information or assist the discovery of cause and effect relationships concerning educational phenomena. This is because, in general, such frameworks can produce theoretical mechanisms that account for both the presence and the absence of the phenomenon studied.

[Kennedy, 1990]

We suggest that this imprecision in applying psychology to peer tutoring is a consequence of relying on gross variables such as gender, domain, age etc and pre/post tests/interviews instead of concentrating on

¹⁹ For example, [Kennedy, 1990] reports on a study using 108 pupils resulting in 17,280 minutes of interaction (without including a further 37 pupils who missed some of the sessions).

analysing the detail of dialogues as has been done in Socratic tutoring [Collins, 1977; Collins & Stevens, 1991]. This could also account for the lack of guidelines as to which tutee characteristics are best for which tutors.

Consider the following hypothesis about peer tutoring:

- the demands of providing a time-limited response to a tutee prevents the tutor from fully appreciating, say, the ramifications of a self-explanation provoked by a tutee's question. Thus, ironically, the source of the self-explanation could prevent the full consequences of the resultant belief revision from being made explicit: a self-limiting benefit.

Hypotheses such as this require the kind of low level dialogue-based research which is absent from the peer tutoring literature and could be made easier using an ISS. In this case an ISS is a good candidate for removing the time factor as it can simply wait for the tutor to respond and is unlikely to present the same demand for attention as a human pupil. Additionally the recording of self-explanation may be inhibited by the presence of a tutee.

A computerised approach is well suited to faithfully recording dialogues as they happen, without overt intrusion into the interaction, rather than relying on post-session recollection. ISSs have the potential to inform peer tutoring research by eliminating many experimental variables (e.g. gender, age, culture) to produce a 'purer' form of learning by teaching experience. This is in sympathy with the call for a more rigorous methodological approach to peer tutoring in [Kennedy, 1990].

4.2.2 Learner Beliefs in Tutorial Groups

The previous section gave some reasons why an ISS could be expected to successfully provide a tool to examine learners' beliefs. A related question is: how is this done in conventional education?

Although learners' knowledge may be contradicted in lectures and reflected in essays the *interactive* nature of the tutorial is the activity designed to identify misconceptions on a regular basis. The 'distinctive potential of small group work is learning which is based on the expression, exploration and modification of ideas' [Rudduck, 1978].

As a means of elucidating learner beliefs tutorials are limited by several organisational and psychological factors.

Tutors often complain that they have difficulty getting students to speak in tutorials and students for their part may regard an invitation to participate as “being picked on”. ... Other common reasons for students’ reluctance to participate in a tutorial are that they don’t know the ground rules, they are afraid of exposing themselves in public, or simply that they sense that the tutor really wants to do all the talking.

[Habeshaw, Habeshaw & Gibbs, 1984]

They come in here, and you ask someone something in the tutorial and they say, “Why are you picking on me”... they don’t articulate it, but that’s what they’re thinking. “Why have you asked me? I haven’t done anything!”

Economics lecturer, interview 1, (lines 1116-1120)

In analysing recordings of small group work [Rudduck, 1978] divides the problems and concerns of students in tutorials into four categories:

Making contributions:

- anticipating the end of other speakers’ contributions
- balancing listening and preparing to contribute
- the discussion moving faster than the student’s thoughts (*leading to*)
- contributions away from the focus of group discussion

Understanding conventions:

- uncertainty of rôles
- how far to go in acknowledging uncertainty or ignorance?

Knowing enough to contribute:

- uncertainty about the agenda
- anxiety that they do not want to be ‘shown up’ in front of their peers
- inadequacy of knowledge compared with the tutor

Assessment:

- is the tutorial being used as informal assessment?
- how far any comments made may circulate outside the tutorial

Some of these problems remain in an ISS interaction (e.g. possibility of actions indirectly affecting formal assessment) but the change from a social to a private context removes most of them. An ISS interaction is still social, in the sense that it is not introspective, but peer-pressure, turn-taking and discussion speed are based on human group attributes and need not generate problems in using an ISS. Although any system can be designed to replicate undesirable effects, it is taken as definitional that an ISS should not, for example, fail to allow a learner a turn in a dialogue or broadcast a learner’s misconception to a public network.

The problems listed above all inhibit the effective discussion of an individual learner’s beliefs about a topic during a tutorial session. The

continued prevalence of the tutorial in higher education largely therefore reflects economic considerations rather than educational objectives. The clear superiority of one-to-one human interaction [Bloom, 1984] is accompanied by greatly increased costs.

The dysfunctional aspects of tutorials described above show that, as well as the potential benefits from the *supply* of an ISS (section 4.2.1), there exists an unsatisfied *demand* for such a tool within existing educational structures.

4.2.3 Socratic Tutoring

The learning by teaching approach has some similarity to Socratic tutoring (or, more generally, inquiry teaching [Collins & Stevens, 1991]) in that ‘the central notion is to force the student to reason for himself’ [Collins, 1977]. However there are several significant differences between these two paradigms.

In the Socratic method the student learns three kinds of things: (1) specific information about a variety of cases; (2) the causal dependencies or principles that underlie these cases; and (3) a variety of reasoning skills. These include such abilities as forming hypotheses, testing hypotheses, distinguishing between necessary and sufficient conditions, making uncertain predictions, determining the reliability or limitation of these predictions, And asking the right questions when there is not enough information to make a prediction.

[Collins, 1977]

Socratic tutoring relies on the tutor selecting a path through a known domain to force a student to confront weaknesses in their knowledge. Rules identified during studies of tutorial interactions illustrate the dependence on the tutor’s domain model, for example:

Rule 16: Ask for a prediction about an unknown cause

If

a student has identified all the primary factors that affect the dependent variable

then

pick a case that is either hypothetical or unlikely to be known and ask the student to predict the likely value of the variable for that case.

[Collins, 1977]

1	T:	Where in North America do you think rice might be grown? [Ask about a known case]
2	S:	Louisiana.
3	T:	Why there? [Ask for any factors]
4	S:	Places where there is a lot of water. I think rice requires the ability to selectively flood fields.
5	T:	Ok. Do you think there's a lot of rice in, say, Washington and Oregon? [Pick a counterexample for an insufficient factor]
6	S:	Aha, I don't think so.
7	T:	Why? [Ask for any factors]
8	S:	There's a lot of water up there too, but there's two reasons. First the climate isn't conducive, and second, I don't think the land is flat enough. You've got to have flat land so you can flood a lot of it, unless you terrace it.
9	T:	What about Japan? [Pick a counterexample for an unnecessary factor]
10	S:	Yeah, well they have this elaborate technology I suppose for terracing land so they can flood it selectively even though it's tilted overall. (section omitted)
11	S:	Do you think they might grow rice in Florida? [Ask for a prediction about an unknown case]

T = Tutor, S = Student

Table 4.2 Fragments of a dialogue on growing grain from [Collins, 1977]

In applying this rule the tutor must know the primary factors of the dependent variable (domain model) and also have information about the likely knowledge of the student (student model). This is also shown in transcripts of Socratic dialogues, as in Table 4.2. Although some tutor questions, lines 3 and 7, are domain independent others, lines 5 and 9, require a complete domain model. Lines 1 and 11 show the use of a student model by the tutor.

Socratic dialogues rely on the tutor's expertise, whereas learning by teaching activities proceed with a relatively ignorant tutee. The contrast between the knowledge requirements of these human-human interactions is clear and there is no reason why it should not apply to their computerised analogues. In addition to placing different demands (and roles) on the learner, Socratic and learning by teaching approaches require different computational architectures.

4.3 Related Systems

There are two main types of non-ILE systems which are relevant in considering the components of an ISS – those that perform knowledge acquisition and machine learning.

Automated Knowledge Acquisition Systems

[Marcus, 1988] characterises automated knowledge acquisition tools as:

tools that can elicit relevant domain knowledge from experts; maintain that knowledge in a form which makes it accessible for analysis, review or modification; and use the knowledge to perform a specific task.

ISSs are similar to automated knowledge acquisition systems [Marcus, 1988] in that they engage in a dialogue with a human user who has superior knowledge about a domain. However, the difference in users could hardly be more extreme; a learner with incomplete and probably inconsistent beliefs and an experienced domain expert. Knowledge acquisition systems have no interest in the educational effects they have on their users (who, as experts, are probably not in need of such effects).

Some knowledge acquisition tools restrict interaction to contexts in which experts find it easier to perform. [Eshelman, 1988] reports that diagnostic experts often have trouble in deciding whether some evidence supports a hypothesis directly or by making alternative hypotheses less likely. Consequently the MOLE_{KA} tool [Eshelman, 1988] is designed to avoid direct questioning of experts and to infer this information from other sources. This behaviour is antithetical to that of an ISS; it should concentrate on areas where the user has difficulty rather than avoiding them. If topics of inconsistency were avoided then learners would come away from a learning by teaching session with a distorted view of their understanding of the domain.

In addition, knowledge acquisition systems are designed to produce expert systems that will function after the end of the knowledge acquisition session. The effects of an ISS are *constrained* to occur *during*

the interaction; there is no independent system to be constructed or future task to be performed²⁰.

The main aspect of knowledge acquisition systems that can be usefully applied in an ISS are the control strategies [Kahn, Nowlan & McDermott, 1985]. For example, [Palthepeu, Greer & McCalla, 1991] suggest *find contradictions, check completeness* and *confirm existing knowledge* as candidate heuristics for guiding a learning by teaching interaction. These issues are discussed further in section 6.2 in the design of an experimental dialogue.

Machine Learning

Machine learning is clearly relevant in trying to construct artificial students, as their real-life counterparts do appear to learn (at least some of the time). The important question is how should an ISS learn? [Michie, Paterson & Hayes-Michie, 1989] report on using an inductive learning algorithm in which the learners provided examples of solutions to equations from which the system induced a general rule. Subjects using the learning version showed a greater improvement than those using a non-learning version (section 4.2.1). However, the hypothetical system described in [Palthepeu, Greer & McCalla, 1991] appears to learn only in the sense of adding new relations to its knowledge base, i.e. rote learning. ISSs are different from machine learning systems in that learning is not an end in itself. Only learning that supports the generation of a useful question (that causes the learner to think) or maintains the integrity of the scenario is necessary. The interactive nature of an ISS means that if machine learning mechanisms are to be used to guide the dialogue the algorithm should be one that permits questions to be asked of the tutor, e.g. [Gasarch & Smith, 1992].

It is an open question as to whether complex machine learning is necessary or whether the style of reasoning in knowledge acquisition systems is sufficient to produce the desired effects in learners. The alternative of using a series of pre-defined simulations [Chan & Baskin, 1990] will restrict the learning of the ISS and requires prior knowledge of *any* model *any* learner may choose to teach to the system: this knowledge is clearly unavailable.

²⁰ Although it is possible that an ISS may provide learners with hardcopy session transcripts or model contents to remind them of the strengths/weaknesses of their domain knowledge.

Other Systems

Instructable systems [MacDonald, 1991] are also designed for knowledgeable users who act as teachers. Instructable systems allow users to automate repetitive tasks by learning them from a human user. As with some automated knowledge acquisition tools the system is designed to minimise cognitive effort on the part of a user who is not intended to learn about the domain during the interaction. Also the user is expected to fully understand the domain (or be able to perform the task) prior to using the system. These restrictions and a concentration on example-based tasks (e.g. guiding robot arm painters) mean that instructable systems, while superficially similar, are a distinct class of systems from ISSs.

4.4 Intelligent Student Systems

The previous section outlined systems which have some similarity to an ISS but which have not been influenced by educational considerations.

The evidence from classroom activities suggests that an ISS could provide an additional type of human-computer educational interaction with cognitive benefits for learners. The analogies with human educational practice suggest two different types of activity involving an ISS:

- a teaching activity where the learner acts as a tutor to the ISS by introducing new material as suggested in [Palthepe, Greer & McCalla, 1991].
- a task-centred activity²¹ where the ISS is monitored and aided by the learner as it attempts some task in the domain. This is closer to the style of interaction envisaged for a two-agent teacher-less LCS [Chan & Baskin, 1990].

These two activities can of course be concatenated – the learner tutors the ISS and then monitors and updates the system as it tests its knowledge on a domain task. However, there is no requirement for the two activities to be linked – it may be more effective for the learner to

²¹ This is different to the *instructable systems* [MacDonald, 1991] approach where the system performs the task independently, and is not monitored, thus freeing the human to do other tasks.

monitor models generated by other students or pre-specified 'buggy'/alternate-viewpoint models

An analogy can be drawn between these ISS activities and conventional programming. The teaching activity resembles program implementation and the monitoring activity is similar to program debugging and maintenance. However here there is no programming language syntax but a set of concepts and relations that enable learners to express their knowledge about the domain.

4.4.1 The Components of an ISS

A sample abstract architecture of a system to perform learning by teaching has been outlined in [Palthepeu, Greer & McCalla, 1991] and includes typical ILE components: student model, dialogue control, pedagogical module etc. The existent [Michie, Paterson & Hayes-Michie, 1989] system consists of an inductive machine learning algorithm, relevant decision factors (attributes) and the next solution step (classes). The binary attributes included *are there any bracketed terms?*, *does the equation have a common factor?* and *are there like terms on opposite sides?* Classes included *multiply out bracketed terms*, *combine like terms* and *divide by the coefficient of the unknown*. The algorithm learns the relationship between the attributes (what to look for) and the classes (what to do next) from the examples provided by the user.

The Domain Model

A major difference between ISSs and ITSs is that, as with real students, there is no requirement for a pre-defined domain model. The system is taught about the domain by the human learner at run-time rather than by a knowledge engineer at compile-time. Although there is no requirement for a pre-defined domain model there is no prohibition of one either. An ISS could have access to a domain model to make strategic decisions yet conceal it from the learner – the computer would be an expert posing as a novice. As [Palthepeu, Greer & McCalla, 1991] point out this destroys the domain independence of an ISS and requires careful design so that the expert knowledge does not become apparent to the learner. If an ISS demonstrated that it already knew the material it would destroy the integrity of the learning by teaching scenario and could confuse and/or de-motivate the learner.

Furthermore, a domain model will commit the ISS to a particular set of viewpoints which may, or may not, intersect with the viewpoints of the users. If the learner views the material through radically different

viewpoint(s) then the pre-defined domain model may be of little use in guiding the interaction.

An ISS with a domain model, an intelligent system taking the part of an ignorant student, can be regarded as the computational equivalent of using *confederates* in psychological experiments.

confederates: in an experimental situation the aides of the experimenter who pose as subjects but whose behaviour is rehearsed prior to the experiment. The real subjects are sometimes termed *naive subjects*.

[Goldenson, 1984]

The Tabula Rasa Assumption

Diametrically opposed to the inclusion of a domain model is the *tabula rasa* assumption [Palthepeu, Greer & McCalla, 1991]; the system is empty of domain knowledge, all of which is provided by the learner.

The *tabula rasa* assumption of a system without a domain model implies that all of the system's domain knowledge originates from the learner. This omission of domain knowledge acquisition means that ISSs should be simpler to develop than ITSs with a lower operating complexity.

The *tabula rasa* assumption is misleading as it conceals the need for a mutually understandable language in which to conduct the interaction. There is clearly a minimum level of domain knowledge that the ISS and the learner must share. In the analogy with programming this knowledge is that contained in the syntax of the programming language. This *conceptual syntax* imposes constraints on the relationships between domain objects; for example, in economics it makes sense to talk about the flow of redundancies from the employed to unemployed but not about the flow of inflation²². At the lowest level of granularity the conceptual syntax may be negotiable but in most situations an ISS would be expected to understand the basic types of relationships in the domain. These restrictions are separate from the content of models constructed by the ISS; the relationship between flows and stocks is independent of whether an economic model is monetarist, Keynesian or Marxist. Different subsets of the conceptual syntax would be required to deal with radically different

²² Flows (vacancies and redundancies) move between stocks (employed and unemployed) whereas inflation is the proportionate rate of increase in the price level.

viewpoints on the domain [Moyse, 1992]. In addition there must be a communication language that allows the user to express agreement and disagreement about concepts expressed in the conceptual syntax.

An ISS can be domain-independent in the sense of not possessing domain models but, unless the learner can tutor the conceptual syntax, will not be entirely devoid of domain knowledge. So the *tabula rasa* assumption must be applied to a system which includes a conceptual syntax.

So the minimum component list for an ISS is therefore: a dynamic domain model (or learnt model), a learning strategy to control the dialogue and update the model, a conceptual syntax and a communication language at the interface. More complex ISSs have an option to include full domain models and modelling of tutoring strategies used by learners.

In terms of the [Michie, Paterson & Hayes-Michie, 1989] system the conceptual syntax equates to the classes and attributes that delimit the behaviour of the inductive learning algorithm. The proposed [Palthepeu, Greer & McCalla, 1991] system disguises its conceptual syntax by not including it in the architecture description, but the hypothesised dialogue reveals it through the assumed mutual language of kinds, types and rules.

4.4.2 ISS Architecture

One of the distinguishing features of ILEs is the student model; 'the component that represents the student's current state of knowledge' [VanLehn, 1988]. The [Michie, Paterson & Hayes-Michie, 1989] system did not include a student model. Although [Palthepeu, Greer & McCalla, 1991] include a student model in their system description they question its utility. In the analogy with peer tutoring however, a student model seems perverse: a tutee generally can't model the knowledge of their tutor. The state of the student's knowledge is reflected in the model built up by the ISS and so a separate student model would be redundant.

The interface between the control strategy and the learnt model of an ISS can be considered to consist of two parts: actions that maintain the model and those which assess its contents. Figure 4.1 shows a schematic diagram of the theoretical architecture of an ISS with an optional domain model. The components are:

- *learnt model*:²³ the model the system has acquired from the learner merged with the 'seeded' initial model (if present).
- *conceptual syntax*: the restrictions on the contents of the learnt model.
- *dialogue strategy*: control of the system's behaviour, including when to update the learnt model. Achieved via conventional programming constructs: variables, conditional tests, loops, etc.
- *model maintenance functions*: that modify the learnt model; such as adding and deleting beliefs acquired from the learner.
- *model access functions*: the mechanisms with which the dialogue strategy tests the state of the learnt model and the domain model (if present). These are read-only functions which leave the learnt model unaltered.
- *interface functions*: communication with the user (dialogs, menus, prompts, graphics etc).
- *domain model*: (optional) a model of expert domain knowledge in the domain in which the system is being used.

The outline input-output cycle is:

- 1) learner provides an input
- 2) input checked against the current conceptual syntax
- 3) dialogue (or learning) strategy takes the checked input and, accessing the learnt model (and optionally the domain model), determines the next action, including, for example, a combination of:
 - modifying the learnt model
 - modifying the conceptual syntax
 - asking the learner a question (which could be confirmatory, exploratory, open or closed) or some other interface action

²³ [Palthepe, 1991] call this component a Domain Knowledge Base.

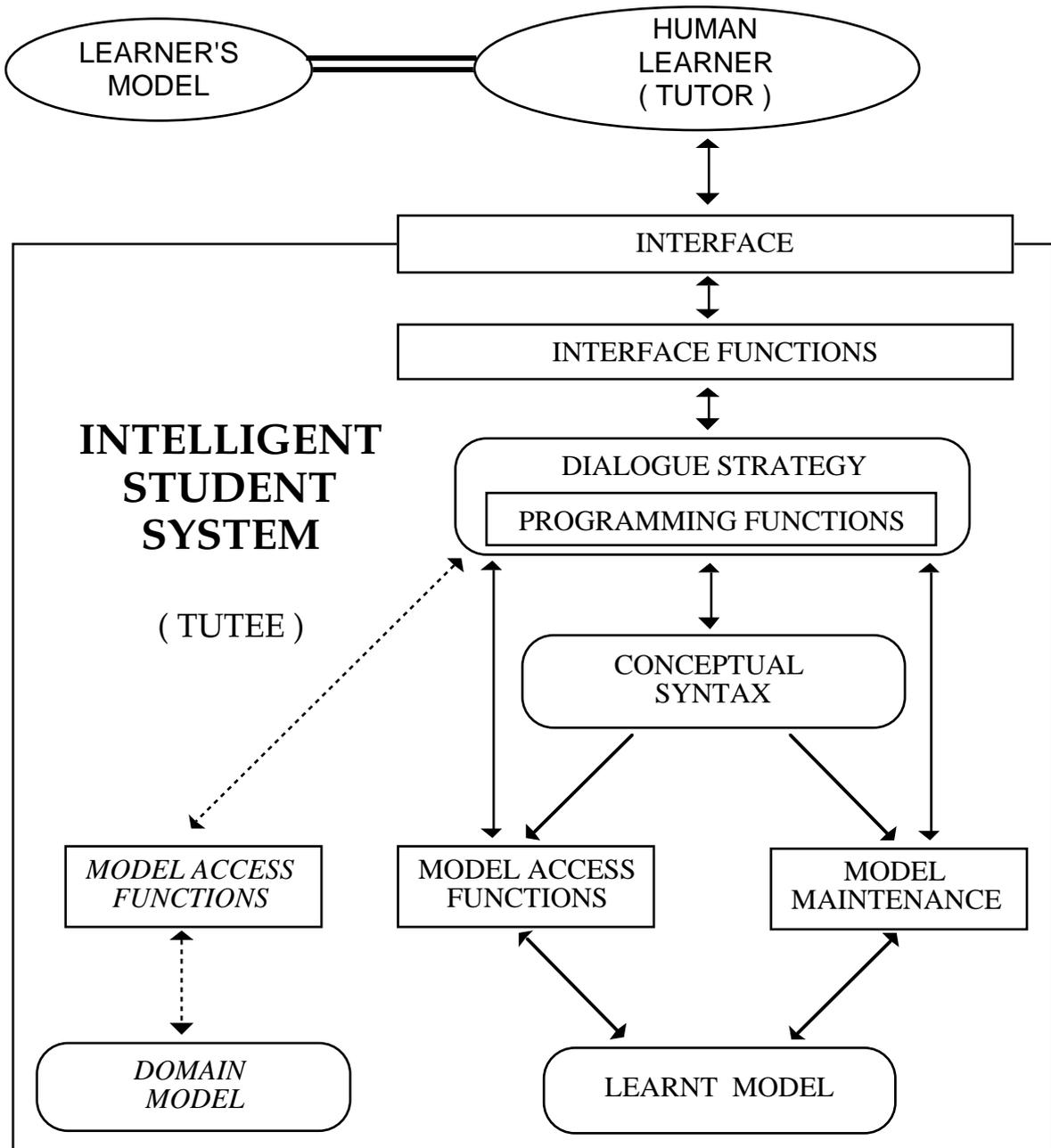


Figure 4.1 Theoretical Architecture of an ISS

4.5 Synthesis

The peer tutoring research suggests that a learner can achieve significant educational benefits through interacting with another learner who is less knowledgeable about a domain. Similarly, an ISS has less knowledge than the learner. This runs counter to the prevailing ILE philosophy of building intelligent systems which are superior to the learner in domain knowledge and have accompanying pedagogical

knowledge to mediate the interaction. The ILE community has been fixated on the conventional teacher-student relationship which has resulted in attempts to build systems which require large amounts of knowledge engineering. Whilst this is not explicitly incorrect the focus on a ITS 'communication' approach [Wenger, 1987] has been, at least partially, a misdirection of the research effort. The input (computational complexity) to output (educational effects) ratio of an ISS approach is considerably better than that of the ITS.

There are two main reasons for the simpler architecture of the ISS:

1) the student model and the domain model of an ITS are merged into the single ISS learnt model. Although the theoretical architecture of the ISS allows a full domain model it is expected that ISSs can function effectively without a fully specified model of the domain. To the extent that ISSs can work with partially full domain models then ISSs are simpler than ITSs.

2) the dialogue strategy of a ISS is based on learning rather than teaching. Teaching effects emerge from placing the human learner in the role of a teacher; they do not have to be explicitly coded in the dialogue strategy. A strategy that teaches is inherently more complex than one that learns because it has to combine information from *two* models (a student model and a domain model) whereas a learning strategy only reasons about *one* model.

The central argument so far can be summarised as:

- ILE research has been biased towards ITSs in domains which can be adequately modelled with single-viewpoints.
- The problems of multiple viewpoints will become increasingly apparent as ILEs move into new domains.
- These problems can be dealt with through improved reasoning techniques or designing new ILEs with knowledge requirements that inherently reduce viewpoint-based difficulties – however any ILE must be 'viewpoint-aware'.
- An ISS is an example of an ILE that attempts to minimise the problems of viewpoints through system design.
- The ISS concept is grounded in current educational practice, peer tutoring, but has only ever been explored once – in a statistical manner in a procedural domain.
- ISSs have the potential for providing a new form of ILE interaction; complementary to, and computationally simpler than, ITSs.

The following empirical research questions arise from the above points:

- 1) can an ISS be used as a learning by teaching tool?
- 2) does it replicate any of the effects of human-human peer tutoring?
- 3) what domain-related knowledge does an ISS require?
- 4) do learners find ISS interaction useful/interesting?
- 5) can ISS results be as general as those from human-human peer tutoring?
- 6) do ISS activities require a concrete task other than teaching?
- 7) does an ISS require an additional student model?
- 8) what type of learning should an ISS perform?

Only *one* experimental study [Michie, Paterson & Hayes-Michie, 1989] has ever attempted to address *any* of these questions (in an atypical domain) and there are sound educational, computational and economic reasons for further investigation of ISSs.

The scope of these questions is clearly greater than the scope of this thesis. Questions 2 and 5 require a greater level of detail and precision in studies of human-human peer tutoring [Kennedy, 1990] before they can be answered in full. Although some aspects of question 2 may be answered relatively swiftly question 5 implies many studies of ISSs covering a diverse set of domains. Similarly, question 8 calls for numerous controlled experiments with ISSs containing different dialogue strategies.

Questions 1, 3, 4, 6 and 7 lend themselves to examination within the resource limitations of this research. Although unequivocal answers are unlikely to emerge from a single study it is with these five questions that the study will be examined in section 7.5. The research therefore requires an ISS to be designed, implemented and tested.

This Chapter has provided the theoretical framework for an investigation into ISSs – the following Chapter describes the design and implementation of an ISS in the domain of economics.

Chapter 5

DENISE: An ISS in the Domain of Economics

5.1 Introduction

The previous Chapter concluded with several research questions about ISSs that can only be answered by experiments with a concrete system. This Chapter describes the selection, and analysis, of the domain of economics and the design of DENISE (Demonstration ENvironment for an Intelligent Student in Economics), an ISS with which to perform such experiments.

5.2 The Domain of Economics

In section 1.3.1 it was noted that the distribution of domains in ILE research was skewed towards formal mathematical domains that were amenable to single-viewpoint descriptions. As one of the claims of this thesis is that ISSs reduce the viewpoint-related problems, that will become increasingly apparent as ILEs expand their domain-coverage, a domain which has inherent-viewpoints is desirable. A separate research paradigm, that of examining 'under-researched' domains, leads to the same conclusion – as domains with inherent viewpoints have been relatively scarce in ILE work. Similarly, the procedural nature of the [Michie, Paterson & Hayes-Michie, 1989] study leads to a preference for a declarative domain.

Domain selection also has to be guided by practical considerations: the availability of domain information (both textual and from human experts), the complexity of prerequisite knowledge, possible sources of test subjects, etc. After consideration of these factors, economics was chosen as the domain for the test of an experimental ISS. The remainder of this section details the justifications for this choice and outlines other relevant work in computational economics.

5.2.1 The Structure of the Domain of Economics

Economics is both a theoretical and an applied domain. Economic theories exist in a conceptual space separate from the measurement of the real economy [Margenu, 1966]. For example, the theoretical concept of

unemployment does not necessarily refer to the published unemployment figures.

In section 1.3.1 economics was considered to be a domain that was less formal than the domains, such as electronics, which dominate ILE research. As with other social sciences, economics is characterised by disagreement amongst its practitioners – and its students. Economic theories describe the real world using different terms and at different levels of abstraction, and consequently generate conflicting conclusions about applying economic policies to the real world.

- Interviewer's speech in *italics*:
L refers to a line in interview 1, 2L to a line in interview 2
- L033 government expenditure is always less efficient than private expenditure, private investment
L034 which is, well, more or less an ideological point of view
L035 *So depending on your ideology you can take either view*
L036 Right
L037 *One would say that the economy would increase productivity, the other would say that this would not happen?*
L038 Not necessarily productivity.
L039 because productivity would be a relation between input and output
L040 and increased government expenditure would just mean that output increases
L041 because demand increases input could increase by the same percentage too
L042 so productivity would remain equal
L043 but since input increases there would be more demand
L044 for like labour which is an input to production
L045 and if you had an unemployment situation this could be relieved
L046 that's more or less the Keynesian view.
L047 Monetarists would say, because government tends to push out private investment
- L132 *What is the effect of a decrease in profits?*
L133 *Now assuming that its a major sector*
L134 Supply side economists would say
L135 decrease in profits is decrease in investments
L136 no money, no investments
L137 and then there would be a downward spiral
L138 less investment, less production, less employment and so forth
L139 *You say supply side economists, does that imply you don't agree with it?*
L140 I would think its not that .. not that easy, that simple
- 2L035 and I guess that we have already talked about it
2L036 that in certain parts of economic theory there is no agreement
2L037 *The kind of Monetarist, Keynesian ...?*
2L038 Right, so we have to agree upon which way the counterpart is following
2L039 before we can go into details
2L040 because details only make sense with the whole thing agreed upon
2L041 so we are talking about the same flows and so forth

Figure 5.1 Three interview extracts with an economist from [Huxor, 1988]

These theories, or viewpoints, are found in economics texts and dialogues and clearly illustrate the alternative conceptual frameworks that economists use to communicate.

[Huxor, 1988] reports the transcripts of two interviews with an economics lecturer, extracts of which are shown in Figure 5.1. These extracts include examples of viewpoint-based reasoning as being integral for communication in the domain, e.g. Keynesian (L046, 2L037), Monetarist (L047, 2L037), supply-side economists (L134, L139). In the first

- 1) Comment on the relevance to Industrial Economics of one of the following:
 - a) The structure/conduct/performance paradigm
 - b) Austrian economics
 - c) Public Choice Theory
 - d) Transactions Costs
 - e) The 'new' industrial economics

- 2) How are the theories of internal labour markets and segmented labour markets related? Is the labour market in Britain segmented?

- 3) Using a theoretical framework of your choice discuss the possible causes of the real appreciation in the value of the US \$ in the first half of the 1980's. What light does this analysis throw on the concept of the 'equilibrium' exchange rate?

- 4) Critically appraise the filtering and trade-off theories of housing markets.

Figure 5.2 Examination questions from [Lancaster-University, 1990]

extract in Figure 5.1 lines L040 to L046 are expressing a Keynesian viewpoint and then L047 puts a Monetarist alternative. In the third extract lines 2L038-2L041 neatly express the importance of agents interacting in the same viewpoint.

The same viewpoint-based reasoning is evident in the assessment of students; Figure 5.2 shows some questions from a final year undergraduate examination paper.

This domain structuring by theory is not simply an aid to explanation when communicating with non-economists or students but is used by economists in their own research. If the viewpoints were only a surface feature introduced during explanations to students they would still be important but less so than if they are a core aspect of the domain. Their centrality can be independently shown using cocitation analysis: cocitation analysis uses the references that authors cite in their writings to produce a map of a domain.

Author Cocitation Analysis (ACA) is a set of data gathering, analytical, and graphical display techniques that can be used to produce empirical maps of prominent authors in various areas of scholarship.

[McCain, 1990]

ACA can be used with cluster analysis to decompose the domain according to 'schools of thought' [McCain, 1983] (or viewpoints). [McCain, 1986] used a panel of macroeconomists to group 41 authors and compared

their similarity judgements with cocitation data. The two approaches share much of the same structure, Monetarists, New Classical Economics, Post Keynesians, etc., with the major difference being accounting for by the economists assigning empirical and econometric authors to particular schools of thought. In addition [McCain, 1990] shows that the migration of certain authors between schools of thought can be identified over time as the bulk of the domain remains stable. The decomposition of the domain in these terms is endemic at all levels of communication about economics.

The language of the *theory* in economics is a pre-requisite for participating in a coherent dialogue with an economist. Ignorance of the theoretical viewpoint of another agent has the potential to render a interaction meaningless and/or inefficient as reasoning effectively takes place within separate languages (which may intersect in unexpected ways).

Viewpoints in economics also appear as levels of abstraction. Economics is often divided into macroeconomics, the whole economy, and microeconomics, individual consumers [Stanlake, 1976]. The concepts and relationships between them are local to the level of the model. This is discussed further in the next section where it is related to work on qualitative causal economics.

5.2.2 Related Work in Computational Economics

Economics has been largely ignored by ILE researchers – the notable exception being the SMITHTOWN [Shute & Glaser, 1990] environment. SMITHTOWN includes a microworld in which students can conduct experiments (e.g. by varying the supply of a particular good) on a simulated town in order to discover economic ‘laws’. The simulation is numerical; with particular prices, numbers of goods consumed, revenues, incomes etc. Other ILE applications, such as the use of microeconomics concepts in [Peachey & McCalla, 1986] and a simulation in [Schiff & Kandler, 1988], have used economics as an adjunct when demonstrating other topics (planning and decision-making respectively). In general, ‘little has been published on economic reasoning and explanation’ [Huxor, 1988].

The use of numerical economical simulations, as opposed to qualitative simulations (e.g. [Forbus, 1984; Kuipers, 1986]), is common in CAI, e.g. [Gudgin, 1987; Hobbs & Judge, 1992; Lumsden & Scott, 1987; Millerd & Robertson, 1987]. Simulations offer students activities which are radically different from a conventional economics curriculum and can give them an opportunity to *apply* the theories they have learnt about in a task-based situation. A common form involves placing students as the

Chancellor of the Exchequer and playing a 'stabilisation game' [Gudgin, 1987] to manage the economy according to some welfare criteria (e.g. inflation below 5% and unemployment below 10%).

The economic simulations used in CAI are *black-boxes* – internal calculations are not available to their users [Breece, 1988]. This lack of transparency can lead to significant problems:

Attempts to test hypotheses may be thwarted because the results arrive after unknown time-lags. Furthermore, the model involves theories with which students may be unfamiliar. These factors could obscure the information provided by the simulation during a run, which in turn could inhibit the learning process.

... the use of entirely inappropriate material imported by students from other sources such as their lectures or their A-level studies.

For many students, misconceptions were deepened by the sophisticated and somewhat unexpected economics used in the simulation.

[Stead, 1990]

In addition, students can suffer from information overload as a small number of equations in the simulation can generate large quantities of data [Lumsden & Scott, 1987]. The students have to map theoretical concepts acquired from their conventional curriculum to the mass of numbers generated by the simulation which cannot communicate in the same conceptual language.

Numerical simulations have a place in economics education but they are not suitable for generating intelligent dialogues about theoretical concepts. As a part of a larger environment, e.g. SMITHTOWN, simulations can be useful but in general they need the addition of a qualitative component as in the SOPHIE system (section 2.2.1).

Although previous work on ILEs and CAI in economics is inappropriate for application to an ISS, there are some non-numerical applications of AI techniques to economics which are relevant.

Qualitative Reasoning in Economics

Economic theories are frequently expressed as qualitative abstractions of highly complex exchange systems for which no complete quantitative models are known or likely to be found

[Farley & Lin, 1990]

The majority of AI work on qualitative reasoning has been in the domains of electrical circuits, medical diagnosis and physical systems [Cohn, 1989]. Qualitative reasoning is thought to more closely approximate to human thought processes than alternatives such as numerical, formal logic or probabilistic reasoning [Console & Torasso, 1990].

Reasons to study formal qualitative models of economics have been listed as:

the lack of consistent quantitative data, the wish to create formal procedures for tracing causal chains, the validation of the structure of quantitative models, and the description of structural changes of economic models.

[Bernsden & Daniels, 1990]

A central objective in the development of qualitative reasoning in economics is to 'fill the gap between the classical number crunching approach and verbal intuitive economic reasoning' [Bernsden & Daniels, 1990].

Economists make extensive use of qualitative models. In the dialogue shown in Figure 5.1 the economist frequently reasons in a qualitative terms, e.g. 'increased government expenditure would just mean that output increases.' Text books frequently represent economic theories as qualitative models [Mado & Sawa, 1989; Dornbusch, 1990] and such reasoning can be extracted from extracts of natural language text [Pau, 1984; Pau, 1986]. The use of qualitative reasoning is, in addition to the reasons noted above, a convenient abstraction for several related reasons:

- to concentrate on the direction of changes in variables (positive or negative)
- to avoid 'hard' mathematics
- to aid reasoning for non-economists such as decision-makers [Farley, 1986]
- to simplify numerical data (as with simulations, above)
- to aid students who are having difficulty with particular concepts, e.g.

Do you think the students even have a good qualitative understanding of economics, rather than a quantitative (one)?

No, I'm not aware of it. I'm not talking about quantitative understanding. I'm talking about when you say a relationship. You know you want to be able to talk about relationships, and that's not a quantitative understanding, you know. You say it's, well, it's inversely related, or it's a linear relationship, or it's a non-linear relationship, or how is it ...? You know, they're puzzled at the words you used, don't ask me why.

Economics lecturer, interview 1, (lines 980-990)
(Interviewer in *italics*)

Qualitative models are usually, though not necessarily, causal²⁴. [Huxor, 1988] includes *part-of* relations in a model obtained from a knowledge elicitation exercise to create a multi-level causal economic model structurally similar to the human physiology model described in [Khuwaja et al, 1992], e.g.

```
link( interest_rates, private_investment, fall ).
link( private_investment, imports, rise ).
link( imports, total_sales, part_of ).
```

[Huxor, 1988]

Although not explicitly stated, the meaning of the first relation, that a rise in *interest_rates* causes a fall in *private_investment*, is clearly intended to also imply that a fall in *interest_rates* causes a rise in *private_investment*. This is the representational approach taken by [Bernsden & Daniels, 1990; Farley & Lin, 1990]. More formally, using the notation of [Kuipers, 1986], the statement that 'demand (D) is a monotonically decreasing function with respect to price level(P)' (or a rise (fall) in the price level has a negative (positive) effect on demand) can be represented as:

$$(1) \quad D = M^-(P)$$

²⁴ A non-causal qualitative model could only describe associations between economic concepts (e.g. high inflation and high unemployment are associated with each other) and so could not be used in a decision-making context. An association between high inflation and high unemployment provides no information that can be used to formulate a policy to, say, reduce unemployment.

[Farley & Lin, 1990]

The causal ordering is from the *independent* variable, the price level (P), to the *dependent* variable, demand (D). A model consisting of such elements, together with *part-of* relations, is capable of generating qualitative predictions of the effects of changes in variables. [Bernsden & Daniels, 1989; Bernsden & Daniels, 1990; Bourguine & Raiman, 1986; Farley, 1986; Farley & Lin, 1990; Farley & Lin, 1991] have all demonstrated that such qualitative models can make predictions consistent with the standard theoretical and numerical predictions of Keynesian economic models. This Keynesian-fixation reflects a lack of variety in choosing theories to model rather than any intrinsic restriction in the technique.

Part-of relations in economic models are treated as a special case of causality, *contemporaneous* causality [Hicks, 1979]; effects from the independent variable influence the dependent variable in the same time period. Standard causal relations are examples of *sequential* causality, where there is a period of decision-making by economic units²⁵ before the effects are propagated. For example, in equation (1) above, a rise in the price level is first perceived by an economic unit which then adjusts its preferences on the basis of this new information and then makes changes in its behaviour which are reflected in an observed change of demand in the market. In *contemporaneous* causality there are no economic units in between the variables, for example:

$$(2) \quad M_d - M_1 - M_2 = 0$$

where

M_d = total money demand

M_1 = transactions money demand

M_2 = speculative money demand

[Bernsden & Daniels, 1990]

There is no decision-making entity between the variables in equation (2): they are merely *balance sheet* equations that reflect a conceptual decomposition of the domain that economists have found useful.

²⁵ An economic unit is any entity that can be regarded as reacting to information and includes consumers, households, workers, firms, governments, trade unions, etc.

The most common use of qualitative causal models is in qualitative simulations [Cohn, 1989]. Given an initial state for the variables changes are propagated to produce qualitative behaviours, e.g. the mass and spring will oscillate with decreasing amplitude [Kuipers, 1986]. In economics almost all of the qualitative models are used for economic simulations, e.g. [Berndsen & Daniels, 1991; Farley & Lin, 1990]. A prerequisite of such a simulation is the existence of the qualitative model; [Cohn, 1989] restates the question that clearly follows:

‘A major unsolved question for reasoning systems of this type is, where are the causal models supposed to come from? All these models are descriptions of situations that carefully include certain features and ignore others in order to produce descriptions that are precisely tailored for the performance of a specific task. ... This question definitely deserves further consideration.’

[Letovsky, 1983]

Acquisition of causal models can be viewed as part of a larger process, that of qualitative simulation. Learning of causal models is typically done by identifying patterns in quantitative data, e.g. [Selfridge, Daniell & Simmons, 1985], rather than interactively acquiring the models from users. The DENISE system described in the following section has an alternative use as an ‘acquirer’ of causal models which could be integrated into a larger educational system where the learner’s task is to run qualitative simulations.

In one sense causal models in economics come from a deeper model of economic systems in terms of agents/actors. This is the model that economists would ‘finally resort to’ [Huxor, 1988] in explaining the domain. The existence of this deep model beneath the common qualitative reasoning provides a possible metric for ISS evaluation:

- if a learner is forced to resort to a deep actor-based economic model in order to explain a concept in qualitative causal terms then the interaction has provoked a self-explanation effect.

Such a pattern of reasoning is also a shift in viewpoint down to a representation of smaller grain-size. Experimental data can be analysed with respect to several perspectives: causal qualitative models, contemporaneous and sequential causality, alternative ‘schools of thought’ and deep agent-based models.

5.2.3 Summary

Qualitative causal models in economics are a good candidate for a conceptual syntax for an experimental ISS for several reasons:

- qualitative reasoning is routinely used in all forms of economic discourse
- qualitative economic models have only been used for simulations and not as the basis for dialogues with computers
- qualitative causal models are built on deep models, based on an agent-action paradigm, which provide a well-understood sub-domain in which to investigate the effects (e.g. self-explanation) on human learners
- economics has been ‘under-researched’ by the ILE community
- model implementation is tractable as shown by qualitative economic simulations
- economic reasoning is viewpoint-based
- the interactive acquisition of causal models may be relevant to mainstream qualitative reasoning and simulation

The design and implementation of a conceptual syntax in DENISE is described in the following section.

5.3 DENISE: An Experimental ISS

DENISE is a development environment, including authoring tools to create and modify dialogue strategies, as well as a run-time experimental system. As described in section 4.4.1 the components of an ISS are: a learnt model, a conceptual syntax, a dialogue strategy, model access functions, model maintenance functions and an optional domain model.

As DENISE is the first system of its type, the appropriate methodology is one of proving the concept of an ISS rather than engaging in extensive experimentation. The dialogue strategy for DENISE has been designed without reference to a domain model to provide a base line with which to assess future additions of domain knowledge. Acquiring a full domain model only to (possibly) discover that it doesn’t contribute to the effects of an ISS is an unsound methodology. Methodology is discussed further in section 6.2.

The following sections describe the implementation of the theoretical ISS components, outlined in section 4.4.2, in DENISE. The dialogue strategy will be explained using an example of authoring a simple

strategy. DENISE has been implemented in MacProlog²⁶ 4.5 on Apple Macintosh computers.

5.3.1 The Learnt Model and Conceptual Syntax in DENISE

The conceptual syntax in DENISE is designed to support the causal qualitative economic reasoning described in section 5.2.2. In addition the learnt model needs to support the different viewpoints that are characteristic of economic knowledge.

The viewpoints of the learnt model are hierarchical in nature with inheritance of beliefs from general to more specific viewpoints. This formulation is consistent with the general approach of the majority of systems in Table 2.2.

As the learnt model has to represent the user's beliefs over time the learnt model must also allow the creation and deletion of viewpoints during the interaction (as opposed to pre-defining them at compile-time).

The conceptual syntax of qualitative relations pervades all of the economic viewpoints that DENISE is designed to represent. Thus the reasoning is global – there are no inference mechanisms local to particular viewpoints. There is complete user-space transparency – all of the learner's beliefs (and the viewpoints that contain them) held in the system are accessible (via dialogue). This means that the system cannot reason in viewpoints which the user has not participated in creating²⁷. The system is thus not imposing any particular pre-defined conceptualisation of the domain upon the user as is the case with an ITS.

The structure of the learnt model is:

```
viewpoint( Viewpoint_Name, Sub_List, Viewpoint_Contents
)
```

The learnt model is based on an permanent viewpoint with Viewpoint_Name of `root`. The Sub_List is a list of the child viewpoints of Viewpoint_Name. The Viewpoint_Contents consists of the qualitative relations of Viewpoint_Name.

²⁶ MacProlog is developed by Logic Programming Associates Ltd.

²⁷ The act of using the system is taken as participating in the special case of the creation of the root viewpoint.

Viewpoints inherit relations from their parent viewpoints unless they contain a different relationship between the same objects. This is similar to the default ascriptional rule used in *Viewgen* (section 2.3.1). In terms of viewpoints the ascriptional rule is: unless a viewpoint contains an alternative relationship between two concepts, assume that such a relationship can be inherited from a parent viewpoint (which itself can inherit).

At its simplest, the relations in the viewpoints of the learnt model can be restricted to just two (a positive and a negative causal relationship), as in [Berndsen & Daniels, 1991; Farley, 1986]. These can be viewed as a specialisation of an unsigned causal relation, the equivalent of 'A is related to B but I'm not quite sure how.' The dual nature of causality in economics suggests that a contemporaneous analogue may be useful, the equivalent of 'A is a part of B but I'm not sure whether it's positive or negative.'²⁸

The basic conceptual syntax used in DENISE is therefore:

- the general unsigned sequential causal relationship
- the positive sequential causal relationship
- the negative (or inverse) sequential causal relationship
- the general unsigned contemporaneous causal relationship
- the positive contemporaneous causal relationship
- the negative contemporaneous causal relationship

The six elements of the conceptual syntax are all binary relationships, which implies that the learnt model will consist of triples, e.g. (interest_rates, neg_seq, money_supply) meaning that a change in interest rates will be reflected, in time, by an opposite change in the money supply. In addition to the six relationships above there is also the question of the representation of completeness; how to maintain a record of whether a set of relationships affecting a dependent variable is complete. It is straightforward to question the user about this issue but the conceptual syntax must allow this information into the learnt model. Therefore, a completeness relationship needs to be added to the six causal relationships

²⁸ Although there is no precedent for his type of relationship, the worst that can happen is that the facility will not be used by learners. An example of a negative part-of relation would be the role of imports in the balance of payments, with exports as the positive part-of relation.

to produce the conceptual syntax, although it is a purely internal construct²⁹.

In section 4.4.1 it was indicated that the learner could change the conceptual syntax of an ISS during an interaction. This operation is theoretically possible but complicated in practice. As a rough analogy consider a program that changed the syntax of its programming language, recompiled itself and continued execution. Alternatively, all of the functions in the model access and model maintenance components would have to be constructed independent of any particular conceptual syntax. These problems are soluble but non-feasible in the context of the resources available for this research. Consequently DENISE is based on a static conceptual syntax. This approach worked in the [Michie, Paterson & Hayes-Michie, 1989] experiments and is implicitly assumed in the proposed [Palthevu, Greer & McCalla, 1991] system.

5.3.2 Model Functions in DENISE

The model access and maintenance functions constitute the interface between the learnt model and dialogue strategy. These functions are the atomic building blocks that the author of the dialogue strategy combines to generate the behaviour of the system. These functions illustrate the application of the viewpoint principle as a central element of knowledge use within the system. The input data to model functions are the plan variables of the dialogue strategy, such as the current viewpoint or the last referenced concept.

Model maintenance functions are similar to model access functions, except that the former alter the learnt model itself instead of the dialogue strategy plan variables. Examples of both types of functions are given – the full lists are in Appendix A.

Model Access Functions

²⁹ This does not violate the user-space transparency principle as it is reflected in the dialogue. When a completeness relation is present, then a list of independent relations has a 'and this list is completed' included. This is to simplify the implementation, as 'completeness' is an unary relation whereas the remainder of the learnt model consists of binary relationships.

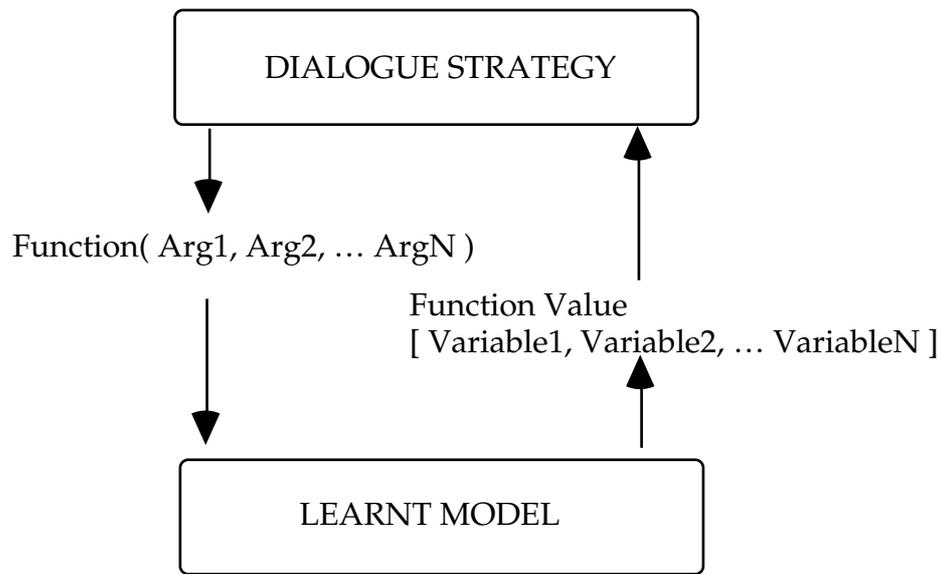


Figure 5.3 The Action of a Model Access Function

The model access functions interrogate the state of the learnt model and return a value to the dialogue strategy. In addition they can also instantiate one or more *plan variables* (see section 5.3.4), as shown in Figure 5.3. A model function need not contain any arguments or affect any plan variables.

Model functions are dependent on a particular conceptual syntax; it is necessary to know the morphology of the learnt model before it is possible to examine its particular characteristics. Given the conceptual syntax described in the previous section then a model access function in DENISE is specified as:

```

find_relation( Viewpoint, Concept1, Concept2 )
{ yes, no }: { Relation }
– is there a relation between Concept1 and Concept2 ?

find_dependent( Viewpoint, Concept1, Relation )
{ yes, no }: { Concept2 }
– is there a dependent variable matching Concept1 and Relation?

sub_viewpoint( Viewpoint )
{ yes, no }: { Viewpoint_List }
– find the immediate child viewpoints of Viewpoint

find_concept_no_independency( Viewpoint, )
{ yes, no }: { Concept2_List }
– find concepts with no causal links to them

```

Figure 5.4 Some Model Access Functions

```

function_name( Argument1, ... ArgumentN )
{ Return Values }: { New Variables }

```

The arguments are instantiated plan variables and are unchanged by the function call. The return value and any new variables created are used in the dialogue strategy (section 5.3.3). The variables serve the purpose of variable procedure parameters in a language such as Pascal [Findlay & Watt, 1985] and allow functions the flexibility to return multiple results. Model access functions leave the learnt model unchanged. The simplest example is:

```

present( Viewpoint, Concept1, Relation, Concept2 )
{ yes, no }:
```

The four-argument boolean access function *present* queries the learnt model as to whether the relationship (*Concept1*, *Relation*, *Concept2*) is present in the viewpoint *Viewpoint*. The value passed back to the dialogue strategy is either *yes* or *no* and there is no change to the plan variables. A further example:

```

peer_viewpoints( Viewpoint )
{ yes, no }: { Viewpoint_List }
```

Here the function *peer_viewpoints* has only one argument but returns the new plan variable *Viewpoint_List* which contains a list of all viewpoints with the same immediate parent as *Viewpoint*. With a library of such model access functions the author of a dialogue strategy can have dynamic access to the learnt model at any stage of an interaction with a

learner. Figure 5.4 lists some more model access functions together with a brief description – complete list in Appendix A.

Model Maintenance Functions

Model maintenance functions are simpler than the access functions in that they do not create plan variables but they do have the side-effect of modifying the learnt model. For example:

```
del_viewpoint( Viewpoint )
{ yes, no }
– delete the viewpoint Viewpoint and its contents
```

Maintenance functions act in the same way as access functions returning a value to the dialogue strategy. In general the value should be *yes*, indicating a successful operation. However if the dialogue strategy is badly designed by its author then maintenance functions will return no values; for example, if *del_viewpoint* is called with a non-existent viewpoint as an argument. The full list of model maintenance functions is listed in Appendix A. A further example:

```
add_to_model( Viewpoint, Concept1, Relation, Concept2 )
{ yes, no }
– add the relationship (Concept1, Relation, Concept2) to Viewpoint
```

5.3.3 The Interface in DENISE³⁰

The model functions outlined in the previous sections enable DENISE to make real-time decisions as to the action to present to the user at the interface.

In Chapter 4 it was noted that an ISS can either be grounded in a task or simply engage in a dialogue about the domain. If an ISS is being taught to perform a task then the task may need to be represented to the learner at the interface. DENISE is not grounded in a particular task (other than teaching) and so the interface consists solely of a dialogue between the system and the user.

³⁰ This section appears before the section on the dialogue strategy as it describes features which are used in the explanation of the authoring component.

At its simplest, a human-computer dialogue can be implemented as a command-line text interface as in SCHOLAR [Carbonell, 1970]. However, such interfaces do not give the user all the information relating to the constraints on valid inputs. In an ISS these constraints take the form of the conceptual syntax. The static conceptual syntax in DENISE means that the interface can easily represent a template to guide and constrain the user during a dialogue.

The representation of dialogue at the interface can essentially take two forms: the user can give the system commands via textual elements (typing, buttons, menus etc) or via direct manipulation (e.g. the Macintosh operating system interface, Designer's Notepad [Haddley & Sommerville, 1990]). The choice of interface style will affect the nature of interaction and the results obtained from any experiments with the system. As indicated in section 4.2.1, a major source of experimental data will be the think-aloud protocols of users. [Svendsen, 1991] shows that during problem-solving a textual interface encourages greater verbalisation by experimental subjects. Users tend to concentrate on the graphical objects themselves rather than on their semantic referents. Practical considerations, and computational complexity, therefore favour a textually based interface design.

The conceptual syntax described in section 5.3.1 is based on relationships between two concepts – the independent and the dependent variable. In addition, the relationships are derived from a finite set. These points suggest a template of three fields (independent variable, relationship and dependent variable) with a menu selection for the relationship field. In combination with a field for textual output from DENISE and buttons to respond to closed questions, a typical dialog window looks like Figure 5.5

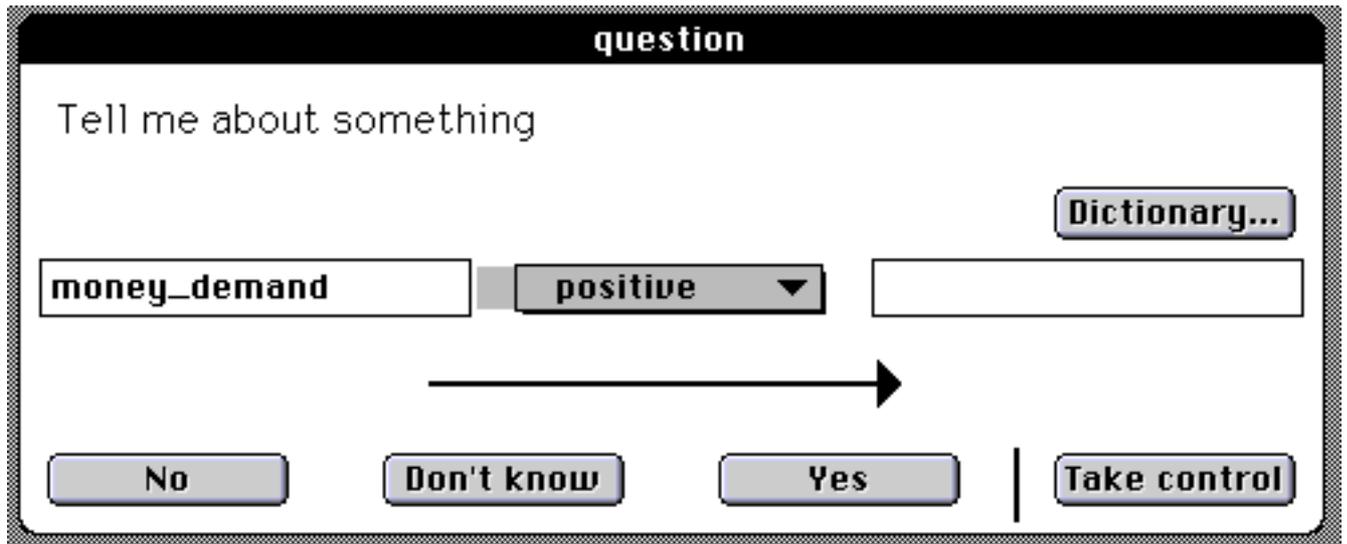


Figure 5.5 A typical dialog window in DENISE

[Palthepeu, Greer & McCalla, 1991] describe a dialogue where the system retains control of the interaction³¹ and the [Michie, Paterson & Hayes-Michie, 1989] system has a large degree of user control. DENISE allows the learner to explicitly take control of the dialogue, and effectively ignore the system's last question, with a button alongside the usual dialogue responses, Figure 5.5.

³¹ Although with frequent 'open' questions like 'Anything else?' or 'What now?'

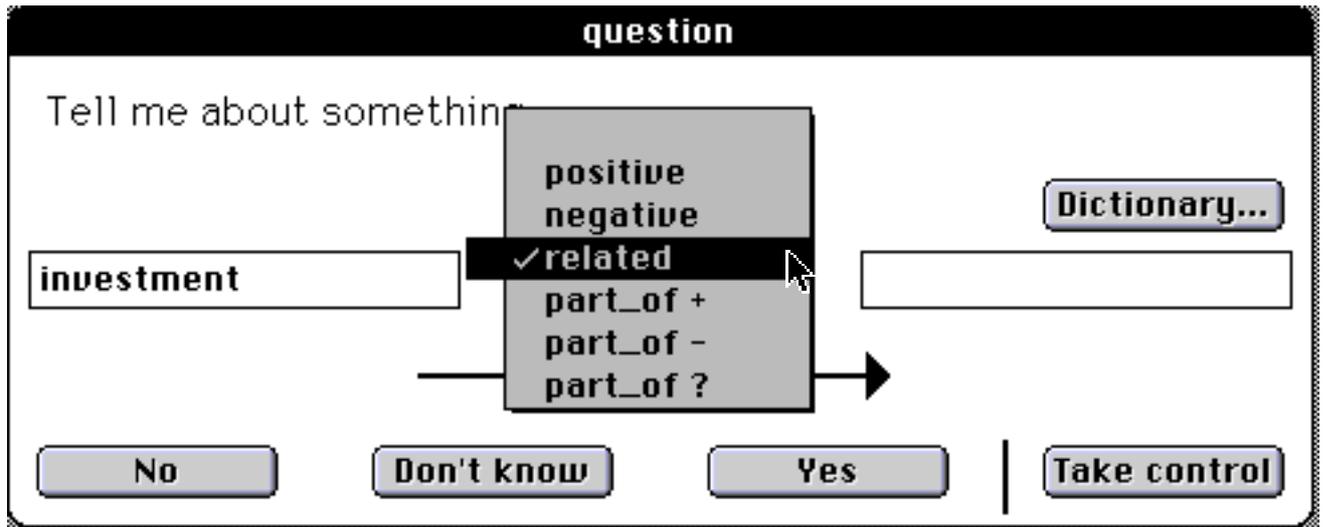


Figure 5.6 A menu constraining input to the conceptual syntax

Figure 5.6 shows the conceptual syntax constraining the user's input by restricting the relation to one of those shown in the popup menu. This format also serves as a clear reminder to the learner of the shared interaction language DENISE is equipped to understand.

The **Dictionary...** button in these dialog windows is an aide to usability rather than a necessary feature of the system. When this button is clicked a separate dialog window is activated which displays a scrollable list of all the concepts that are present in the learnt model, see Figure 5.7. If a



Figure 5.7 The dictionary of previously used terms

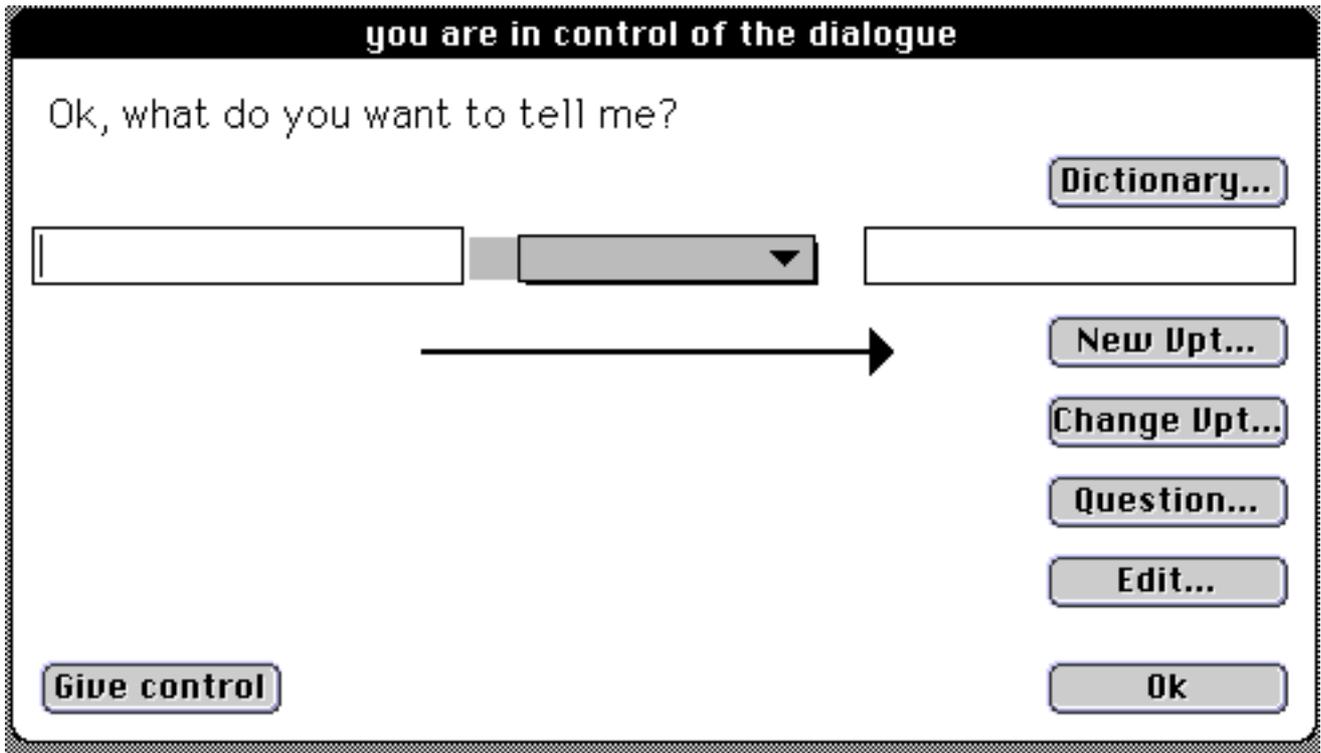


Figure 5.8 The user takes control of the dialogue

concept is selected from this list the dialog window closes and the selected item appears in the main window in whichever edit field contains the cursor. Pilot testing quickly revealed that users tend to forget the exact name they have given a concept, e.g. 'money_demand' or 'demand_for_money'.

Providing a list of all terms they have used reduces the cognitive load on learners in using the system and allows them to concentrate on the domain rather than on the mechanics of DENISE. In addition, list selection is much quicker and easier than (re)typing terms. If users have made typing errors then the alphabetical listing of the term dictionary also makes them easier to locate and correct, Figure 5.7.

The dialog windows return values and instantiate variables in the same way as a model access function – interrogating the user rather than the learnt model. These 'interface functions' are treated identically to the model functions in the dialogue strategy as just a source of information to decide on which action to process next.

The user can take control of the dialogue and is presented with the window shown in Figure 5.8. In this mode the user is not questioned by the system at all and has access to several additional functions. The user

Node Structure:

- Node Name
where Node Name is unique in the dialogue strategy
- Action Sequence: M Actions where $M \geq 1$
where Action is one of
{ Model Access Function, Model Maintenance Function
Interface Function, Programming Function }
- Arc Sequence: N Arcs where $N \geq 1$
where Arc = [New Node Name, Condition]

Figure 5.9 The Structure of a Node in the Dialogue Strategy

can create new viewpoints and change between existing viewpoints. The user can also question DENISE using question marks in the edit fields and the relationship menu as a form of database query within the current viewpoint, e.g. (? , pos, inflation) = 'give me all of the concepts that positively affect inflation.'

5.3.4 Dialogue Strategy in DENISE

The dialogue strategy controls the behaviour of DENISE – it determines what the system does next.

The dialogue strategy, or plan, in DENISE is represented as a network of nodes in a similar manner to the discourse management network [Woolf & McDonald, 1984]. This structure is straightforward to design and implement and allows a strategy author to precisely locate the status of the system at any point in a dialogue.

Figure 5.9 shows the composition of nodes in the dialogue strategy. Each node consists of a sequence of *actions*, for example, a model function or a dialog window to present to the user and a number of *arcs* to other nodes. The final action in the sequence must be a function that returns a value which is then used to determine which node is processed next. In addition, there are plan variables that persist over the network and sub-networks (or *procedures*).

The manipulation of these variables is achieved through a set of standard imperative programming functions: these functions take plan variables as inputs and return values (and create new variables). For

example, functions exist to assign new variables, modify variables, compare variables, etc. All these facilities combine to produce a rich structure for a dialogue strategy author to control the behaviour of the system.

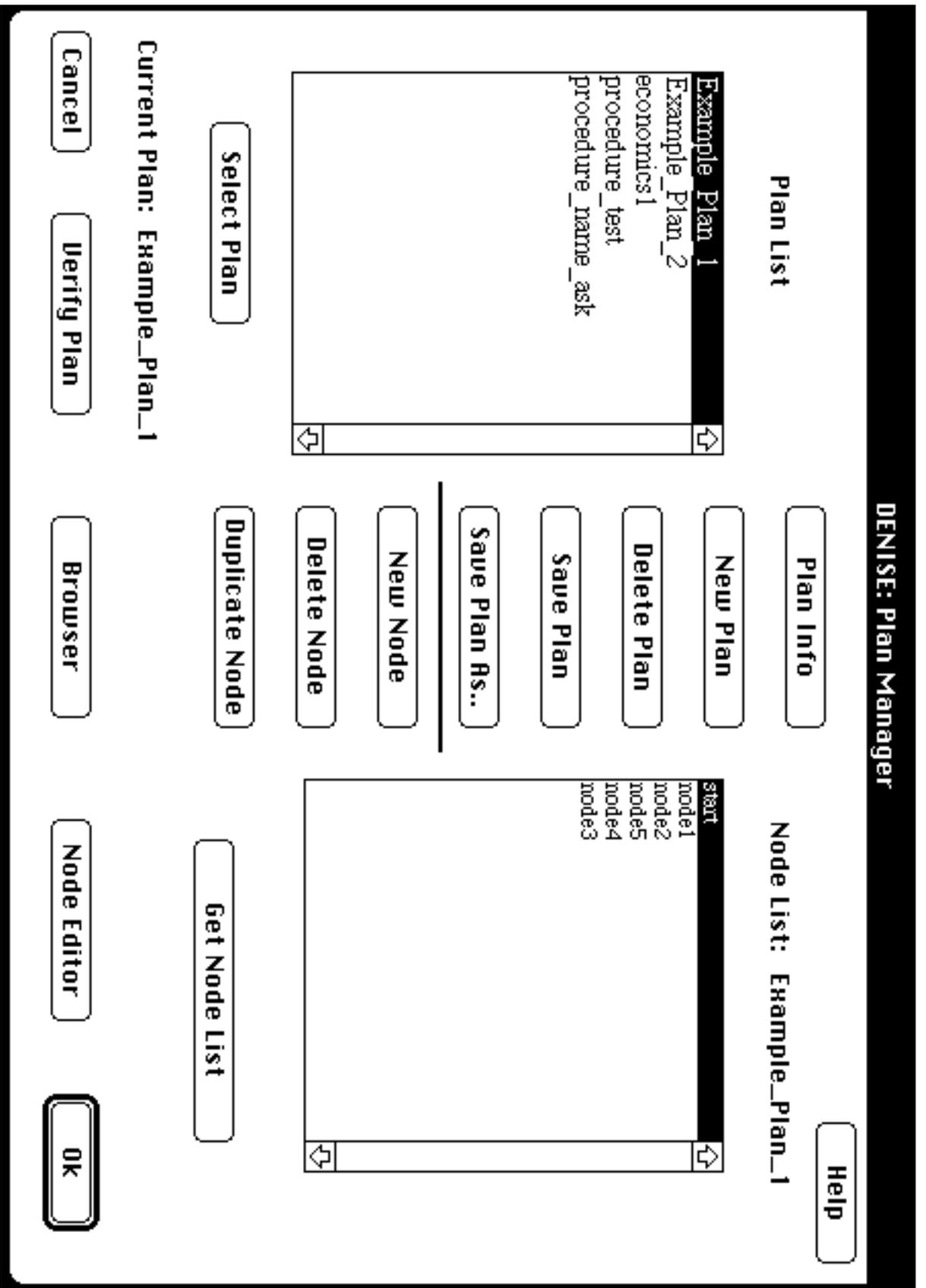


Figure 5.10 The Main Screen of Plan Manager

The structure of a dialogue strategy will be shown through an example of authoring a simple plan using the *Plan Manager* tools in the DENISE system. As a baseline the conceptual syntax and model functions will be restricted to those shown in previous sections.

Figure 5.10 shows the main screen of the authoring environment in DENISE: Plan Manager. Existing strategies (or plans) and partial strategies (or procedures) are shown under **Plan List** on the left. In this case there are three plans and two procedures (which always start with 'procedure_'). On the right is a list of all nodes in the currently selected plan: 'Example_Plan_1', containing only six nodes. The execution of a plan requires a recognised entry point; this is provided by the requirement that every plan must contain a 'start' node.

Figure 5.10 shows buttons which lead to the three other parts of the Plan Manager authoring environment:

- Node Editor: editing of actions and arcs of a particular node.
- Browser: single-stepping through existing plans or sub-procedures
- Verify Plan: produces an analysis of the structure of a plan

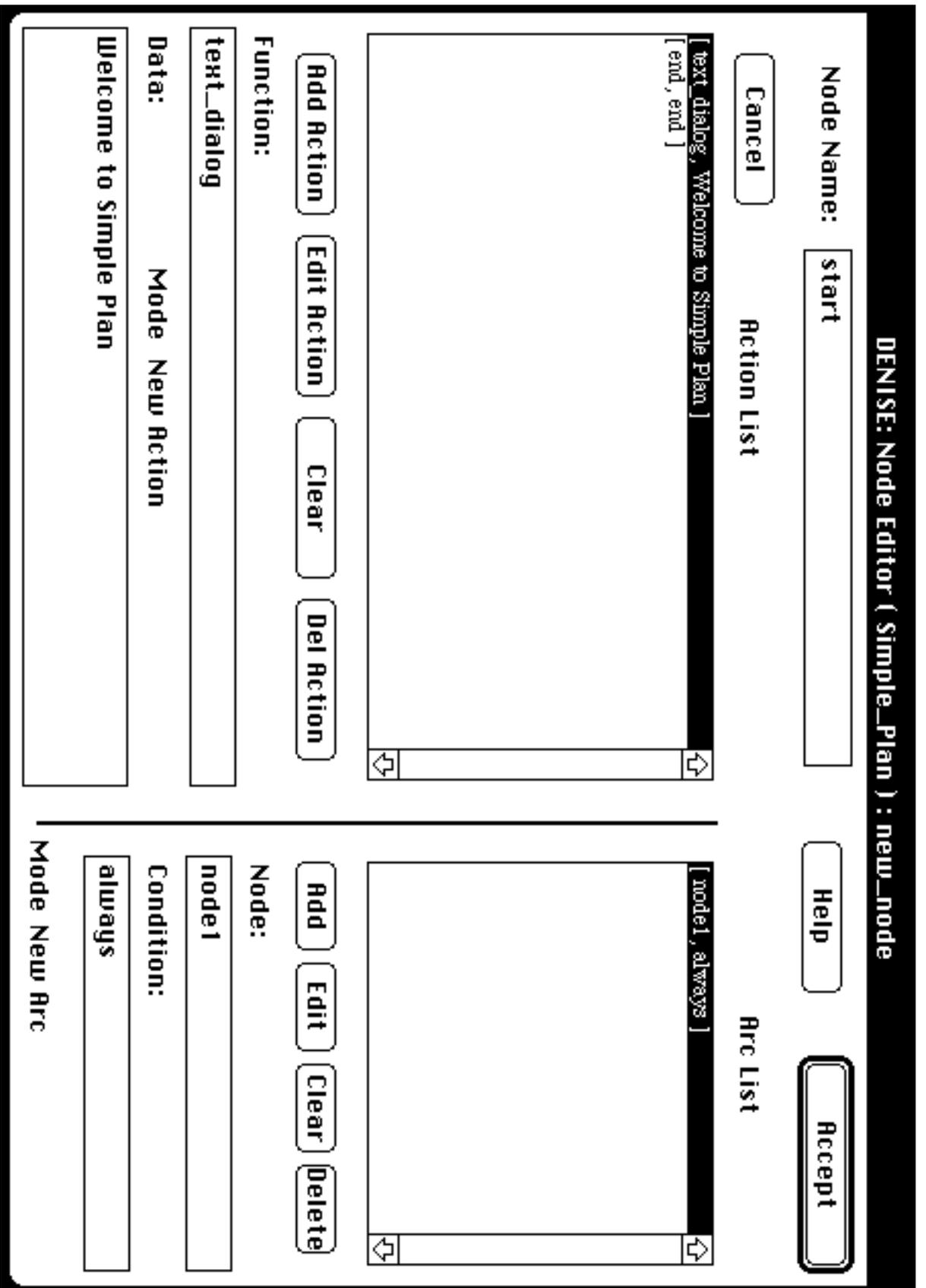
The **New Plan** button creates a new empty plan, called 'Simple Plan' in this case. Clicking on **New Node** opens the node editor as shown in Figure 5.11. The actions of the node are shown on the left, the arcs on the right. The two mode-status messages indicate the last edits performed on the node.

The name of this node has been edited to be 'start' as it will be the entry point to 'Simple Plan'. In addition, one action and one arc have been added to the node. The action is to call an interface function called 'text_dialog' with a text message as its sole argument. The arc is a link to a non-existent node (at present), 'node1', with a condition of 'always'³².

'Simple Plan' consists of just six nodes in total and is listed in full in Appendix B; its structure is shown in Figure 5.12. The graphical complexity

³² 'always' is a special condition which does not rely on the returned value of a function and overrides any other conditions to always proceed to the named node – 'node1' in this case.

of even a simple plan is a further argument for avoiding a direct-manipulation interface. Executing 'Simple Plan', DENISE just expands the learnt model by finding concepts with no cause; nodes with no causal in-arcs. Input from the user is added to the root viewpoint of the learnt model without duplication. The system guides the structure of the interaction as the interface functions called do not allow the user to assume control.



Dialogue strategies exist independently of the ISS component of DENISE. The choice of which dialogue strategy to execute is made at run-time from the library of existing strategies. Plan Manager allows a user to author a dialogue strategy rather than rely on a limited selection provided by the system author. The user is limited by the model functions provided by the system author but these allow the development of arbitrarily complex dialogue strategies through well-known programming language constructs, e.g. iteration, branching, procedures, argument passing, variable scope, etc.

5.4 Summary

A system has been designed and implemented within which to conduct experiments to investigate the use of an ISS. The domain for investigation, economics, has been analysed so that informed judgements can be made on the experimental data that is generated. The next Chapter describes the particular dialogue strategy and experimental method used in examining the effects of interactions with DENISE.

Chapter 6

Experimentation

6.1 Introduction

This Chapter takes the DENISE system shell outlined in Chapter 5, initialises it with a particular dialogue strategy and describes the experimental methodology adopted for evaluating the ISS concept.

In order to evaluate the theoretical ISS architecture proposed at the end of Chapter 4 the DENISE shell requires three elements: a conceptual syntax, a set of functions (model access, model maintenance, interface and programming) and a dialogue strategy. The first two elements were discussed in sections 5.3.1 and 5.3.2 respectively. The dialogue strategy, section 6.2, will determine the behaviour of the system, specifying how much control the user has over the direction and content of the interaction.

Section 6.3 describes the experimental methodology adopted for the evaluation of DENISE. The collection and analysis of data from various sources is discussed with respect to a novel ILE such as an ISS. Section 6.4 outlines the experimental procedures. Analysis of the results obtained is reported in Chapter 7.

6.2 The Experimental Dialogue Strategy

There are many possible dialogue strategies that can be associated with a particular set of available functions. Although the strategy, functions and conceptual syntax are in some ways jointly determined it is in the dialogue strategy that the behaviour of the ISS is embodied.

The two previous pieces of work on learning by teaching do not give much information at precisely what behaviour the ISS should generate. The [Michie, Paterson & Hayes-Michie, 1989] system's strategic behaviour is limited to checking the validity of user input with respect to 'well-formed' equations. The interaction is driven almost entirely by the user's own wishes.

[Palthevu, Greer & McCalla, 1991] are more forthcoming in that they suggest several pedagogical strategies from two sources: knowledge

validation and dialogue control. Knowledge validation heuristics suggested are:

- confirm existing knowledge
- elaborate new knowledge
- find contradictions
- check for completeness of knowledge

These general heuristics are instantiated by their (implicit) node-based conceptual syntax, for example elaborate new knowledge could be either elaborating new nodes ('Can you list them?') or elaborating new properties of nodes ('Can you say something more about mammals?').

The second source of strategic information comes from dialogue control. [Palthepeu, Greer & McCalla, 1991] list several suggestions:

- providing continuity
- attempting to cooperate with student preferences
- encouraging a shift in initiative (e.g. by using a non-committal default responses, such as 'Ok please proceed')
- following 'familiar' knowledge acquisition patterns

A component called 'pedagogical mediation' is responsible for selecting the next action from the suggestions provided by knowledge validation and dialogue control. As an example they propose that:

apparent contradictions discovered by Knowledge Validation would normally outrank all other considerations since it is important to clarify contradictions in the student's mind by pointing them out.

[Palthepeu, Greer & McCalla, 1991]

The broad thrust of the [Palthepeu, Greer & McCalla, 1991] proposals is undoubtedly correct; students do check what they know, ask questions to expand their knowledge, etc. (and these system actions should provoke the user into cognitive activity). Students also tend to react to explicit contradictions, although people do maintain inconsistent beliefs with surprising persistence [Gardenfors, 1988].

The design of a dialogue strategy is the answer to the question: how should an agent behave in order to cause its teacher to examine their own knowledge? Aside from the [Palthepeu, Greer & McCalla, 1991] proposals there appears to be no other directly relevant work to formulate such an

answer (although there is no guarantee that any human-human guidelines would transfer over to an ISS interaction).

The design of the experimental dialogue strategy is therefore to take these suggestions as a baseline and expand and adapt them to the experimental domain of economics. For example, the elaborate new knowledge heuristic could be interpreted as:

Given a concept C ,

- ask for a new relationship where C is the independent variable
e.g. what does a change in C affect?
- ask for a new relationship where C is the dependent variable
e.g. changes in which concepts would cause a change in C ?

The check for completeness of knowledge heuristic can be similarly transferred as:

- ask whether C is an independent variable to other dependent variables
e.g. does C affect anything else other than {list of existing relationships}?
- ask whether C is a dependent variable to other independent variables
e.g. does anything other than {list of existing relationships} affect C ?

Similar transfers apply to the contemporaneous causal relationships.

An important extension over the [Palthepu, Greer & McCalla, 1991] proposed system is that DENISE supports viewpoint-based reasoning and can therefore reason between viewpoints, ask about similarities and differences between viewpoints, switch the dialogue focus to another viewpoint etc. For example, a candidate heuristic is to try to identify similarities between viewpoints and move them 'higher' up the learnt model:

- e.g. as relationship R is the same in both *Viewpoint1* and *Viewpoint2* is it reasonable to move it into *Viewpoint3*?
{where *Viewpoint3* is a common parent of *Viewpoint1* and *Viewpoint2*}

There is a danger in the design of the dialogue strategy that the system will become repetitive (and possibly less interesting for the user) and explore new topics in the same manner as it done in the past. Two

mechanisms exist to counter this tendency: initiative shifts and explicit repetition avoidance. Initiative shifts are produced by the system simply responding with a non-committal default response, as above, which forces the responsibility for the direction of the interaction back onto the user. Also, the dialogue strategy can maintain a history of the interaction and always select a different option to the last similar choice. A less computationally intensive solution is to introduce a random programming function so that, at various points, the dialogue strategy becomes indeterminate.

The combination of these various elements into a strategy is somewhat arbitrary as this process has never been attempted before. Some principles, such as an emphasis on contradictions, can be justified with reference to psychological theories. Contradictions are a source of cognitive (or conceptual) conflict [Hewson & Hewson, 1984; Nussbaum & Novick, 1982] and so can serve as a trigger for self-explanation and belief revision. Also, a contradiction is likely to most effective when presented as soon as possible after the belief it is contradicting ('That's not what you just said' is more likely to be effective than 'That's not what you said an hour ago'). As to the relative merits of other heuristics the best approach is probably to await empirical results. The 'pedagogical mediation' component of the [Palthevu, Greer & McCalla, 1991] system is noticeably lacking on further elaboration concerning this issue.

The strategic issues involving viewpoints are equally underspecified. Many systems reviewed in Chapter 2 would switch to another viewpoint when reasoning in the current viewpoint failed. Such behaviour does not provide heuristics to determine when an ISS should try to shift the dialogue into a new viewpoint. A simple solution would be to simply abrogate the responsibility and leave viewpoint shifts to the initiative of the learner.

In this exploratory research there doesn't seem to be a case for including extensive machine learning techniques to infer additional beliefs to those in the learnt model. This is for the same reasons as the omission of a domain model – to provide a baseline from which to proceed. Instead, single 'proposals' (e.g. try and find if this concept has any related independent variables) are made and used as the basis of a question to the learner.

The experimental dialogue strategy is embodied in the plan 'economics-exp' which is listed in full in Appendix B. The strategy centres around expanding the learnt model through the elaborate knowledge

heuristic. It incorporates indeterminism through the use of random numbers to try to minimise repeated sequences although significant use is made of default non-committal responses. The strategy allows the user to take control of the dialogue and continue the interaction without interruption from the system. The strategy also incorporates explicit viewpoints and requests to add new viewpoints to the learnt model.

6.3 Experimental Methodology

It is generally accepted that a good evaluation methodology is lacking in most intelligent tutoring systems (ITS) research [Baker, 1991; Park, Perez & Seidel, 1987; Shute, 1990]. ... What factors contribute to the current lack of sound evaluation – why is it so difficult or rare? One reason is that, as a research field, ITS is relatively new and we do not have an accepted cadre of formal research methods or metrics.

[Murray, 1993]

In response to comments such as this the evaluation of ITSs has recently shown a trend towards more formal summative methods, e.g. [Legree, Gillis & Orey, 1993; Shute, 1993; Shute & Regian, 1993]. This tendency has become prominent after many years of research refining the concept of the ITS through qualitative formative evaluation of partial systems in ‘toy’ domains. The achievement of a summative environment for ITS evaluation can be regarded as a indication of the maturity of the field. However there is still a valuable place for formative evaluation: [Twidale, 1993] notes several problems in the controlled summative evaluation of ILEs:

- rigorous experiments are large, slow and costly
- a controlled experiment only really measures one thing
- a controlled experiment produces averaged out figures of overall performance
- unexpected interactions may lead to misleading results
- the possibly overpowering effect of the interface
- learning to learn with an ILE takes time

These points have to be balanced against the argumentative power of statistically valid controlled experimental results [Shute, 1993]. The progression from informal evaluation of ITSs to the expectation of formal controlled evaluation has taken many years. The ISS concept, in contrast, is right at the beginning of this sequence; only *one* system has ever been built and evaluated.

[Murray, 1993] lists three requirements for a research area to be classed as 'exploratory' and therefore to be amenable to formative research:

- few or no empirical studies have been published
- there are no generally acknowledged experimental or evaluation methods, standards or metrics
- there is no consensus in the field on the most important issues, problem areas, and trade-offs

There has been one empirical study published [Michie, Paterson & Hayes-Michie, 1989]. As ISS interactions are presently *black-boxes*, there is no history of problem areas, trade-offs or evaluation methods. These considerations, together with the limitations noted by [Twidale, 1993] above, and the resource constraints on this research lead to the adoption of a formative qualitative methodology for the evaluation of ISSs in general and DENISE in particular.

Evaluation Method	
<i>example</i>	
Usage in evaluation of DENISE	
1.	Outside Assessment <i>e.g. expert panels</i> Not used
2.	Wizard of Oz Experiments <i>e.g. human simulates system behaviour</i> Not used
3.	Existence Proofs <i>e.g. prototype system, toy domains, inductive argument</i> DENISE
4.	Observation and the Qualitative Classification of Phenomena <i>e.g. task analysis</i> Observation of users interacting with DENISE
5.	Structured Tasks and the Quantitative Classification of Phenomena <i>e.g. interviews, questionnaires, statistical methods, concept maps</i> Post-Session Interview
6.	Performance Metrics <i>e.g. feature usage, cognitive change</i> Feature Usage
7.	Internal Evaluation <i>e.g. program traces</i> DENISE internal system trace
8.	Comparison Studies <i>e.g. corroboration and duplication</i> Comparison with peer tutoring studies

Table 6.1 Evaluation Methods for Exploratory
ITS Research from [Murray, 1993]

The experiments are not intended to answer all of the ISS research questions posed at the end of Chapter 4 (which would be unrealistic) but to make the first attempt at defining the area.

Methods

Although ITSs differ from ISSs in their architecture, role and intended effects on their users the evaluation methods used are still relevant. [Murray, 1993] lists eight main evaluation methods for

'exploratory' ITS research; Table 6.1 lists them together with their analogues in the evaluation of DENISE. The only method that is not available for evaluating DENISE is number two, the *Wizard of Oz* experiment. The Wizard of Oz experiment is characterized by an expert communicating with a user via a terminal rather than face-to-face. The expert can then simulate the behaviour of a computer program and the user is unaware that a person is responding rather than a computer system. At present the system is not capable of replacing the control of the dialogue strategy with human instructions.

DENISE is in itself an existence proof (category three) of the ISS concept; these systems are feasible. Observation of users' think-aloud protocols and of the actual dialogue they engage in with DENISE will constitute the data for method four. This data can be analysed with respect to the conceptual syntax of the interaction, the theoretical viewpoints and domain concepts of the domain of economics.

Users were also asked to complete a questionnaire about their experience with the system. Recording of sessions, both externally and within DENISE, allows analysis of feature usage during interactions. The internal system trace will also record the frequency with which various parts of the dialogue strategy are used. The comparison studies are limited by the novel nature of ISS research. The [Michie, Paterson & Hayes-Michie, 1989] system was evaluated with pre and post-tests and a brief six-question questionnaire³³. There is no reported work on interactions between learners and learning-by-teaching systems.

As noted in section 4.2.1, there appear to be no protocols of tutor-tutee interactions in peer tutoring schemes (observations in category four of the methods in Table 6.1). However, some qualitative research (interviewing and questionnaires) does exist for peer tutoring schemes and can be used as a basis for comparison with DENISE.

³³ Questions included: 'what did you like about the program?', 'what improvements would you make to the program?', 'do you have any other comments about the program?'. Two four-point scales (very boring-slightly boring-quite interesting-very interesting, worse than before-the same-a little better-a lot better) were used to measure interest and equation understanding, respectively. Only the answers to these final two questions were reported.

It is also possible to identify where evidence of specific effects should be located, for example, evidence of self-explanation (if present) will appear in categories four and five, evidence of belief revision in category four. Categories one, two (not used) and three (the existence proof) have been dealt with prior to the actual experiments.

6.4 Experimental Procedures

The system was pilot tested with members of the Computing Department. The main result of this was to add the dictionary mechanism shown in Figure 5.7. This proved an effective mechanism for preventing typing mistakes and for speeding up the user's input to the system. The wording of some buttons was also changed to reduce ambiguity as to their function.

Subjects

The domain of economics, as with most social science domains, is understood at many different levels of detail according to subjects' previous exposure to relevant material. This can range from the qualitative causal understanding expressed in the conceptual syntax of DENISE to a detailed mathematical understanding of, say, production functions. Subjects varied in knowledge from 'naive economics' to having taken a formal economics courses.

Subjects were given a short introduction (about five minutes) to qualitative causal reasoning, using the related domain of politics as an example, to ensure they shared the conceptual syntax of the system. The examples are shown in Appendix C. Subjects were then familiarised with the interface of the system, the use of the mouse, etc.

The subjects were given instructions to teach DENISE about economics and asked to think-aloud during the session. Subjects were paid for their participation. Five subjects were recorded using the system.

Physical Environment

The experiments were conducted on a Macintosh LCII computer and were recorded using a video camera. Experimenter interventions were restricted to prompts to encourage the subjects to verbalise their reasoning or to recover from an interface problem. At the end of the session subjects were informally asked to give their comments in general terms about the system.

Chapter 7

Experimental Results

7.1 Introduction

This Chapter presents and analyses the results of the user trials with the DENISE ISS system. These results appear to be the first ever study of users interacting with an ISS, or anything similar.

Five subjects were recorded interacting with the system and interviewed immediately after the session (total time about 45 minutes). The first two subjects (A & B) had previously taken formal economics courses whereas the other three (C, D & E) had no experience of economics education. All of the subjects were computer literate and had no difficulties with mouse and keyboard operation.

7.2 Experiences with the Interface

The interface was generally well received by the subjects with few observed problems or complaints afterwards. The subjects often exhibited long (up to 30 seconds) pauses between entering relations into the system. However, these were clearly due to cognitive processes, as once subjects announced they had found a relation they usually proceeded swiftly to communicate it to DENISE.

Table 7.1 summarises the subjects' usage of various system features. The sequential causal relations were more popular than contemporaneous relations although subject C constructed a model without any negative relations at all. All of the subjects used the dictionary feature extensively throughout the interactions; as suggested by the pilot tests. Only three of the subjects (A, B & E) took control of the dialogue from DENISE and only two new viewpoints were created (by B & E)³⁴.

One specific problem reported by subject B was a variance in the 'continue' button which was labelled 'yes' in all the dialog windows except the 'take control' dialog where it was labelled 'ok'. The subject was temporarily confused about whether it would have the same effect as the 'yes' button. This is a minor version of a similar problem observed by

³⁴ Although E's new viewpoint was never used.

Subject	A	B	C	D	E
Feature					
<i>positive relation</i>					
<i>negative relation</i>			X		
<i>related relation</i>		X	X		
<i>part_of_+</i>		X	X		
<i>part_of_-</i>		X	X	X	X
<i>part_of_?</i>		X	X		X
Dictionary					
Take Control			X	X	
Create Viewpoint	X		X	X	
Change Viewpoint	X		X	X	X
Question	X		X	X	
Edit Concept		X	X	X	X

Table 7.1 Feature use by the experimental subjects

[Twidale, 1989] in testing the EPIC system, where a badly worded prompt caused users to change their behaviour.

Other problems with the interface were minor; such as the dictionary window overlapping the main dialog window and, after an inadvertent click by the user, preventing the user from continuing until it had been closed.

The dictionary acquired an extra function than the one for which it was designed; namely speeding up input and reducing typing errors. The

dictionary dialog window was used as a source of inspiration for subjects who couldn't think of a new relationship, particularly when DENISE gave a default non-committal response:

B: I did find I was using the dictionary for ideas, once I'd built up a dictionary

B: rather than coming out with new stuff

B: So I was restricting the amount of relations, the things I was telling it about

B: was between all the different things I'd told it

Typical interview comments about the interface:

C: It was fairly obvious what was going on from the buttons,

C: there wasn't anything left unclear

D: The windows are easy to use

E: It was all pretty straightforward, yeah

The success of the interface can be attributed to the piloting, and modifications, of DENISE and the close mapping of the conceptual syntax to the display actually presented to the subjects.

7.3 Dialogues

DENISE records the dialogues of users, an example is shown in Figure 7.1. This dialogue shows some of the basic actions of the experimental dialogue strategy: creating an initial viewpoint, elaborating knowledge and using default non-committal responses to encourage the user to take the initiative.

The subjects were generally agreed that the default responses were 'harder' to respond to.

A: When it asks you to tell you something you hadn't already told it-

E³⁵: A blank, default one, rather than-

A: Yeah, cos, you can think of things

A: to link things together with a lot easier if

A: you've got something there you can think, oh well

³⁵ E = experimenter

A: that makes me think of something else

A: whereas if you haven't got anything there

A: you got to think of something to link something to in the first place

Talking about questions that asked about a specific concept:

B: Yeah, that was easier cos I only had to think of half a thing

B: plus it was making me think as well

B: forcing me, sparking things off in my mind without me having

B: to spark them off in it

D: Welcome to DENISE and the economics_exp1 plan

D: I'll have to place your beliefs into a viewpoint so the first thing to do is to give it a name

U: Create a new viewpoint called *{subject's first name}* beneath root

D: Ok. What else do you want to tell me?

D: Ok, tell me something about economics

U: price is negatively related to sales

D: Ok so what affects price then

U: demand is positively related to price

D: Ok, go on

U: tax is negatively related to spending power

D: Ok so what affects tax then

U: govt_spending is positively related to tax

D: Ok, go on

U: govt_spending is negatively related to unemployment

D: Ok so what affects govt_spending then

U: unemployment is positively related to govt_spending

D: Ok, go on

U: mortgage_rates is negatively related to house buying

D: Ok so what affects mortgage_rates then

U: building_society_greed is positively related to mortgage_rates

D: Tell me something about economics

U: economic_optimism is positively related to house buying

D= DENISE U=USER

Figure 7.1 A portion of DENISE's dialogue record of subject B

One subject particularly disliked the default responses:

D: it would have been better if it had just pulled something off

D: the menu of things I'd already said at random

D: and said what affects this

Subject response times at default responses tended to be longer when the initiative was passed to them. Together with the infrequent use of the 'take control' aspect of the system it seems reasonable to conclude that the subjects were generally uncomfortable in the role of initiative taker.

One subject went as far as:

A: Definitely put me off teaching for life

A: There's no way I stand up in front of a classroom of people and make a fool of myself.

A: A computer I can just about handle I suppose.

This preference for computers over people was suggested in section 4.2.2. The positive part of the default response is that the subjects agreed it made them think about the domain. However, the rate of interaction with the system fell to virtually zero when presented with the default response.

Subjects showed a reluctance to verbalise their reasoning despite continued prompting from the experimenter.

E: What's it like?

A: Terrifying, absolutely terrifying

E: Why?

A: Well I don't know.

A: Someone else watching you

A: while you're trying to think about

A: something you don't know much about

The presence of the experimenter has clearly had an adverse effect on the subjects' propensity to verbalise their reasoning. Where verbalisation did occur it was mainly in the form of the relationship that

Selects **negative** relationship, clicks on 'yes' button

they subsequently entered into the system. At no point did any of the subjects verbalise a shift to a deep actor-based model of the economy. It is impossible to tell whether this means that no such model was used or whether it was used but not verbalised.

Sometimes it appeared that it hardly mattered whether the system was intelligent. For example, subject B:

Enters **price increase** in the independent concept edit box

Enters **sales** in the dependent concept edit box

B: Now have I mucked it up by putting price increase...

B: Cos I'm almost assuming the direction already-

B: If I was just to say price on this-'

Changes **price increase** to **price**

Selects **negative** relationship, clicks on 'yes' button

Here the act of explicitly stating the relationship causes the subject to realise that price increase is not a sensible concept as an independent variable. Similarly Subject B again,

Enters **government_spending** in the independent concept edit box

Enters **unemployment** in the dependent concept edit box

B: More government spending more....

B: oh no, I've the cause going the wrong way here

B: ok, there's a negative response, I can have that

Here the subject is about to enter a positive relationship when he realises it is incorrect, and then realises that a negative relationship does in fact exist between the two concepts.

7.4 Models

All of the subjects managed to communicate a qualitative economic model to DENISE, for example Figure 7.1. However nearly all the models

Subject	A	B	C	D	E
unique concepts	11%	69%	55%	61%	78%

Table 7.2 Percentage of unique concepts generated

were a single-viewpoint and the one subject (B) that did create a new viewpoint did so more out of experimentation than a purposeful commitment to describe an alternative view.

The positive and negative causal relationships overwhelmed the other relationships in rate of use. Many of the relationships would be conventionally regarded as 'misconceptions', e.g. (subject D)

U: unemployment is negatively related to state of economy

This states that a rise(fall) in unemployment would cause the economy to worsen(improve).

The domain concepts entered into DENISE varied widely ranging from traditional economic concepts to politics and even to religion and wars (subject D):

U: religion part_of_+ wars

D: Ok go on

U: religion is negatively related to crime

This sort of *domain drift* is a consequence of exploring the learner's viewpoint rather than restricting the interaction to a well defined domain. Learners do not necessarily have the same conception of a domain as the system designers. The wide variety of domain topics covered suggest that the models acquired are indeed individualised. Many of these concepts would not be encountered in economics courses. The concepts used by subjects can be compared with concepts used by the other subjects, as in Table 7.2. 55% of concepts referred to by subject C were not referred to by any of the other subjects. Subject A, who had economics experience, concentrated on a small number of well known concepts (e.g. exports, imports, interest rates, unemployment etc) and produced a more tightly connected model.

The unfortunate lack of alternative viewpoints expressed is a consequence of the freedom of the system, if it does not force users into moving viewpoints and they don't want to then they will remain in the current viewpoint. This is a form of cognitive economy [Scanlon & O'Shea, 1988].

Subject E drew an analogy with supervision of undergraduate students which is reminiscent of peer tutoring reports:

E: I did some demonstrating in biology to some of the undergrads

E: and, you know, you're there doing a higher degree

E: and everything and they're just at the beginning

E: and you talk to them, it gets you thinking

E: some of the things they ask you

E: you think , Oh I never thought of it that way

7.5 Summary

Bearing in mind that the sample size for the trials was small the results can be summarised as:

- DENISE can capture individualised models from users
- users are forced to examine their own beliefs
- users do revise their beliefs as they interact with DENISE
- *domain drift* occurs
- users were reluctant to verbalise their reasoning in the experimental situation
- users prefer to remain in a viewpoint rather than move to a new one
- default non-committal responses slow down interactions but may provoke reasoning

Research Questions

In section 4.5 eight research questions were outlined and five were identified as being appropriate to this research:

Question 1: can an ISS be used as a learning by teaching tool?

Yes. The experiences with DENISE indicate that the ISS concept can be used to provoke human learners to examine their own knowledge about a domain. This self-examination, or reflection, is held to be the critical part of the learning by teaching paradigm.

Question 3: what domain-related knowledge does an ISS require?

It is clear that an ISS requires some knowledge of the domain. However, this knowledge need not be of the 'full domain model' type required by the ITS approach. The DENISE experiment provides support for the view that only knowledge of the *type* of domain concepts is necessary. The conceptual syntax in DENISE contained no knowledge about the domain of economics yet none of the subjects realised that the dialogue was essentially domain-independent.

Question 4: do learners find ISS interaction useful/interesting?

Yes. The subjects, although not explicitly learners in the domain, found the interaction particularly interesting. Without an interest in the domain they did not find the dialogue *useful* but could see how a dialogue in an appropriate domain might be helpful. All of the subjects thought the role reversal aspect of the ISS interaction was interesting and stimulating.

Question 6: do ISS activities require a concrete task other than teaching?

Maybe. The limitations of the dialogue strategy and the lack of a perceived goal to the interaction were apparent in the post-experiment interviews. A definite goal to the session would have provided focus to the dialogues and possibly prevented the feeling of 'being lost' in the domain.

Question 7: does an ISS require an additional student model?

Probably Not. The subjects were not aware of the 'ignorance' of the ISS during the interactions and yet proceeded to develop several complicated economics relationships. It does seem to be the case that the system didn't need to know anything about the learners to provoke a viable dialogue.

Further discussion is found in section 8.1.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This Chapter summarizes the research contributions of the thesis and outlines possible directions for future work on ISSs.

The thesis is based around two senses of viewpoints:

- the viewpoints of domains
- the viewpoint of the learner

Traditional ILEs have tended to reduce the importance of both of these concepts and emphasized the role of domain knowledge. This thesis represents a part of the trend towards knowledge negotiation [Moyses & Elsom-Cook, 1992] rather than knowledge communication [Wenger, 1987].

Research into learning by teaching systems is still in an embryonic stage. These systems have the potential to produce novel forms of learning interactions which do not require the computational sophistication of many ILEs. The lack of a domain model produces domain independence at the expense of the designer's ability to control the direction of the interactions. Initial experiments with DENISE tend to support the basic premise that these systems can acquire individualised models from users.

The discussion of the DENISE experiments can be examined in the context of four design issues which are important for the design of ISSs. These four areas cover: the learnt model, control of the interaction, domain knowledge and conceptual syntax.

Learnt Model

The learnt model in DENISE was empty at the beginning of each subject's session. This means that there are no initial reference points for the dialogue strategy and the system has no curriculum to follow. Learners can 'drift' around the domain (if they wish) as they retain overall control of the dialogue. The initially empty learnt model contrasts sharply with peer tutoring studies as real tutees have considerable knowledge about the

domain. A possible alternative is to 'prime' the learnt model with some initial beliefs to more precisely locate the dialogue in a particular domain.

Subjects found it difficult to remember which concepts and relationships they had previously used and frequently referred to the dictionary mechanism. This appears to be because the only external reflection of the system's learning came in the content of DENISE questions. This was particularly noticeable when the system produced a default non-committal response. In the [Michie, Paterson & Hayes-Michie 1989] system users could test the effectiveness of the currently induced rule against equations thus providing an indication of the state of the learnt model. In a declarative domain it may be that some sort of assessment or visualisation of the system's learning is needed to maintain focus in a mixed-initiative dialogue. A graphical representation such as presented in the SemNet system [Fisher, 1992] may be appropriate.

The notion of 'drifting' out of a domain is a value judgement; if it is a realistic representation of the subject's conception of the domain in question then it is impossible for the subject to be 'out' of their own domain. When a system sets out to explore a *subject's* viewpoint of a domain it seems unreasonable to object if it is different to someone else's viewpoint.

Control of the Interaction

A human tutor has both domain and pedagogical knowledge with which to control the direction of a dialogue with a student. When a learner is placed in the role of a tutor there is considerably less control knowledge and the dialogue can easily 'drift' across the domain. In order to prevent this drift the ISS must take greater responsibility for the direction of the interaction than a human tutee.

Task-based interaction, such as most peer tutoring and the [Michie, Paterson & Hayes-Michie 1989] system's equation-solving, provides an implicit control of the interaction as there are clear goals and sub-goals to be attained. This task-based control allowed the system to function with a minimal dialogue strategy which left learners in complete control. Several subjects expressed a preference for explicit direction in the interactions with DENISE rather than an open-ended dialogue.

In a declarative domain implicit task-based control is easily lost and the ISS needs to provide alternatives if the system is to be used for specific pedagogical objectives. It may be that an explicit curriculum or domain

knowledge (either a 'primed' learnt model or a conventional domain model) needs to be added to the ISS architecture for declarative domains.

A related issue is which partner, if any, should have the final say about the direction of a dialogue. If the system wishes to explore a new viewpoint and the learner does not then how is the interaction to continue. In classrooms the teacher's view usually prevails whereas many computer systems control the dialogue. The sharing of control when traditional roles are reversed is a complicated area and is a candidate as a good reason to include non-domain related information in an ISS.

Domain Knowledge

DENISE functions without any prior knowledge of economics; other than the knowledge encoded in the conceptual syntax. Thus DENISE currently represents one extreme along a spectrum of possible ISS configurations – a dialogue strategy which does not refer to a domain model. At the other end is an ISS with a full domain model in the classical ITS tradition. Here the dialogue strategy can compare user statements with some 'correct' model. In such a situation the ISS is deceiving the learner – externally appearing to be a novice whilst internally having the knowledge of an expert. Such a system would go beyond a straightforward analogue of peer tutoring to move towards an 'optimal' student.

Conceptual Syntax

The conceptual syntax (system and user) in DENISE is fixed at compile-time and cannot be changed by the learner or the system. The constrained nature of the DENISE system meant that some subjects wished to express relationships that went beyond the qualitative causal links in the system.

There are three complementary approaches to producing a system with a more complicated conceptual syntax. Firstly, simply use a richer conceptual syntax; this could include structural knowledge, order of magnitude reasoning or knowledge about the temporal aspects of relationships. Secondly, the initial user conceptual syntax could be added to during the interaction – either by the system or a selection by the user. This could be achieved through a differential model – with a more complex system conceptual syntax gradually being added to the user conceptual syntax. Thirdly, allow the user to modify the conceptual syntax. This solution produces an adaptable system but requires that the dialogue strategy is constructed on the 'shifting foundations' of a changing

conceptual syntax. This may be technically possible but greatly increases the difficulty of authoring the dialogue strategy.

Contributions

This thesis brings together a variety of work from different fields to delineate a framework for ISSs.

An ISS and associated authoring facilities have been designed, implemented and tested in the domain of economics. The claim to originality is simply that this thesis represents the first sizable piece of work on these systems.

The issues highlighted by this research are:

- the rationale for investigating ISSs
- the components necessary for their use
- ISSs without domain models are feasible
- ISSs are capable of forcing learners to examine their own beliefs
- ISS have significant cost and complexity advantages over other forms of ILE
- it is difficult to restrict or plan interactions when the learner has final control of the dialogue

The technical aspects of reasoning with viewpoints have been emphasized at the expense of introducing them into dialogues as explicit objects that are manipulable by the learner. Viewpoints should migrate from the inference engines of ILEs to the interface, where the learner can experience them directly.

In addition to the relative computational simplicity of ISS compared to ITSs they also have a possible organizational advantage. In placing an ITS in a school classroom the existing teacher may well feel threatened by the 'machine that is replacing her'. An ISS is in no sense 'de-skilling', as it is relatively ignorant. ISSs therefore may be appropriate Trojan horses for the dissemination of knowledge-based software into the education system.

In sum, ISSs are novel educational systems with the potential to bring a new form of interaction into the ILE field.

8.2 Future Work

There is a large amount of work to be done on refining and testing ISSs to discover if their potential can be realised. The central goal must be

more studies and more empirical data from different domains with different conceptual syntaxes.

The research questions listed in section 4.5 are a starting point to delimiting the future work in this area. Significantly, some of these questions relate to peer tutoring research rather than the development of computer systems for educational applications.

However, it is not necessary for peer tutoring research to produce clear evidence that the learning by teaching paradigm is a successful technique. ISSs are something new; approaches which do not work with human-human interactions may still function with human-computer interactions. The concentration of peer tutoring research on matching together tutors and tutees suggests that an appropriately designed adaptable ISS could be more successful than a human tutee.

Some of the central issues of the ISS concept that deserve attention are:

- is domain knowledge needed?

The role of domain knowledge in the ISS may well determine its success. If ISSs, in practice, require similar domain models to ITSs then they will be subject to the same inflexibility and brittleness. If ISSs can function with little domain knowledge (as DENISE does) then they will have significant advantages over ITSs and could become widespread educational tools.

- the possibilities of a task-based ISS where the system has a problem to solve so that the learner has an explicit goal.

The 'drift' around the domain that users of DENISE experienced needs to be addressed if ISSs are to be used in real-world educational settings. This can be addressed in two main ways: a more directive dialogue strategy and a visualization of the learnt model. The impact of providing a graphical representation of the learnt model during the interaction is clearly an important area of research.

- how should ISSs learn to aid their users?

The mechanism that the ISS uses to learn from the user can take one of two main forms. A mechanical algorithm, as in DENISE, or a model of human learning. Our expectation is that a cognitively accurate computer learner is not necessary but this needs to be examined using controlled experiments.

- can ISSs be used as a tool for assessment?

Assessing procedural knowledge is straightforward; to determine whether some agent knows how to do task *t*, simply present the agent with task *t* and observe. If the agent can repeatedly perform

similar tasks to t then it is reasonable to conclude that the agent does indeed possess the appropriate procedural knowledge.

Assessing declarative knowledge is more complicated. Conventional education relies on essays and multiple choice questions or tries to proceduralize the assessment in, for example, physics problems. The work presented here suggests that a possible assessment method for declarative knowledge can be generated through providing a concrete teaching task for a learner.

8.3 Endpiece: A One Sentence Summary of the Thesis

So, encapsulating the central theme of this thesis in one sentence:

The task of teaching an agent is an effective means of operationalizing learners' knowledge for educational benefit.

or

You don't really know what you know, or know whether what you think you know is the same as what you really know, until you can teach what you know to some agent that doesn't know it – but is capable of learning it.

References

Acker, L., Lester, J., Souther, A. and Porter, B. (1991), Generating coherent explanations to answer students' questions, in Burns, H., Parlett, J.W. and Redfield, C.L. (Eds.), *Intelligent Tutoring Systems: Evolutions in Design*, Hillsdale, NJ: Lawrence Erlbaum, 151-176.

Addanki, S., Cremonini, R. and Penberthy, J.S. (1991), Graphs of models, *Artificial Intelligence*, 51(1-3), 145-177.

Anderson, J.R. (1988), The expert module, in Polson, M.C. and Richardson, J.J. (Eds.), *Foundations of Intelligent Tutoring Systems*, Hillsdale, NJ: Lawrence Erlbaum, 21-54.

Attardi, G. and Simi, M. (1984), Metalanguage and reasoning across viewpoints, *European Conference on Artificial Intelligence*, Pisa, Italy, Elsevier Science, 413-422.

Baker, E. (1991), Technology assessment: policy and methodological issues, in Burns, H., Parlett, J.W. and Redfield, C.L. (Eds.), *Intelligent Tutoring Systems: Evolutions in Design*, Hillsdale, NJ: Lawrence Erlbaum.

Ballim, A. and Wilks, Y. (1991), *Artificial Believers: the Ascription of Belief*, Hillsdale, NJ: Lawrence Erlbaum.

Ballim, A., Wilks, Y. and Barnden, J. (1991), Belief ascription, metaphor and intensional identification, *Cognitive Science*, 15(1), 133-171.

Berliner, D. (1985), Laboratory settings and the study of teacher education, *Journal of Teacher Education*, 36(6), 2-8.

Berliner, D. (1989), Being the teacher helps students learn, *Instructor*, 98(9), 12-3.

Berndsen, R. and Daniels, H. (1989), Sequential causality and qualitative reasoning in economics, in Pau, L.F. (Ed.), *Expert Systems in Economics, Banking and Management*, Elsevier Science, 125-135.

Berndsen, R. and Daniels, H. (1991), Qualitative economics: an implementation in PROLOG, *Computer Science in Economics and Management*, 4(1), 1-13.

Bernsden, R. and Daniels, H. (1990), Qualitative dynamics and causality in a Keynesian model, *Journal of Economic Dynamics and Control*, 14(2), 435-450.

Black, J.H., Turner, T.J. and Bower, G.H. (1979), Point of view in narrative comprehension, memory, and production, *Journal of Verbal Learning and Verbal Behaviour*, 18(2), 187-189.

Bloom, B.S. (1984), The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring, *Educational Researcher*, 13(6), 4-16.

Bobrow, D.G. and Winograd, T. (1977), An overview of KRL, a knowledge representation language, *Cognitive Science*, 1(1), 3-46.

Bonar, J.G. (1991), Interface architectures for intelligent tutoring systems, in Burns, H., Parlett, J.W. and Redfield, C.L. (Eds.), *Intelligent Tutoring Systems: Evolutions in Design*, Hillsdale, NJ: Lawrence Erlbaum Associates, 35-67.

Bonar, J.G. and Cunningham, R. (1988), Bridge: tutoring the programming process, in Psozka, J., Massey, L.D. and Mutter, S.A. (Eds.), *Intelligent Tutoring Systems: lessons learned*, Hillsdale, N.J.: Lawrence Erlbaum Associates, 409-434.

Bourgine, P. and Raiman, O. (1986), Economics as reasoning on a qualitative model, *1st International Conference on Economics and Artificial Intelligence*, Aix-en-Provence, France., Pergamon Press, 121-125.

Brachman, R. and Schmolze, J. (1985), An overview of the KL-ONE knowledge representation system, *Cognitive Science*, 9(2), 171-216.

Brecht, B. and Jones, M. (1988), Student models: the genetic graph approach, *International Journal of Man-Machine Studies*, 28(5), 483-504.

Breece, J.H. (1988), Computerized economic modelling: from laboratory to classroom, *Social Science Computer Review*, 6(4), 498-514.

Brown, J.S. and Burton, R.R. (1975), Multiple representations of knowledge for tutorial reasoning, in Bobrow, D.G. and Collins, A. (Eds.), *Representation and Understanding*, New York: Academic Press, 311-349.

Brown, J.S., Burton, R.R. and Bell, A.G. (1975), SOPHIE: a step towards creating a reactive learning environment, *International Journal of Man-Machine Studies*, 7(5), 675-696.

Brown, J.S., Burton, R.R. and Kleer, J.d. (1982), Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III, in Sleeman, D. and Brown, J.S. (Eds.), *Intelligent Tutoring Systems*, London: Academic Press, 227-282.

Brown, J.S., Burton, R.R. and Zydel, F. (1973), A model-driven questioning-answering system for mixed-initiative computer-assisted instruction, *IEEE Transactions on Systems, Man and Cybernetics*, 3(3), 248-257.

Burns, H.L. and Capps, C.G. (1988), Foundations of intelligent tutoring systems: an introduction, in Polson, M.C. and Richardson, J.J. (Eds.), *Foundations of Intelligent Tutoring Systems*, Hillsdale, NJ: Lawrence Erlbaum, 1-19.

Burton, R.R. (1982), Diagnosing bugs in a simple procedural skill, in Sleeman, D. and Brown, J.S. (Eds.), *Intelligent Tutoring Systems*, London: Academic Press, 157-183.

Burton, R.R. and Brown, J.S. (1979), Toward a natural language capability for computer-assisted instruction, in O'Neil, H.F. (Eds.), *Procedures for Instructional Systems Development*, London: Academic Press, 273-313.

Burton, R.R. and Brown, J.S. (1982), An investigation of computer coaching for informal learning activities, in Sleeman, D. and Brown, J.S. (Eds.), *Intelligent Tutoring Systems*, London: Academic Press, 79-98.

Carbonell, J.R. (1970), AI in CAI: an artificial intelligence approach to computer-assisted instruction, *IEEE Transactions on Man-Machine Systems*, 11(4), 190-202.

Chan, T.-W. and Baskin, A.B. (1988), "Studying with the prince": the computer as a learning companion, *ITS-88*, Montreal, Canada, 194-200.

Chan, T.-W., Chung, I.-L., Ho, R.-G., Hou, W.-J. and Lin, G.-L. (1992), Distributed learning companion system: WEST revisited, *Second International Conference on Intelligent Tutoring Systems*, Montreal, Canada, Springer-Verlag, 643-650.

Chan, T.W. and Baskin, A.B. (1990), Learning companion systems, in Frasson, C. and Gauthier, G. (Eds.), *Intelligent Tutoring Systems: at the Crossroads of Artificial Intelligence and Education*, Norwood, N.J.: Ablex, 6-33.

Chi, M.T.H., Bassok, M., Lewis, M.W., Reimann, P. and Glaser, R. (1989), Self-explanations: how students study and use examples in learning to solve problems, *Cognitive Science*, 13(2), 145-182.

Clancey, W.J. (1983), The epistemology of a rule-based expert system - a framework for explanation, *Artificial Intelligence*, 20(3), 215-251.

Clancey, W.J. (1987), *Knowledge-Based Tutoring: The GUIDON Program*, Cambridge, MA: The MIT Press.

Cohn, A.G. (1989), Approaches to qualitative reasoning, *Artificial Intelligence Review*, 3(2-3), 177-232.

Collins, A. (1977), Processes in acquiring knowledge, in Anderson, R.C., Spiro, R.J. and Montague, W.E. (Eds.), *Schooling and the Acquisition of Knowledge*, Hillsdale, N.J.: Lawrence Erlbaum, 339-363.

Collins, A. (1985), Component models of physical systems, *7th Annual Conference of the Cognitive Science Society*, 80-89.

Collins, A. and Brown, J.S. (1988), The computer as a tool for learning through reflection, in Mandl, H. and Lesgold, A. (Eds.), *Learning Issues for Intelligent Tutoring Systems*, New York: Springer-Verlag, 1-18.

Collins, A. and Gentner, D. (1983), Multiple models of evaporation processes, *5th Annual Conference of the Cognitive Science Society*, 1-5.

Collins, A. and Stevens, A.L. (1991), A cognitive theory of inquiry teaching, in Goodyear, P. (Ed.), *Teaching Knowledge and Intelligent Tutoring*, Norwood, NJ: Ablex, 203-230.

Console, L. and Torasso, P. (1990), Hypothetical reasoning in causal models, *International Journal of Intelligent Systems*, 5(1), 83-124.

Corte, E.D., Verschaffel, L. and Schrooten, H. (1991), Computer simulation as a tool in studying teachers' cognitive activities during error diagnosis in arithmetic, in Goodyear, P. (Ed.), *Teaching Knowledge and Intelligent Tutoring*, Norwood, N.J.: Ablex, 367-378.

Costa, E., Duchenois, S. and Kodratoff, Y. (1988), A resolution based method for discovering students' misconceptions, in Self, J. (Ed.), *Artificial Intelligence and Human Learning: Intelligent Computer-Aided Instruction*, London: Chapman and Hall, 156-164.

Cumming, G. and Self, J. (1989), Collaborative intelligent educational systems, *4th International Conference on AI and Education*, Amsterdam, Holland, IOS, 73-80.

Date, C.J. (1990), *An Introduction to Database Systems*, Addison-Wesley.

deKleer, J. and Brown, J.S. (1983), Assumptions and ambiguities in mechanistic mental models, in Gentner, D. and Stevens, A.L. (Eds.), *Mental Models*, Hillsdale, N.J.: Lawrence Erlbaum Associates, 155-190.

Dillenbourg, P. (1991), *Human-Computer Collaborative Learning*, Unpublished PhD thesis, Computing Department, Lancaster University, U.K.

Dillenbourg, P. and Self, J. (1990), *A Framework for Learner Modelling*, AI Technical Report 49, Lancaster University. 7/90.

Dillenbourg, P. and Self, J.A. (1992), PEOPLE POWER: A human-computer collaborative learning system, *Second International Conference on Intelligent Tutoring Systems*, Montreal, Canada, Springer-Verlag, 651-660.

Dinsmore, J. (1991), *Partitioned Representations*, Dordrecht, Netherlands: Kluwer Academic.

diSessa, A. (1986), Models of computation, in Norman, D.A. and Draper, S.W. (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction*, Hillsdale, NJ: Lawrence Erlbaum, 201-218.

Dornbusch, R. and Fischer, S. (1990), *Macroeconomics*, (5th Edition), New York: McGraw-Hill.

Douglas, S.A. (1991), Tutoring as interaction: detecting and repairing tutoring failures, in Goodyear, P. (Ed.), *Teaching Knowledge and Intelligent Tutoring*, Norwood, N.J.: Ablex, 123-147.

Elmasri, R. and Navathe, S.B. (1989), *Fundamentals of Database Systems*, Redwood City, CA: Benjamin/Cummings.

Elsom-Cook, M. and Spensley, F. (1990), A tutor for procedural skills, in Elsom-Cook, M. (Ed.), *Guided Discovery Tutoring: A Framework for ICAI Research*, London: Paul Chapman, 57-69.

Emde, W. (1987), An inference engine for representing multiple theories, in Morik, K. (Ed.), *Knowledge Representation and Organization in Machine Learning*, Berlin: Springer-Verlag, 148-176.

Eshelman, L. (1988), MOLE: A knowledge-acquisition tool for cover-and-differentiate systems, in Marcus, S. (Ed.), *Automating Knowledge Acquisition for Expert Systems*, London: Kluwer Academic, 37-80.

Fagin, R. and Halpern, J.Y. (1988), Belief, awareness and limited reasoning, *Artificial Intelligence*, 34(1), 39-76.

Falkenhainer, B. and Forbus, K.D. (1991), Compositional modeling: finding the right model for the job, *Artificial Intelligence*, 51(1-3), 95-143.

Farley, A.M. (1986), Qualitative modelling of economic systems, *1st International Conference on Economics and Artificial Intelligence*, Aix-en-Provence, France., Pergamon Press, 41-44.

Farley, A.M. and Lin, K.-P. (1990), Qualitative reasoning in economics, *Journal of Economic Dynamics and Control*, 14(2), 465-490.

Farley, A.M. and Lin, K.-P. (1991), Qualitative reasoning in microeconomics: an example, *Decision Support Systems and Qualitative Reasoning: Proceedings of the IMACS International Workshop*, Toulouse, France, North-Holland, 303-306.

Fauconnier, G. (1985), *Mental Spaces: Aspects of Meaning Construction in Natural Language*, London: The MIT Press.

Ferguson-Hessler, M. and Jong, T.d. (1990), Studying physics tests: differences in study processes between good and poor performers, *Cognition and Instruction*, 7(1), 41-54.

Filman, R.E. (1988), Reasoning with worlds and truth maintenance in a knowledge-based programming environment, *Communications of the ACM*, 31(4), 382-401.

Finch, I. (1988), Explanation and viewpoints, *The 4th Alvey Explanation Workshop*, Manchester, UK, Alvey Knowledge Based Systems Club, 227-241.

Findlay, W. and Watt, D.A. (1985), *Pascal: an introduction to methodical programming*, London: Pitman.

Fink, P.K. and Luth, J.C. (1987), Expert systems and diagnostic expertise in the mechanical and electrical domains, *IEEE Transactions on Systems, Man, and Cybernetics*, 17(3), 340-349.

Fisher, K. (1992), SemNet: a tool for personal knowledge construction, in Kommers, P.A.M., Jonassen, D.H. and Mayes, J.T. (Eds.), *Cognitive Tools for Learning*, Berlin: Springer-Verlag, 63-75.

Forbus, K.D. (1984), Qualitative process theory, *Artificial Intelligence*, 24(1-3), 85-168.

Foss, C.L. (1987), *Learning from errors in AlgebraLand*, Technical Report IRL-87-0003 Institute for Research on Learning, Palo Alto.

Fresko, B. and Chen, M. (1989), Ethnic similarity, tutor expertise and tutor satisfaction in cross-age tutoring, *American Educational Research Journal*, 26(1), 122-140.

Gardenfors, P. (1988), *Knowledge in Flux: Modeling the Dynamics of Epistemic States*, London: MIT Press (Bradford Book).

Gasarch, W.I. and Smith, C.H. (1992), Learning via queries, *Journal of the ACM*, 39(3), 649-674.

Gentner, D. and Gentner, D.R. (1983), Flowing waters or teeming crowds: mental models of electricity, in Gentner, D. and Stevens, A.L. (Eds.), Hillsdale, N.J.: Lawrence Erlbaum Associates, 155-170.

Glachan, M. and Light, P. (1982), Peer interaction and learning: can two wrongs make a right?, in Butterworth, G. and Light, P. (Eds.), *Social Cognition: studies of the development of understanding*, Brighton, UK: Harvester Press,

Goldenson, R.M. (Ed.) (1984), *Longman Dictionary of Psychology and Psychiatry*, White Plains, N.Y.: Longman.

Goldstein, I.P. (1982), The genetic graph: a representation for the evolution of procedural knowledge, in Sleeman, D. and Brown, J.S. (Eds.), *Intelligent Tutoring Systems*, London: Academic Press, 51-77.

Goodlad, S. (1985), Putting science in context, *Educational Research*, 27(1), 61-67.

Goodlad, S. and Hirst, B. (1989), *Peer Tutoring: a Guide to Learning by Teaching*, London: Kogan Page.

Goodlad, S. and Hirst, B. (Eds.) (1990a), *Explorations in Peer Tutoring*, Oxford, UK: Blackwell.

Goodlad, S. and Hirst, B. (1990b), Explorations in peer tutoring, in Goodlad, S. and Hirst, B. (Eds.), *Explorations in Peer Tutoring*, Oxford, UK: Blackwell, 1-25.

Goodyear, P. and Stone, C. (1992), Domain knowledge, epistemology and intelligent tutoring in social science, in Moyse, R. and Elsom-Cook, M. (Eds.), *Knowledge Negotiation*, London: Academic Press, 69-95.

Greer, J. and McCalla, G. (1989), A computational framework for granularity and its application to educational diagnosis, *Eleventh International Joint Conference on Artificial Intelligence*, Detroit, Michigan, Morgan Kaufmann, 477-482.

Grice, H.P. (1968), Utterer's meaning , sentence meaning and word meaning, *Foundations of Language*, 4, 225-42.

Gudgin, A.N. (1987), Computer-assisted learning of introductory economics, *British Journal of Educational Technology*, 18(2), 84-93.

Guha, R.V. and Lenat, D.B. (1990), Cyc: a midterm report, *AI Magazine*, 11(3), 32-59.

Habeshaw, S., Habeshaw, T. and Gibbs, G. (1984), *53 Interesting Things to do in Seminars and Tutorials*, Technical and Educational Services.

Haddley, N. and Sommerville, I. (1990), Integrated support for systems design, *Software Engineering Journal*,

Halff, H.M. (1988), Curriculum and instruction in intelligent tutors, in Polson, M.C. and Richardson, J.J. (Eds.), *Foundations of Intelligent Tutoring Systems*, Hillsdale, N.J.: Lawrence Erlbaum Associates, 79-108.

Hayes-Roth, B. (1985), A blackboard architecture for control, *Artificial Intelligence Journal*, 26(3), 251-321.

Hendrix, G.G. (1979), Encoding knowledge in partitioned networks, in Findler, N.V. (Ed.), *Associative Networks: Representation and Use of Knowledge by Computers*, London: Academic Press, 51-92.

Hewson, P.W. and Hewson, M.G.A. (1984), The role of conceptual conflict in conceptual change and the design of science instruction, *Instructional Science*, 13(1), 1-13.

Hicks, J. (1979), *Causality in economics*, Oxford, UK: Blackwell.

Hobbs, P. and Judge, G. (1992), Computers as a tool for teaching economics, *Computers and Education*, 19(1/2), 67-72.

Hollan, J.D., Hutchins, E.L. and Weitzman, L. (1984), STEAMER: an interactive inspectable simulation-based training system, *AI Magazine*, 5(4), 15-27.

Horan, J.J., Girolomo, M.A.d., Hill, R.L. and Shute, R.E. (1974), The effect of older-peer participant models on deficient academic performance, *Psychology in the Schools*, 11(2), 201-12.

Huxor, A. (1988), *Explanation and Understanding: An Architecture for Economic Elaboration*, Technical Report TR-LP-26, European Computer-Industry Research Centre GmbH. 1/ 88.

Johnson-Laird, P.N. and Byrne, R.M.J. (1991), *Deduction*, Hillsdale, NJ: Lawrence Erlbaum.

Kahn, G., Nowlan, S. and McDermott, J. (1985), Strategies for knowledge acquisition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(5), 571-522.

Kaufmann, H. and Grumbach, A. (1986), MULTILog: Multiple worlds in logic programming, *Seventh European Conference on Artificial Intelligence*, Brighton, UK, 291-305.

Kennedy, M. (1990), Controlled evaluation of the effects of peer tutoring: are the 'learning by teaching' theories viable?, in Goodlad, S. and Hirst, B. (Eds.), *Explorations in peer tutoring*, Oxford, UK: Blackwell, 58-72.

Khuwaja, R.A., Evans, M.W., Rovick, A.A. and Michael, J.A. (1992), Knowledge representation for an intelligent tutoring system based on a multilevel causal model, *Second International Conference on Intelligent Tutoring Systems*, Montreal, Canada, Springer-Verlag, 217-224.

Kieras, D.E. and Bovair, S. (1984), The role of a mental model in learning to operate a device, *Cognitive Science*, 8(3), 255-273.

Kim, W. (1990), *Introduction to Object-Oriented Databases*, Cambridge, MA: MIT Press.

Kimball, R. (1982), A self-improving tutor for symbolic integration, in Sleeman, D. and Brown, J.S. (Eds.), *Intelligent Tutoring Systems*, London: Academic Press,

Kuipers, B. (1986), Qualitative simulation, *Artificial Intelligence*, 29(3), 289-338.

Kumata, Y. and Atsumi, M. (1988), A design of viewpoints-directed knowledge system for decision support, in Kovacs, P. and Straub, E. (Eds.), *Governmental and Municipal Information Systems*, Elsevier Science, 281-298.

Lancaster-University (1990), *Department of Economics, Examinations*,

Langley, P., Wogulis, J. and Ohlsson, S. (1987), Rules and principles in cognitive diagnosis, in Fredericksen, N. (Ed.), *Diagnostic Monitoring of Skill and Knowledge Acquisition*, Hillsdale, NJ: Lawrence Erlbaum, 217-50.

Legree, P.J., Gillis, P.D. and Orey, M.A. (1993), The quantitative evaluation of intelligent tutoring system applications: product and process criteria, *Journal of Artificial Intelligence in Education*, 4(2/3), 209-226.

Letovsky, S. (1983), *Qualitative and Quantative Temporal Reasoning*, Technical Report DCS/RR*296, Yale University.

Levesque, H. (1984), A logic of implicit and explicit belief, *Proceedings of the National Conference on Artificial Intelligence*, Los Altos: Morgan Kaufmann,

Light, P. and Glachan, M. (1985), Facilitation of individual problem solving through peer interaction, *Educational Psychology*, 5(3/4), 217-225.

Littman, D. and Soloway, E. (1988), Evaluating ITSs: the cognitive science perspective, in Polson, M.C. and Richardson, J.J. (Eds.), *Foundations of Intelligent Tutoring Systems*, Hillsdale, N.J.: Lawrence Erlbaum Associates, 209-242.

London, R.V. (1991), *Student Modeling with Multiple Viewpoints by Plan Recognition*, Unpublished PhD thesis, Stanford University.

London, R.V. (1992), Student modeling to support multiple instructional approaches, *User Modeling and User-Adapted Interaction*, 2(1-2), 117-154.

Lumsden, K. and Scott, A. (1987), Teaching macroeconomics by simulations, in Kent, W.A. and Lewis, R. (Eds.), *Computer Assisted Learning in the Humanities and Social Sciences*, Oxford: Blackwell Scientific Publications, 73-79.

MacDonald, B.A. (1991), Instructable systems, *Knowledge Acquisition*, 3(4), 381-420.

Mado, S. and Sawa, T. (1989), Expert system for qualitative causal inference in economics, *Dynamic Modelling and Control of National Economies 1989*, Edinburgh, UK, Pergamon, 529-531.

Marcus, S. (Ed.) (1988), *Automating Knowledge Acquisition for Expert Systems*, London: Kluwer Academic.

Margenu, H. (1966), What is a theory?, in Krupp, S.R. (Ed.), *The Structure of Economic Science: essays on methodology*, Englewood Cliffs, NJ: Prentice-Hall, 25-38.

Mariani, J.A. (1992) "Realising Relational-style Operators and Views in the Oggetto OODBMS", Internal Document, Computing Department, Lancaster University:

Martins, J.P. and Shapiro, S.C. (1983), Reasoning in multiple belief spaces, *Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, 370-373.

Martins, J.P. and Shapiro, S.C. (1988), A model for belief revision, *Artificial Intelligence*, 35(1), 25-79.

McCain, K.W. (1983), The author cocitation structure of macroeconomics, *Scientometrics*, 5, 277-289.

McCain, K.W. (1986), Cocited author mapping as a valid representation of intellectual structure, *Journal of the American Society for Information Science*, 37(3), 111-22.

McCain, K.W. (1990), Mapping authors in intellectual space: a technical review, *Journal of the American Society for Information Science*, 41(6), 433-443.

Micarelli, A., Mungo, F., Nucci, F. and Aiello, L. (1991), SAMPLE: An intelligent educational system for electrical circuits, *Journal of Artificial Intelligence in Education*, 2(3), 83-99.

Micarelli, A., Mungo, F., Nucci, F.S. and Aiello, L. (1992), Reasoning with worlds and truth maintenance in an intelligent tutoring system, *Expert Systems with Applications*, 4(1), 147-155.

Michie, D., Paterson, A. and Hayes-Michie, J. (1989), Learning by teaching, *2nd Scandinavian Conference on Artificial Intelligence '89*, Tampere, Finland, IOS, 307-331.

Millerd, F.W. and Robertson, A.R. (1987), Computer simulations as an integral part of intermediate macroeconomics, *Journal of Economic Education*, 18(2), 269-286.

Minsky, M. (1985), A framework for representing knowledge, in Brachman, R.J. and Levesque, H.J. (Eds.), *Readings in Knowledge Representation*, San Mateo, CA: Morgan Kaufmann, 246-262.

Moyse, R. (1988), *Multiple Viewpoints for Intelligent Tutoring Systems*, CITE report 39, Institute of Educational Technology, Open University. 5/88.

Moyse, R. (1990), *Multiple Viewpoints for Intelligent Tutoring Systems*, Unpublished PhD thesis, The Institute of Educational Technology, The Open University, U.K.

Moyse, R. (1991), Multiple viewpoints imply knowledge negotiation, *Interactive Learning International*, 7(1), 21-37.

Moyse, R. (1992), A structure and design method for multiple viewpoints, *Journal of Artificial Intelligence in Education*, 3(2), 207-233.

Moyse, R. and Elsom-Cook, M. (Eds.) (1992a), *Knowledge Negotiation*, London: Academic Press.

Moyse, R. and Elsom-Cook, M. (1992b), Knowledge negotiation: an introduction, in Moyse, R. and Elsom-Cook, M. (Eds.), *Knowledge Negotiation*, London: Academic Press, 1-19.

Mugny, G. and Doise, W. (1978), Individual and collective conflicts of centrations in cognitive development, *European Journal of Social Psychology*, 9(1), 105-9.

Murray, T. (1993), Formative qualitative evaluation for exploratory research, *Journal of Artificial Intelligence and Education*, 4(2/3), 179-207.

Nathan, M.J., Johl, P., Kintsch, W. and Lewis, C. (1989), An unintelligent tutoring system for solving word algebra problems, *4th International Conference on AI and Education*, Amsterdam, Holland, IOS, 169-176.

Newell, A. (1982), The knowledge level, *Artificial Intelligence*, 18(1), 87-127.

Nussbaum, J. and Novick, S. (1982), Alternative frameworks, conceptual conflict and accommodation: toward a principled teaching strategy, *Instructional Science*, 11(3), 183-200.

Nwana, H.S. (1990), Intelligent tutoring systems: an overview, *Artificial Intelligence Review*, 4(4), 251-277.

Ohlsson, S. (1986), Some principles of intelligent tutoring, *Instructional Science*, 14(3-4), 293-326.

Ohlsson, S. (1992), Artificial instruction: a method for relating learning theory to instructional design, in Jones, M. and Winne, P.H. (Eds.), *Adaptive Learning Environments: Foundations and Frontiers*, New York: Springer-Verlag, 55-83.

Ohlsson, S., Ernst, A.M. and Rees, E. (1992), The cognitive complexity of learning and doing arithmetic, *Journal for Research in Mathematics Education*, 23(5), 441-467.

Palthepeu, S., Greer, J. and McCalla, G. (1991), Learning by teaching, *The International Conference on the Learning Sciences*, Evanston, Illinois, AACE, 357-363.

Park, O.-C., Perez, R.S. and Seidel, R.J. (1987), Intelligent CAI: old wine in new bottles, or a new vintage?, in Kearsley, G. (Ed.), *Artificial Intelligence and Instruction: Applications and Methods*, Addison-Wesley, 11-45.

Pau, L.F. (1984), Inference of the structure of economic reasoning from natural language analysis, *Dynamic Modelling and Control of National Economies 1983*, Pergamon Press, 157-163.

Pau, L.F. (1986), Inference of functional economic model relations from natural language analysis, in Pau, L.F. (Ed.), *Artificial Intelligence in Economics and Management*, Elsevier Science, 173-183.

Peachey, D.R. and McCalla, G.I. (1986), Using planning techniques in intelligent tutoring systems, *International Journal of Man-Machine Studies*, 24(1), 77-98.

Petrie-Brown, A. (1989), Intelligent tutoring dialogue: the structures of an interaction, *4th International Conference on AI and Education*, Amsterdam, Holland, IOS, 195-201.

Petrie-Brown, A. (1990), Discourse and dialogue: concepts in intelligent tutoring interactions, *Journal of Artificial Intelligence in Education*, 1(2), 21-29.

Pirolli, P. and Bielaczyc, K. (1989), Empirical analyses of self-explanation and transfer in learning to program, *Eleventh Conference of the Cognitive Science Society*, Ann Arbor, Michigan, Lawrence Erlbaum,

Quinlan, J.R. (1986), Induction of decision trees, *Machine Learning*, 1(1), 81-106.

Rogoff, B. (1990), *Apprenticeship in Thinking: cognitive development in social context*, New York, N.Y.: Oxford University Press.

Ross, P. (1987), Intelligent tutoring systems, *Journal of Computer Assisted Learning*, 3(1), 194-203.

Rudduck, J. (1978), *Learning Through Small Group Discussion: a study of seminar work in higher education*, Guildford, UK: Society for Research into Higher Education.

Sandberg, J., Barnard, Y. and van-der-Hulst, A. (1992), Interviews on AI and education: Kurt VanLehn and David Merrill, *AI Communications*, 5(2), 85-91.

Saunders, D. (1992), Peer tutoring in higher education, *Studies in Higher Education*, 17(2), 211-218.

Scanlon, E. and O'Shea, T. (1988), Cognitive economy in physics reasoning: implications for designing instructional materials, in Mandl, H. and Lesgold, A. (Eds.), *Learning Issues for Intelligent Tutoring Systems*, New York: Springer-Verlag, 258-277.

Schiff, J. and Kandler, J. (1988), DecisionLab: a system designed for user coaching in managerial decision support, *ITS-88*, Montreal, Canada, 154-161.

Self, J. (1988a), Bypassing the intractable problem of student modelling, *ITS-88*, Montreal, Canada, 18-24.

Self, J. (1988b), Knowledge, belief and user modelling, in O'Shea, T. and Sgurev, V. (Eds.), *Artificial Intelligence III: Methodology, Systems, Applications*, Elsevier Science, North-Holland, 3-9.

Self, J. (1992), Computational viewpoints, in Moyse, R. and Elsom-Cook, M. (Eds.), *Knowledge Negotiation*, London: Academic Press, 21-40.

Selfridge, M., Daniell, J. and Simmons, D. (1985), Learning causal models by understanding real-world natural language explanations, *Applications of AI*, IEEE.

Shapiro, S.C. (1979), The SNepS semantic network processing system, in Findler, N.V. (Ed.), *Associative Networks: Representation and Use of Knowledge by Computers*, London: Academic Press, 179-203.

Shelley, A. and Sibert, E. (1991), Computer simulation in teacher education: enhancing teacher decision making, in Goodyear, P. (Ed.), *Teaching Knowledge and Intelligent Tutoring*, Norwood, N.J.: Ablex, 341-365.

Shute, V. (1993), An experiential approach to teaching and learning probability: Stat Lady, *World Conference on Artificial Intelligence in Education*, Edinburgh, Scotland, AACE, 177-84.

Shute, V. and Regian, J.W. (1993), Principles for evaluating intelligent tutoring systems, *Journal of Artificial Education*, 4(2/3), 245-271.

Shute, V.J. (1990), Rose Garden Promises of Intelligent Tutoring Systems: Blossom or Thorn?, Presented at *Space Operations, Applications and Research Symposium*, Albuquerque, NM,

Shute, V.J. and Glaser, R. (1990), A large scale evaluation of an intelligent discovery world: Smithtown, *Interactive Learning Environments*, 1(1), 51-77.

Sime, J.-A. and Leitch, R. (1992), A learning environment based on multiple qualitative models, *Second International Conference on Intelligent Tutoring Systems*, Montreal, Canada, Springer-Verlag, 116-123.

Simon, H.A. (1983), Why should machines learn?, in Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. (Eds.), *Machine Learning I*, Los Altos, CA: Morgan Kaufmann, 25-37.

Sleeman, D. (1983), Inferring student models for intelligent computer-aided instruction, in Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. (Eds.), *Machine Learning: an artificial intelligence approach*, Palo Alto, CA: Morgan Kaufmann, 483-510.

Sperber, D. and Wilson, D. (1986), *Relevance: Communication and Cognition*, Oxford, UK: Basil Blackwell.

Stanlake, G.F. (1976), *Introductory Economics*, Longman.

Stead, R. (1990), Problems with learning from computer-based simulations: a case study in economics, *British Journal of Educational Technology*, 21(2), 106-117.

Stevens, A., Collins, A. and Goldin, S.E. (1982), Misconceptions in students' understanding, in Sleeman, D. and Brown, J.S. (Eds.), *Intelligent Tutoring Systems*, London: Academic Press, 13-24.

Stevens, A.L. and Gentner, D. (1983), Introduction, in Stevens, A.L. and Gentner, D. (Eds.), *Mental Models*, Hillsdale, NJ: Lawrence Erlbaum, 1-6.

Svendsen, G.B. (1991), The influence of interface style on problem solving, *International Journal of Man-Machine Studies*, 35(3), 379-397.

Swartout, W.R. (1983), XPLAIN: a system for creating and explaining expert consulting programs, *Artificial Intelligence*, 21(3), 285-325.

Topping, K. (1988), *The Peer Tutoring Handbook: Promoting Co-operative Learning*, Beckenham, Kent: Croom Helm.

Twidale, M. (1989), *The Use of Explicit Intermediate Representations in Intelligent Tutoring Systems*, Unpublished PhD thesis, Computing Department, Lancaster University.

Twidale, M. (1991), Student activity in an intelligent learning environment, *Intelligent Tutoring Media*, 2(3/4), 113-127.

Twidale, M. (1993), Redressing the balance: the advantages of informal evaluation techniques for intelligent learning environments, *Journal of Artificial Intelligence in Education*, 4(2/3), 155-178.

VanLehn, K. (1988), Student modeling, in Polson, M.C. and Richardson, J.J. (Eds.), *Foundations of Intelligent Tutoring Systems*, Hillsdale, NJ: Lawrence Erlbaum, 55-78.

Weiner, J.L. (1980), BLAH, a system which explains its reasoning, *Artificial Intelligence*, 15(1-2), 19-48.

Wenger, E. (1987), *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*, Los Altos: Morgan Kaufmann.

White, B.Y. and Fredericksen, J.R. (1987), Qualitative models and intelligent learning environments, in Lawler, R.W. and Yazdani, M. (Eds.), *Artificial Intelligence and Education, Volume 1: Learning Environments and Tutoring Systems*, Norwood, NJ: Ablex, 281-305.

White, B.Y. and Fredericksen, J.R. (1990), Causal model progressions as a foundation for intelligent learning environments, *Artificial Intelligence*, 42(1), 99-157.

Wilks, Y. and Ballim, A. (1987), Multiple agents and the heuristic ascription of belief, *Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, 118-124.

Wilks, Y. and Bien, J. (1983), Beliefs, points of view and multiple environments, *Cognitive Science*, 7(2), 96-119.

Williams, M.D., Hollan, J.D. and Stevens, A.L. (1983), Human reasoning about a simple physical system, in Gentner, D. and Stevens, A.L. (Eds.), *Mental Models*, Hillsdale, NJ: Lawrence Erlbaum, 131-153.

Wolf, B. (1988), Representing complex knowledge in an intelligent tutor, in Self, J. (Ed.), *Artificial Intelligence and Human Learning: Intelligent Computer-Aided Instruction*, London: Chapman and Hall, 3-27.

Wolf, B. and McDonald, D.D. (1984a), Building a computer tutor: design issues, *IEEE Computer*, 17(9), 61-73.

Yerushalmy, M. (1991), Student perceptions of aspects of algebraic function using multiple representation software, *Journal of Computer Assisted Learning*, 7(1), 42-57.

Yogev, A. and Ronen, R. (1982), Cross-age tutoring – effects on tutors' attributes, *Journal of Educational Research*, 75(5), 261-8.

Young, R.M., Green, T.R.G. and Simon, T. (1989), Programmable user models for predictive evaluation of interface designs, *Conference on Human Factors in Computing, CHI'89*, Austin, Texas, ACM, 15-19.

Appendix A

Dialogue Strategy Functions in DENISE

A.1 Introduction

The dialogue strategy functions provide information to determine the next action of the system. There are four types of function:

- *model access functions*: these interrogate the state of the learnt model
- *model maintenance functions*: these modify the contents of the learnt model
- *interface functions*: these present a question, dialog, menu etc. to the user
- *programming functions*: these allow plan variables to be compared, tested, created, modified etc and allow access to miscellaneous system resources such as random number generators.

The generic template of a function specification is given below:

```
function_name( Argument1, ... ArgumentN )
  { Return Values } : [ New Variables ]
– natural language description
```

The arguments may be constants or plan variables (which DENISE instantiates before passing to the function code). The return values are a complete list of all values the function can return to the dialogue strategy. This value can be ignored by the strategy if desired. The list inside the [] brackets is any new plan variables the function creates; an empty list signifies that no new variables are to be created.

A.2 Model Access Functions

Model access functions are characterised by being read-only operations; they do not alter the contents of the learnt model.

```
present( Viewpoint, Concept1, Relation, Concept2 )
  { yes, no } : []
– is the given relationship in the given viewpoint
```

find_relation(Viewpoint, Concept1, Concept2)
 { *yes, no* }: [*Relation*]
 – is there a relation between *Concept1* and *Concept2* ?

find_dependent(Viewpoint, Concept1, Relation)
 { *yes, no* }: [*Concept2*]
 – is there a dependent variable matching *Concept1* and *Relation*?

sub_viewpoint(Viewpoint)
 { *yes, no* }: [*Viewpoint_List*]
 – find the immediate child viewpoints of *Viewpoint*

find_concept_no_independent_var(Viewpoint,)
 { *yes, no* }: [*Concept2_List*]
 – find concepts with no causal links to them

find_concept_no_dependent_var(Viewpoint)
 { *yes, no* }: [*Concept1_List*]
 - find concepts with no causal links from them

peer_viewpoints(Viewpoint)
 {*yes, no* } : [*Viewpoint_List*]
 - find the peer viewpoints to *Viewpoint*

contradiction(Viewpoint, Concept1, Relation , Concept2)
 {*yes, no* } : [*Message*]
 - find contradictions and generate an appropriate message

A.3 Model Maintenance Functions

Model maintenance functions alter the contents of the learnt model; adding and deleting relationships and viewpoints.

add_to_model(Viewpoint, Concept1, Relation, Concept2)
 { *yes, no* }
 – add the relationship (*Concept1, Relation, Concept2*) to *Viewpoint*

del_viewpoint(Viewpoint)
 { *yes, no* }
 – delete the viewpoint *Viewpoint* and its contents

add_viewpoint(New_Viewpoint, Parent_Viewpoint)
 { *yes, no* }
 - add a new viewpoint beneath *Parent_Viewpoint*

```

remove( Viewpoint, Concept1, Relation, Concept2 )
{ yes, no }
- remove the relationship (Concept1, Relation, Concept2) from
Viewpoint

```

A.4 Interface Functions

Interface functions causes changes in the user's perception of the system. In DENISE they display different text messages, menus, buttons etc.

```

causal_dialog_mixed( Message )
{ yes, no, dk, mixed }: [ Concept1, Relation, Concept2 ]
- general dialog window
causal_dialog_control
{ yes, mixed }: [ Concept1, Relation, Concept2 ]
- let the user take control
causal_dialog_mixed1(Message, Concept2 )
{ yes, no, dk, mixed }: [ Concept1, Relation, Concept2 ]
- ask for an independent concept
causal_dialog_mixed2( Message, Concept1)
{ yes, no, dk, mixed }: [ Concept1, Relation, Concept2 ]
- ask for a dependent concept
yes_no_dialog(Message )
{ yes, no }: []
- simple choice
text_dialog(Message)
{ yes }: []
-display a message
ask_new_viewpoint
{yes, no}: [ Viewpoint, Parent_Viewpoint ]
- ask for a new viewpoint

```

A.5 Programming Functions

Programming functions allow the dialogue strategy to take decisions not immediately based on the result of a model access function or an interface function. These include typical programming activities such as testing, comparing and assigning variables. Access is also provided to facilities such as counting list contents and random numbers.

`assign(Value)`
{ *yes* }: [*New_Variable*]
– *New_Variable* becomes *Value*

`assign(Variable)`
{ *yes* }: [*New_Variable*]
– *New_Variable* takes the value of *Variable*

`head(List)`
{ *yes* }: [*Head*]
– takes the head of *List* (Lisp: `cdr`)

`count(List)`
{ *yes* }: [*Number*]
– list length

`random(Probability)`
{ *yes, no* }: []
– return *yes* if a random number between 1 and 100 is less than *Probability*, otherwise *no*

Appendix B

The Experimental Dialogue Strategy

B.1 Strategy

This appendix lists the code for the experimental dialogue strategy used in Chapter 6; the plan 'economics_exp1'.

The strategy is made up of a network of nodes connected by arcs as shown in Figure 5.12. The network is represented as clauses in LPA MacProlog 4.5 with the form:

```
node( Node_Name, Action_List, Arc_List ).
```

Plan variables are prefixed with a \$, e.g. \$current_viewpoint.

B.2 'economics_exp1'

```
/* economics_exp1 */
```

```
node( start, [
```

```
  [ text_dialog, 'Welcome to DENISE and the economics_exp1 plan' ],
```

```
  [ text_dialog, 'I'll have to place your beliefs into a viewpoint so the first
  thing to do is to give it a name' ] ],
```

```
  [
```

```
    [ viewpoint_create, always ] ] ).
```

```
node( node1, [
```

```
  [ causal_dialog_mixed, 'Ok, tell me something about economics' ] ],
```

```
  [
```

```
    [ node_false_start, no ],
```

```
    [ node_false_start, dk ],
```

```
[ control, mixed ],
```

```
[ node2, yes ] ] ).
```

```
node( node2, [
```

```
[ present, [ $current_viewpoint, $concept1, $relation, $concept2 ] ] ],
```

```
[
```

```
[ node_intermediate, no ],
```

```
[ node_already_present, yes ] ] ).
```

```
node( node3, [
```

```
[ causal_dialog_mixed, 'I already know that. Tell me something new' ] ],
```

```
[
```

```
[ control, mixed ],
```

```
[ node1, dk ],
```

```
[ node1, no ],
```

```
[ node2, yes ] ] ).
```

```
node( control, [
```

```
[ causal_dialog_control, null ] ],
```

```
[
```

```
[ node1, control ],
```

```
[ control, yes ] ] ).
```

```
node( node_independent_test, [
```

```
[ find_concept_no_independent_var, '$current_viewpoint ' ] ],
```

```
[
```

```
[ node_ask_independent, yes ],
[ node_dependent_test, no ] ] ).
```

```
node( node_ask_independent, [
  [ head, '$independent_list' ],
  [ causal_dialog_mixed1, [ "Ok so what affects $head then", $head ] ] ],
[
  [ node_check_present, yes ],
  [ control, mixed ],
  [ node1, dk ],
  [ node1, no ] ] ).
```

```
node( viewpoint_create, [
  [ viewpoint_create_dialog, null ] ],
[
  [ viewpoint_create, no ],
  [ node1, yes ] ] ).
```

```
node( node_false_start, [
  [ causal_dialog, 'Well that"s not a very helpful start is it? You"re
supposed to be teaching me things. Come on, tell me something about
economics.' ] ],
[
  [ control, mixed ],
  [ node2, yes ],
  [ node_false_start, dk ],
```

[node_false_start, no]]).

node(node_already_present, [

[causal_dialog, 'You've already told me \$concept1 \$relation \$concept2
Tell me something new']],

[

[control, mixed],

[node2, yes],

[node1, dk],

[node1, no]]).

node(node_contradiction_test, [

[contradiction, '[\$current_viewpoint, \$concept1, \$relation, \$concept2]']],

[

[node_intermediate1, yes],

[node_intermediate, no]]).

node(node_intermediate1, [

[yes_no_dialog, '\$message']],

[

[node_intermediate, yes],

[node_default_response, no]]).

node(node_intermediate, [

[add_to_model, '\$current_viewpoint \$concept1 \$relation \$concept2'],

[random, '85']],

```

[
  [ node_other_viewpoint, two ],
  [ node_independent_test, one ] ] ).

node( node_dependent_test, [
  [ find_concept_no_dependent_var, '$current_viewpoint' ] ],
  [
    [ node_default_response, no ],
    [ node_ask_dependent, yes ] ] ).

node( node_ask_dependent, [
  [ head, '$dependent_list' ],
  [ causal_dialog2, [ "What does $head affect then", $head ] ] ],
  [
    [ node_default_response, dk ],
    [ node_default_response, no ],
    [ node2, yes ],
    [ control, mixed ] ] ).

node( node7, [
  [ add_to_model, '$current_viewpoint $concept1 $relation $concept2' ],
  [ random, '40' ] ],
  [
    [ node_intermediate, one ],

```

[node_dependent_test, two]]).

node(node8, [
 [present, '\$current_viewpoint \$concept1 \$relation \$concept2']],
 [
 [node2, yes],
 [node_default_response, no]]).

node(node_default_response, [
 [causal_dialog, 'Ok, go on']],
 [
 [node2, yes],
 [node1, dk],
 [node1, no],
 [control, mixed]]).

node(node_other_viewpoint, [
 [yes_no_dialog, 'Can you think of another viewpoint besides
 \$current_viewpoint']],
 [
 [node_other_viewpoint2, yes],
 [node_default_response, no]]).

node(node_new_viewpoint, [
 [yes_no_dialog, 'Change to the viewpoint \$new_viewpoint']],

```
[  
  [ node_default_response, no ],  
  [ node10, yes ] ] ).
```

```
node( node10, [  
  [ assign, '$current_viewpoint $new_viewpoint' ] ],  
  [  
    [ node_default_response, always ] ] ).
```

```
node( node_check_present, [  
  [ present, '$current_viewpoint $concept1 $relation $concept2' ] ],  
  [  
    [ node_already_present, yes ],  
    [ node7, no ] ] ).
```

```
node( node_other_viewpoint2, [  
  [ ask_new_viewpoint, null ] ],  
  [  
    [ node_new_viewpoint, yes ],  
    [ node_default_response, no ] ] ).
```

Appendix C

Experimental Examples

C.1 Examples

These are the examples that were given to the experimental subjects in the domain of politics:

state_of_economy, pos, gov_opinion_poll_rating

sporting_failure, neg, gov_opinion_poll_rating

ice_cream_sales, pos, opp_opinion_poll_rating

gov_opinion_poll_rating, part_of_+, gov_lead

opp_opinion_poll_rating, part_of_-, gov_lead

There are also two unsigned relationships, `related`, which is a causal relationship without a sign and `part_of_?`, which is a constituent relationship without a sign.

The models are structured into viewpoints: these may be people's beliefs (yours, mine, Karl Marx), parties (Conservative, Labour), institutions (TUC), etc.

