AN EXPLANATION-DRIVEN UNDERSTANDING-DIRECTED MODEL FOR INTELLIGENT TUTORING SYSTEMS

Benita Mary Cox
Imperial College of Science, Technology & Medicine

A thesis submitted for
The Degree of Doctor of Philosophy
of the University of London

July,1989

Acknowledgements

I would like to thank my supervisor, Dr. John Jenkins, for his guidance through this work. I am also particularly grateful to Dr. Elizabeth Pollitzer for her constant assistance, encouragement, support and interest over the last three years.

The author was supported financially by the Science and Engineering Research Council.

My thanks to my parents for their continuing trust in my abilities and encouragement and to Ian Mitchell, for his useful comments and discussions.

I am also grateful to all those at Imperial College who have offered help and friendship over the course of the research.

Finally, I thank my husband and children, for their fortitude over the last three years.

Abstract

This thesis describes a set of design principles, namely, the Explanation-Driven, Understanding-Directed (EDUD) model, for the development of an Intelligent Tutoring System shell. The thesis addresses and proposes solutions to three crucial problems of Intelligent Tutoring System (ITS) design. The first problem concerns the design of a User Model, capable of identifying the user's current knowledge of the instructional material and the way in which this knowledge grows. The second concerns a method of constructing a Knowledge Base which may be applicable to a range of subjects and which is governed by a set of specially devised maxims. Finally, a Tutoring Module is developed which is amalgamated with the proposed User Model and Knowledge Base via the established design principles.

The EDUD model incorporates a number of principles derived from psychology and pedagogy as well as techniques from artificial intelligence (AI). Cognitive Structuralist theory provides principles for a layered hierarchical structure of generalised human understanding on which the User Model is based. The Tutoring Module contains a set of explanation-seeking questions, which are presented to the user in response to information obtained from the User Model. The purpose of the questions is to establish either the current level of the user's understanding or to provide the user with explanations that will drive him to a 'higher' level of understanding.

The practical utility of these principles is demonstrated through the operation of the EDUD model. Finally a system called MEDUD has been constructed to serve as an example of an ITS based on the Explanation-Driven, Understanding-Directed design principles.

TABLE OF CONTENTS

Tables

Figures

Synopsis

This synopsis provides a brief overview of the work described in the thesis and as such it is not intended to carry any detailed exposition of the concepts involved.

GENERAL BACKGROUND AND MOTIVATION FOR THE THESIS

The potential advantages of using computers for instructional purposes has given rise to considerable research effort in this area in recent years. Traditional Computer Based Instruction (CBI) has made rapid advances because it has incorporated ideas from the cognitive sciences as well as techniques from the discipline of Artificial Intelligence (AI). Furthermore, it has taken advantage of the increased capabilities of computer hardware and software. It is the convergence of CBI, AI and the cognitive sciences which has given rise to the appearance of Intelligent Tutoring Systems (ITSs).

Despite the impressive advances in ITS technology over the past fifteen years, many problems remain unresolved. Some of these difficulties are specific to particular areas of tutoring, for example, the difficulty of representing every possible solution without combinatorial explosion common to arithmetic tutoring systems. Other problems concern teaching techniques in general and cut across individual subject areas and are therefore generic to ITS design. This research identifies and proposes solutions to some of the problems which are generic to ITS design. In particular, it establishes design principles for a User Model, Knowledge Base and Tutoring Module which are applicable across several teaching domains and as such provide a basis for an ITS shell design. Whilst several shells have been implemented for individual modules of ITSs, such as BIP (Wes77), which controls the task selection process of the Tutoring Module and PIXIE (Sle82) which is a data-driven diagnostic/remedial system for the User Model, the shell model described in this thesis shows how the abovementioned three modules of an ITS can be unified both structurally and operationally.

The objective of the proposed design principles is to contribute to three crucial problems common to existing ITSs. The first one concerns the need to have an adequate and accurate representation of the user's initial knowledge of a domain and the way in which this knowledge develops as his understanding of the taught area expands and/or deepens (User Model). To date, ITS design has usually been determined by the specific requirements of a selected subject, incorporating in the User Model only those psychological processes that are assumed to be associated with teaching that subject. The result of this approach has been that few generic principles for the design of the User Model have emerged which are portable across knowledge domains or flexible enough to form a User Model shell. The User Model described and demonstrated in this work is portable between a range of domains.

The second problem addressed in this thesis concerns the development of a generalised knowledge decomposition method applicable to a range of subjects and satisfying the requirements of a Knowledge Base of an ITS. By and large the design of current ITSs has been subject-specific and the adopted knowledge decomposition methods reflect the requirements of tutoring the given subject. As a result it is difficult to apply a particular knowledge decomposition method to another domain. This thesis is concerned with the problem of decomposing knowledge for the purpose of its incorporation into a general-purpose shell. A method of knowledge structuring is described which satisfies this condition for subjects which use non-numeric models and/or heuristic procedures for solving problems, for example, medicine, engineering and business (Cla87).

The last problem addressed in this thesis is that of providing a generalised teaching strategy. The objective has been to design a Tutoring Module which can be integrated both structurally and operationally with the User Model and Knowledge Base.

RESEARCH METHOD

The basic method of research has been to pursue three complimentary activities, namely, to conduct a literature survey to identify current issues in ITS, to develop a theory by which the stated objectives could be achieved and to demonstrate the applicability of the theory in its implementation in a computer program (Chapter 2).

A literature survey is provided which presents an overview of the historical perspective of current problems and concerns of ITS (Chapter 3). This survey has led to a conclusion that there was a need to incorporate ideas from a number of disciplines into the design framework. Psychology provided insights into how individuals learn and principles for the design of the User Model; pedagogics provided principles for designing tutoring strategies and Artificial Intelligence provided the techniques for implementing these principles in a computer program (Chapter 5).

The basic view taken in this thesis is that the critical issue in ITS design is the link between the way in which students learn, the way in which a system explains knowledge and the way in which the user achieves an understanding of the explanations presented to him. Thus the ITS model described herein may be appropriately called 'Explanation Driven' and 'Understanding Directed' (EDUD).

DETAILS OF THE EDUD MODEL

The key notions of the EDUD model are explanation and understanding. The importance of explanation has been long recognized in expert system design. There are differences, however, between the demands put on explanation facilities in expert systems and in tutoring systems. These distinctions are discussed in detail in the thesis. Having done so, the thesis suggests a method for constructing explanation facilities suitable for ITSs. The basic objective has been to provide explanation facilities capable of communicating information to the user appropriate to the current state of his knowledge.

The basis for this explanation facility has been obtained from Achinstein's work on scientific explanation (Ach83). Briefly, he has developed a typology of basic categories of explanation which included identity explanations, functional explanations, causal explanations and derivational/complex explanations. These categories comprehensively describe declarative and procedural knowledge. In addition a further category called hypothetico- deductive explanations has been incorporated into this typology (Chapter 7). The latter is concerned with a definition of how new potential can be realized from existing knowledge.

The view taken in the thesis is that the purpose of explanation is to increase the user's understanding of a given subject matter. A more detailed definition of what is meant by understanding is contained in the main body of the thesis. In particular a distinction is drawn between seeing understanding as a general process of moving from one knowledge state into a higher knowledge state and seeing understanding in terms of practical manifestions demonstrated in particular subject areas. It has hitherto been the practice of ITS development to concentrate on the latter aspect, that is to analyse the particular logic behind individual problem-solving techniques rather than to identify the general processes by which different students may come to establish an understanding of why solutions to the given problems 'make sense'. The view taken in this research has been that in many knowledge domains it is advantagous if the teaching system is designed to first present the relevant facts and procedures to the user making sure that he is able to recall these and then lead him to apply that knowledge creatively in hypothetico-deductive reasoning. That is, in such a tutoring system the process of understanding naturally culminates in successful problem-solving rather than using the method of achieving a particular state of understanding by repeatedly presenting the user with a particular kind of problem. Therefore, in the EDUD model it is the facts and procedures that are the precursors of successful problem-solving that are used as the focus for the model of the process of understanding. In this context a hierarchical framework for identifying levels of understanding has been devised (Chapter 8) based on a classical psychological theory of learning, namely, the cognitive structuralist theory.

Briefly, the theory proposes that learning can be seen as a progression from less to more advanced states of understanding. This progression is reflected in the evolutionary growth of cognitive structures. This term is used as meaning the process by which an internal representation of an external reality steadily corresponds more closely to that reality as understanding is increased. Classical cognitive structuralist theory views the order in which this development takes place as universally sequential (Chapter 9).

There are considerable advantages in adopting this approach to the design of the User Model. For example, once the levels of understanding are identified and their sequential arrangement specified, it is possible to pinpoint the conceptual relations already understood at a specific phase of the acquisition of knowledge. Consequently it is possible to specify those relations which still need tuition. In this way the hierarchy of levels of understanding provide a frame of reference for monitoring the understanding process as well as establishing a reference point from which to control explanations driving this process (Chapter 12). The levels of understanding identified in the thesis are:

The stage of figurative knowing
The stage of functional understanding
The stage of cause/effect understanding
The stage of complex-derivational understanding
The stage of hypothetico-deductive reasoning

The process of understanding is seen as the progressive ascent through the levels of this hierarchy. Essential to this progression is the propensity to ask appropriate explanation- seeking questions. Specific types of questions correspond to the levels of the hierarchy of understanding. For example, figurative knowing results from the question "what is it?" and functional understanding results from the question "what does it do?". The set of questions associated with the hierarchy of understanding reflects the 'pragmatic rules of inference'. A detailed definition of the pragmatic rules of inference is given in the thesis (Chapter 10). It is demonstrated in the thesis that the corpus of these questions can be taken together and systematically applied to a body of knowledge thereby analysing it for incorporation into an ITS

Knowledge Base. This 'knowledge decomposition' method can be applied across a number of domains. (Chapter 13). The development of general rules for knowledge decomposition represents the second critical research area addressed by the thesis and a contribution towards ITS techniques.

Finally, the thesis has addressed the problem of teaching strategy in ITSs. The relevant issues considered included the questions of when to intervene, what to discuss, which presentation strategy to use and how much to say in the teaching interaction (Chapter 11). The framework devised as a solution to this problem takes the student progressively from the lowest level to the highest level of understanding, ensuring that no knowledge was presented before the necessary anterior understanding has been achieved (Chapter 14).

Very briefly, then, the thesis proposes a unifying design framework for an ITS shell and in particular for the design of the User Model, Knowledge Base and Tutoring Module. In this framework the notions of understanding and explanation are seen as central. The model is based on a hierarchy of explanations and a hierarchy of levels of understanding whereby one maps directly onto the other. Furthermore, these hierarchies enable a systematic decomposition of a body of knowledge and its communication to the user via a linked set of explanation seeking questions which exemplify pragmatic rules of inference. The practicality of these ideas has been demonstrated in a prototype ITS (MEDUD) dealing with the subject of car engines.

Throughout the period of the research, the ITS research community has expressed considerable interest in the ideas and results discussed in the thesis as it has been presented in seminars, conferences and publications (See Appendix B).

STRUCTURE OF THE THESIS

The thesis has been divided into four parts.

Part 1 provides the background and motivation (Chapters 1 and 2) for the research described in Parts 2 and 3 and discusses past and current developments in ITS research (Chapter 3).

Part 2 introduces the Explanation-Driven Understanding-Directed model (Chapter 4) and presents arguments as to the desirability of approaching ITS design from both a multi-disciplinary (Chapter 6) and a domain-independent framework (Chapter 5). The theoretical issues involved in the development of EDUD are discussed. (Chapters 9,10,11). Particular attention is paid to the concepts of explanation (Chapter 7) and understanding (Chapter 8).

Part 3 is concerned with the application of the theoretical principles described in Part 2 in the design of an ITS. In particular, their incorporation in a User Model (Chapter 12), a framework for knowledge decomposition (Chapter 13) and a Tutoring Module (Chapter 14) are described.

Part 4 describes the implementation of these principles in a small program, MEDUD (Chapter 15).

The thesis has been structured so that it is perhaps best read sequentially. However, the reader may find it easier to approach the theoretical aspects of the work by reading the thesis in conjunction with the description of their implementation in the MEDUD program, which is discussed in Chapter 15. Moreover, the arguments for taking a multi-disciplinary approach and the inclusion of domain-independent principles in ITS design are included for the sake of completeness but may be overlooked.

Chapter 1

INTRODUCTION

1.1 MOTIVATION FOR THE RESEARCH

The motivation for this research stemmed from the author's prior interest in how people reason and learn about their environment. This interest was enhanced by the emergence of artificial intelligence techniques which afforded opportunities for the formalisation of cognitive theories in a form suitable for computer implementation. In addition, the increased availability of, and growing interest in, using computers for education offered the possibility of practical applications for the research idea.

1.2 AIMS OF THE RESEARCH

The primary objective of this research has been to develop generic design principles for Intelligent Tutoring Systems (ITSs) which would serve as the framework for the design of an ITS shell.

In pursuing this objective the decision was taken to invert the traditional approach to ITS design which involves analysing a preselected task-domain and building a suitable tutoring system to meet its purposes. Instead it was decided to try to identify a set of fundamental principles on which ITS design in general could be built. Only then would specific tasks and domains be selected and analysed.

The primary task was to isolate relevant principles in the disciplines of artificial intelligence, pedagogy and psychology which could be built upon to support the design of an Intelligent Tutoring System. Secondly, the most common components found in current Intelligent Tutoring Systems,namely, the User Model, the Knowledge Base and the Tutorial Module, were identified and considered in relation to these broader principles. From this matrix, general principles of ITS design were established.

Finally, the practical operability of these principles was demonstrated in a small tutoring system.

From the outset, the notions of understanding and explanation were seen as central to the design of knowledge communication systems and therefore were tightly defined. As used these concepts form the focal point of the research and the basis for describing the proposed design method, as the EDUD model (Explanation-Driven, Understanding-Directed).

A schematic representation of the research project is shown in Figure 1.1.

1.3 RESEARCH PLAN

In order to achieve the aims described above, the following objectives had to be achieved:

1. formulation of an explicit theory of the cognitive process of understanding which is suitable for computer implementation,

2. identification of principles of explanation relevant to the design of a tutoring system,

3. selection and application of those techniques of artificial intelligence which allow the incorporation of the above principles in a computer model,

4. design and implementation of a User Model based on the principles identified in tasks 1,2, and 3,

5. development and illustration of a knowledge decomposition method to reflect the principles identified in tasks 1,2 and 3,

6. selection of an explicit pedagogic strategy to maximise the effective use of the principles of understanding and explanation described,

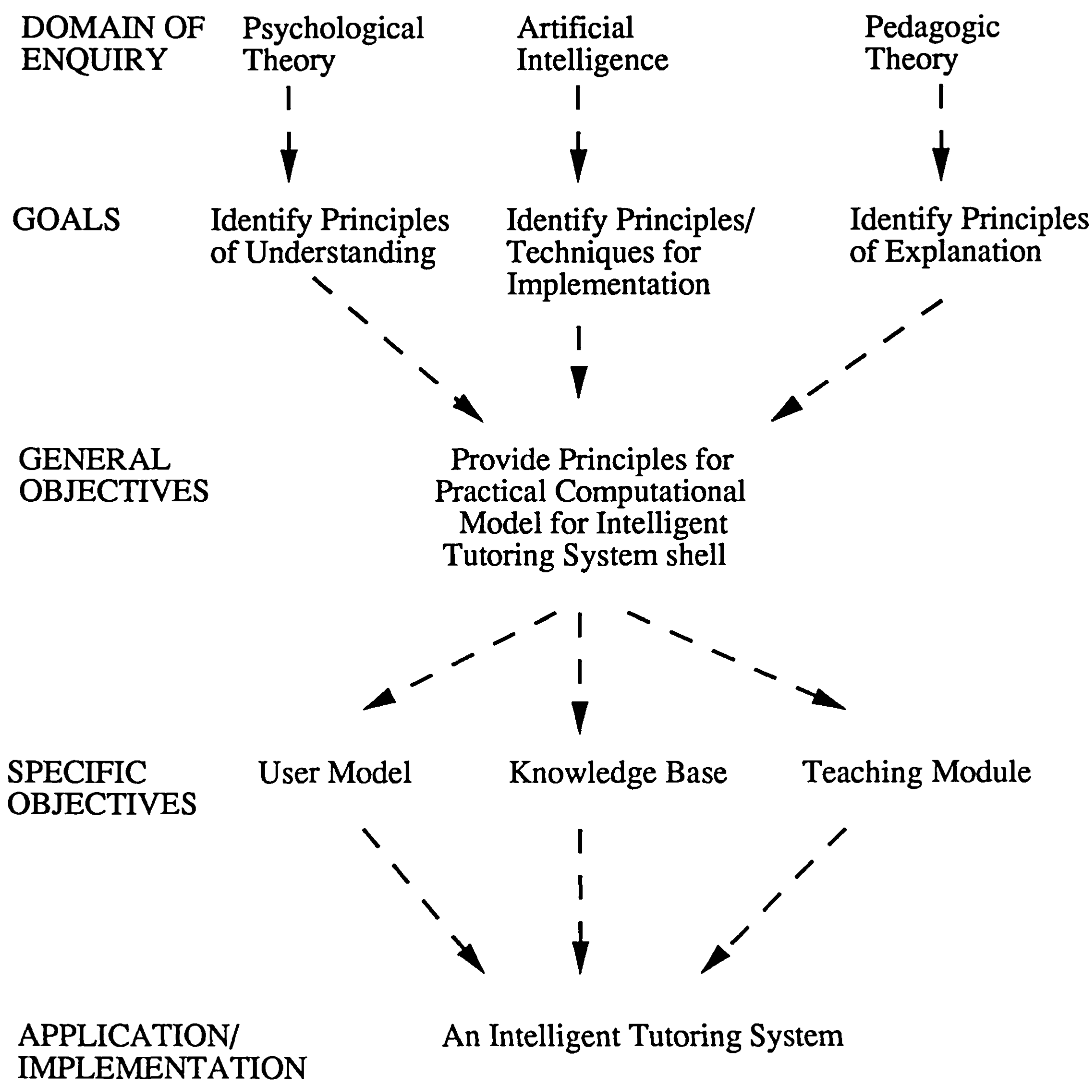

Figure 1.1
Schematic Representation of the Research Project

7. implementation of the design model in a computer program.

1.4 RESEARCH RESULTS

The tasks mentioned in the Research Plan were satisfied as follows:

1. The fundamentals of the cognitive theory on which the EDUD model is built have been derived from the cognitive structuralist approach to learning (Gar78) and the theory of pragmatic rules of inference (Nis87). Both of these theories emanate from the discipline of psychology.

2. The principles for providing appropriate explanations which would serve to increase the user's understanding of a domain in the EDUD model have been based on the theory of understanding as an epistemological process, derived from philosophy of education (Ham78). In addition, the EDUD model spells out categories of explanation, which define the content of explanations and which are derived from the principles of philosophy of science (Ach83).

3. The techniques of logic programming and knowledge representation formalisms, in particular those of semantic networks, frames and production rules have been used in the implementation of the EDUD model.

4. A User Model reflecting the above principles has been described and demonstrated in the MEDUD program.

5. A method for knowledge decomposition has been described and demonstrated by analysing knowledge from two separate domains, those of motor car mechanics and medicine.

6. In the EDUD model an epistemological approach as the basis for teaching has been described.

7. A small program, the MEDUD program, has been written to illustrate the incorporation of the principles of EDUD in a working system.

1.5 ASSESSMENT OF THE RESULTS

The main contribution of this research lies in the identification of a set of generic principles for computerised instruction/learning, and the formulation of these principles in the Explanation-Driven, Understanding-Directed (EDUD) model as a framework for the design of an ITS shell (see Figure 1.2).

The principles in question provide:

a) a framework for the design of a generic User Model, enabling a dynamic assessment of the user's level of understanding of a particular knowledge domain which facilitates the adaptation of the system's performance to the user's current knowledge level,

b) a formal method for the decomposition, organisation and retrieval of the domain knowledge to be incorporated in Intelligent Tutoring System shell design,

c) a framework for Teaching Modules which enables a systematic approach to be adopted to the selection of explanations for the user of a given knowledge-domain,

d) an implementation of the findings of this research in a computer-based teaching system.

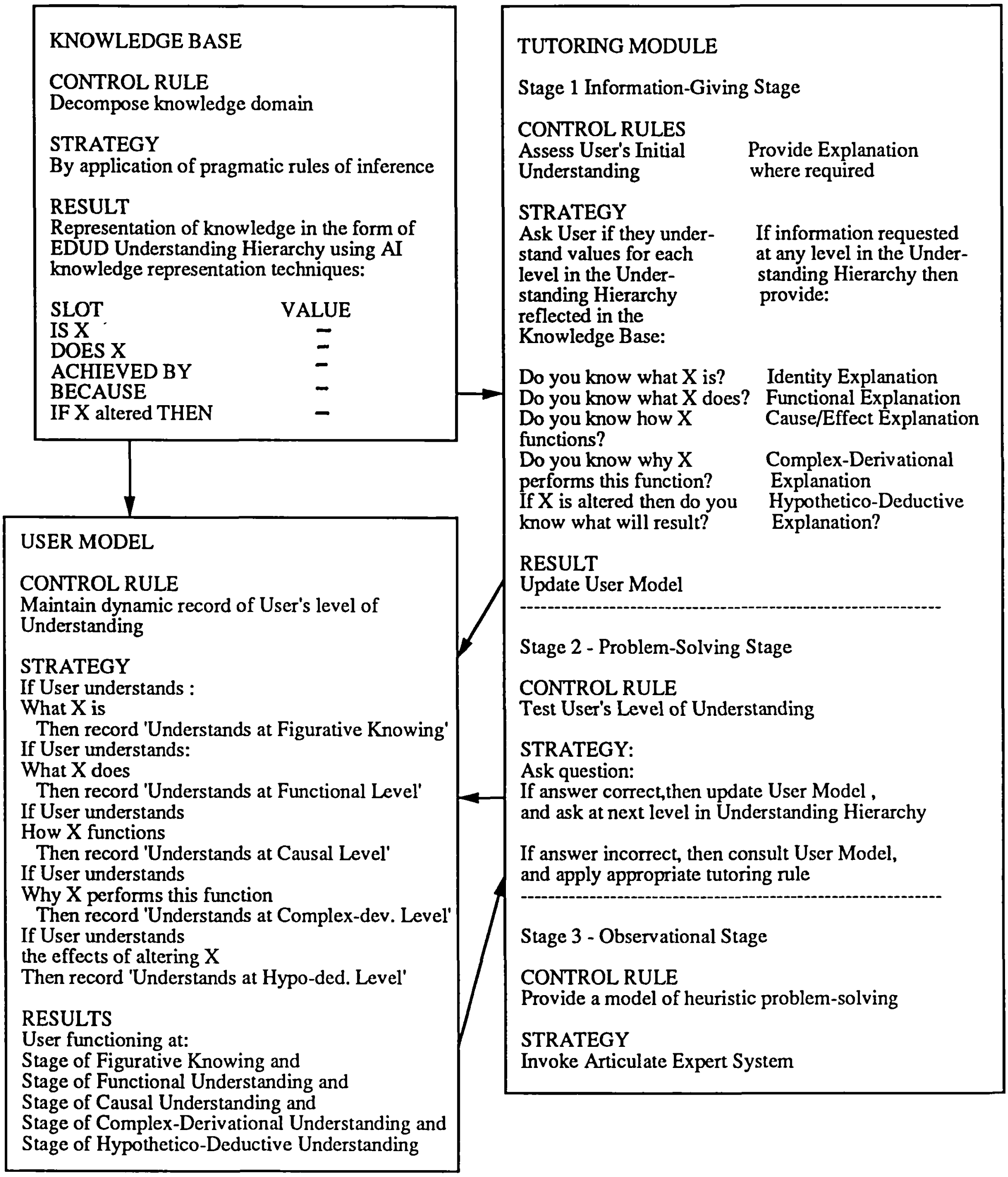

Figure 1.2
Schematic Representation of the EDUD Model

Chapter 2

RESEARCH APPROACH

There have been three complementary activities performed in the course of this research. These are:

2.1 LITERATURE SURVEY

This survey was necessary to identify the major current issues in Intelligent Tutoring System design, in particular, the importance placed on understanding and explanation as the focal design concepts. In addition, the search tried to identify research into the incorporation of generic or psychological principles in system design. The results of these analyses are discussed in detail in Chapter 3.

2.2 THEORY DEVELOPMENT

The research methods adopted are those applied by the artificial intelligence research community rather than the traditional research methods employed in disciplines such as cognitive psychology.

The subject of this thesis is 'knowledge communication' and an objective of the research is to clarify the relationship between explanation and understanding and the role played by these concepts in the knowledge communication process. The research has not , therefore, been based on empirical observation of computer- based learning but aims to elicit principles and construct supporting arguments for path(s) between surface observations and underlying structures (Gil88). This varies from task analysis where the predominant objective is to identify and teach problem-solving skills.

It will be argued that whilst findings from research in psychology may provide a measure of the observable changes in behaviour consequent upon the learning process, to date no measurable manifestations of cognition have proved adequate to describe complex changes in perception which result, for example, from increased understanding. Accordingly, the method employed in this research has not been to identify and quantify the results of psychological experimentation but rather to develop a theory which will underpin the design principles of the EDUD model for ITS design. These principles are demonstrated by applying them in a practical context in the knowledge domains of engineering and medicine.

The objective of this research is to base the EDUD model on mechanisms of knowledge communication rather than observations of events. As such this approach may be considered as leading to a 'deep' theory as opposed to a 'shallow' theory of ITS design. The validation of the proposed theory needs, therefore, to be based on analytical rather than experimental grounds.

This approach to establishing the correctness and usefulness of a theory has been influenced by the notion of competitive argumentation as described by Van Lehn(Van,83).

2.3 COMPUTATIONAL IMPLEMENTATION

A demonstration of the completeness of the theories is provided by their implementation in a computer model of an ITS, namely, the MEDUD program. This program shows that the stages identified on the basis of the theory of understanding adopted in the model are in fact capable of generating the explanations ascribed to them and that knowledge decomposed in accordance with EDUD principles is suitable for tuitional purposes.

Chapter 3

STATE OF THE ART IN ITS DESIGN

3.1 INTRODUCTION

Before discussing the EDUD approach in detail, a summary of background information on ITSs and a precis of the findings from the literature search are necessary.

3.2 BACKGROUND TO THE EMERGENCE OF INTELLIGENT TUTORING SYSTEMS

Computers have been used in education for almost 30 years (Man88). However, new opportunities provided for traditional approaches to computer-based instructional systems, have resulted in the emergence of the current generation of Intelligent Tutoring System designs. The main reasons for this development are as follows:

(i) the application of artificial intelligence (AI) techniques to computer aided instruction. Many of the techniques developed in the field of artificial intelligence are appropriate to instructional system design. Amongst the techniques which have made significant contributions to current ITS design are logic programming, production models of reasoning, knowledge representation techniques and knowledge engineering tools. In addition to AI techniques, there is a number of more specific expert system design features which have influenced the approach to computer-based instructional design. An important example of this is the principle of the provision and incorporation of the knowledge-base as a separate and explicit entity within a system. In other words, the separation of domain-specific knowledge from the reasoning mechanisms which manipulate such knowledge. The incorporation of this principle into ITS has allowed for the instructional strategy to be separated from the particular domain knowledge to be taught by the system,

(ii) the emergence of new communication media for the delivery of teaching material. Advances in hardware technology, such as increased memory capacity and computational speed have contributed to the performance of instructional systems. In addition, facilities such as low cost personal computers and networks have made distributed teaching possible. Advances in software technology such as fast and flexible graphics and human-computer interface packages, have also been of particular importance to the design of instructional systems.

(iii) Finally, the emergence of the current generation of ITS design may be seen as originating from attempts to deal with the shortcomings of Computer Aided Instruction (CAI) systems, in particular their inflexible presentation of teaching material as well as their limited capabilities for adaptive diagnosis and feedback.

A schematic representation of the major contributions to existing ITS design is shown in Figure 3.1.

3.3 DIFFERENCE BETWEEN COMPUTER AIDED INSTRUCTION (CAI) AND INTELLIGENT TUTORING SYSTEMS (ITSs)

The question of the way in which Intelligent Tutoring Systems differ from traditional Computer-aided Instructional systems may be approached at several levels. At the broadest level it may be said that ITSs are simply an attempt to improve CAI systems by the incorporation of AI techniques, but, in fact, the approach to ITS design is more than this. The approach to ITS design is fundamentally different from that followed in the design of CAI systems. This difference stems, in part, from the fact that ITS design has evolved as a sub-discipline of artificial intelligence and therefore has its roots in the discipline of computer science. The design of computer-aided instruction systems, on the other hand, has been strongly influenced by disciplines such as psychology and pedagogics (Kea87). Accordingly, the focus in ITS has been on technical aspects of system design.

However, the most important difference between CAI and ITS design is

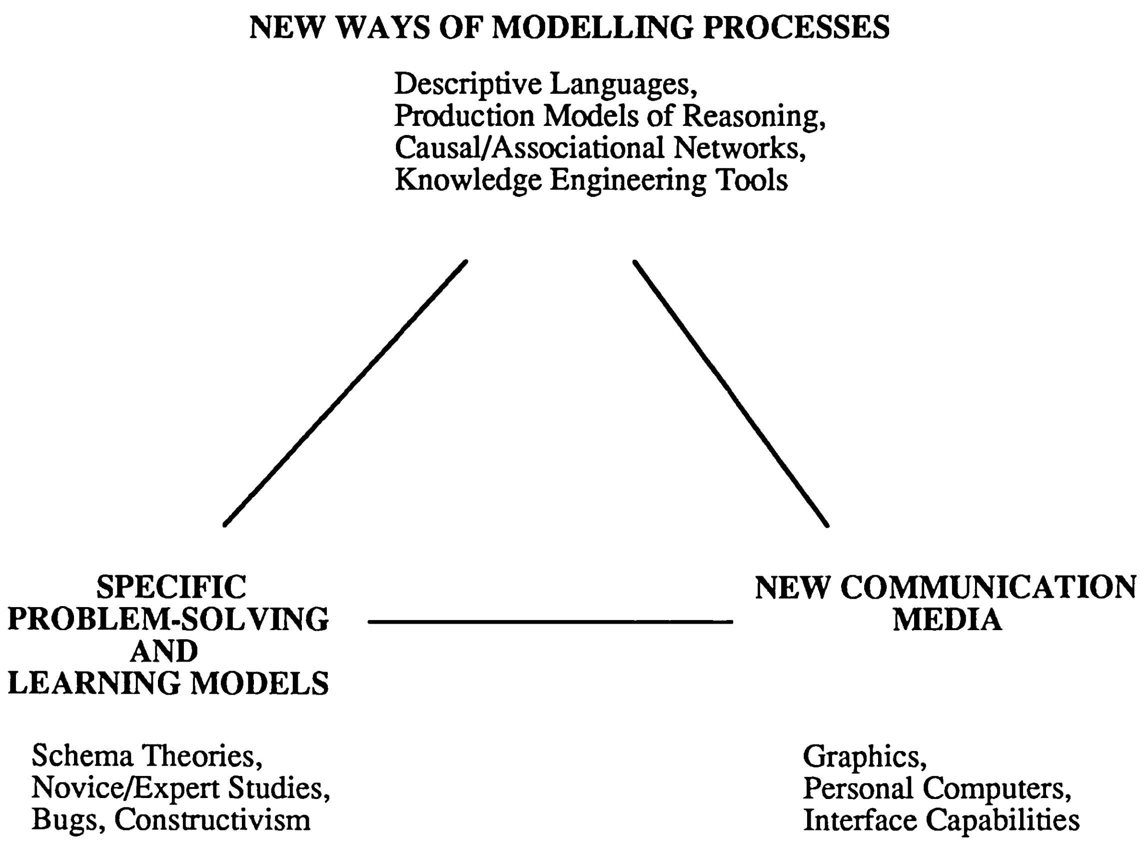

Figure 3.1
Interacting Contributions to New Instructional Technology
(Adapted from Cla87).

in their approach to the form of interaction with the user. ITS has the objective of creating 'flexible' and 'adaptive' systems. By contrast, CAI systems have pre-structured and inflexible forms of interaction. Conventional CAI systems stored all the information necessary for teaching a particular task in a prestructured form. Thus, the teaching content, anticipated user responses and decisions as to which route the user should be encouraged to take through the material were all pre-stated - usually in the form of an authoring language which allowed the teacher to write his own material. The typical instructional format in these systems was to 'expose' the user to some pre-selected information and then to test his understanding of it by evaluating his answers with reference to set solutions. Correct answers prompted presentation of the subsequent set of prestored alternatives. The choice of which alternatives to select for presentation to the user was usually determined by a branching network which generated material which it was believed was suitable for the user's level of competence (Sup67). However, the fact that each branch of the network had to be specifically pre-stated, resulted in a lack of flexibility in these systems.

Intelligent Tutoring Systems on the other hand are intended to overcome the problem of inflexibility by being 'adaptive' to the user's needs. Briefly, an adaptive tutoring system is one in which the student is routed through the material by a program which may follow different teaching strategies and whose task- domain is designed to fit a particular student's competence. The intention is that these systems be capable of a flexible response to different students as well as the one student as he gains competence and understanding over time.

3.4 A REVIEW OF THE MAJOR PARADIGMS IN EXISTING INTELLIGENT TUTORING SYSTEMS

Adaptive tutoring systems are currently based on five major paradigms. These are:

3.4.1. Coaches.

Instructional systems which are based on a coaching paradigm typically function by observing the user's performance, diagnosing where he/she is having difficulty, isolating those areas which are causing the difficulties and providing advice that will help the user to improve performance. Examples of coaching paradigms included in current ITS designs are the provision of supportive environments in which the user may explore. Such an environment is provided in the WUSOR system (Gol77) which is designed to foster the user's ability to make proper logical and probabilistic inference from the information given by providing him with a game environment.

3.4.2. Mixed Initiative Tutors.

The basic objective of a mixed initiative tutor is to share the learning initiative between the system and the user. The ideal is for the system to be able both to instigate questions from the answers to which it can gauge what the student is trying to achieve, and for it to be capable of responding to questions initiated by the user. In this way control of the interaction is shared through an exchange of questions and answers in a conversational mode with either the user or the system directing the dialogue.

The basic teaching strategy used in these tutors is that of a Socratic dialogue. The approach underlying this teaching method is that learning is encouraged by asking the user pertinent questions which direct him through a process of 'debugging' his own misconceptions. Examples of mixed-initiative systems are SCHOLAR which teaches Geography (Col85) and SOPHIE which is concerned with electronic trouble-shooting (Bro82).

3.4.3 The Microworld Concept.

Much of the research over the past six years has been centered on this approach. Like coaches, microworlds provide a supportive environment which allow a user to explore a particular problem domain. However, these systems differ from coaches in that the user receives no direct advice or guidance from the microworld software. The user is in full control of the activity and the only way the system can direct the course of action is by modifying the environment.

The intention in these systems is to facilitate learning-by-doing - to assist with "transforming factual knowledge into experiential knowledge" (Sle82). An example of this type of system is the LOGO programing environment which enables a user to learn geometry through exploration and experimentation (Pap80).

3.4.4. Diagnostic Tutors

These systems diagnose any misconceptions or 'bugs' that the user might have in his knowledge. Amongst the most well-known of these systems is BUGGY (Bro78) which uses a procedural network to identify bugs in the user's ability to perform subtraction tasks. Considerable research has also been carried out in the area of debugging of user's programming skills, for example, PROUST which identifies bugs in users learning Pascal (Joh84).

3.4.5. Expert Systems

Expert Systems may be adapted for teaching purposes by communicating the knowledge encapsulated in their knowledge-bases to a large number of students. Research into the feasibility of adapting existing Expert Systems for the purposes of tutoring has been pioneered by Clancey. In his program GUIDON (Cla81) Clancey augmented the domain expertise of MYCIN (Sho76) (an expert system designed to diagnose blood disorders) with additional levels of domain knowledge in order to help explain and organise the domain rules for teaching purposes.

3.5 SELF IMPROVING TUTORS

Whilst not a major paradigm in ITS design, mention must also be made of the work on self improving tutors. These systems are designed to acquire superior problem solving approaches, to those they have been programmed to achieve,from the users themselves (Kim82). In the Quadratics Tutor, for example, if the user's method of solving an equation requires fewer steps than that incorporated in the system's solution (O'Sh79) then the system adopts this method in preference to that which it has been programmed to perform.

3.6 MAJOR COMPONENTS OF ITSs

Whilst no 'common' architecture has yet emerged for Intelligent Tutoring System design, these systems are commonly constructed with four major components: a Knowledge Base, a User Model, a Tutorial Module and a User Interface. A general model of the architecture of an Intelligent Tutoring System is shown in Figure 3.2 (Kea87).

3.6.1. The Knowledge Base

This module comprises the facts and rules of a particular domain. The importance of the knowledge domain is a major aspect of all knowledge-based system design, and in particular of ITSs. This is because in ITSs the information contained in the knowledge base functions not only as in conventional expert systems as a source of knowledge which generates explanations and responses to the user but also as a standard for evaluating the user's performance. It must, therefore, be tailored to incorporate detailed ideas and processes rather than simply comprise knowledge about a domain at particular expert level, a broad theoretical orientation and general inference tools. In an ITS the nature of the stored knowledge determines not only the content of a tutorial interaction but also its goal structure (Ste82).

3.6.2. The User Model

This module provides an internal representation of the user's level of functioning and is based on the assumption that an effective communication tutor requires some knowledge of the student.

Early CAI systems represented the user's level of functioning in simple

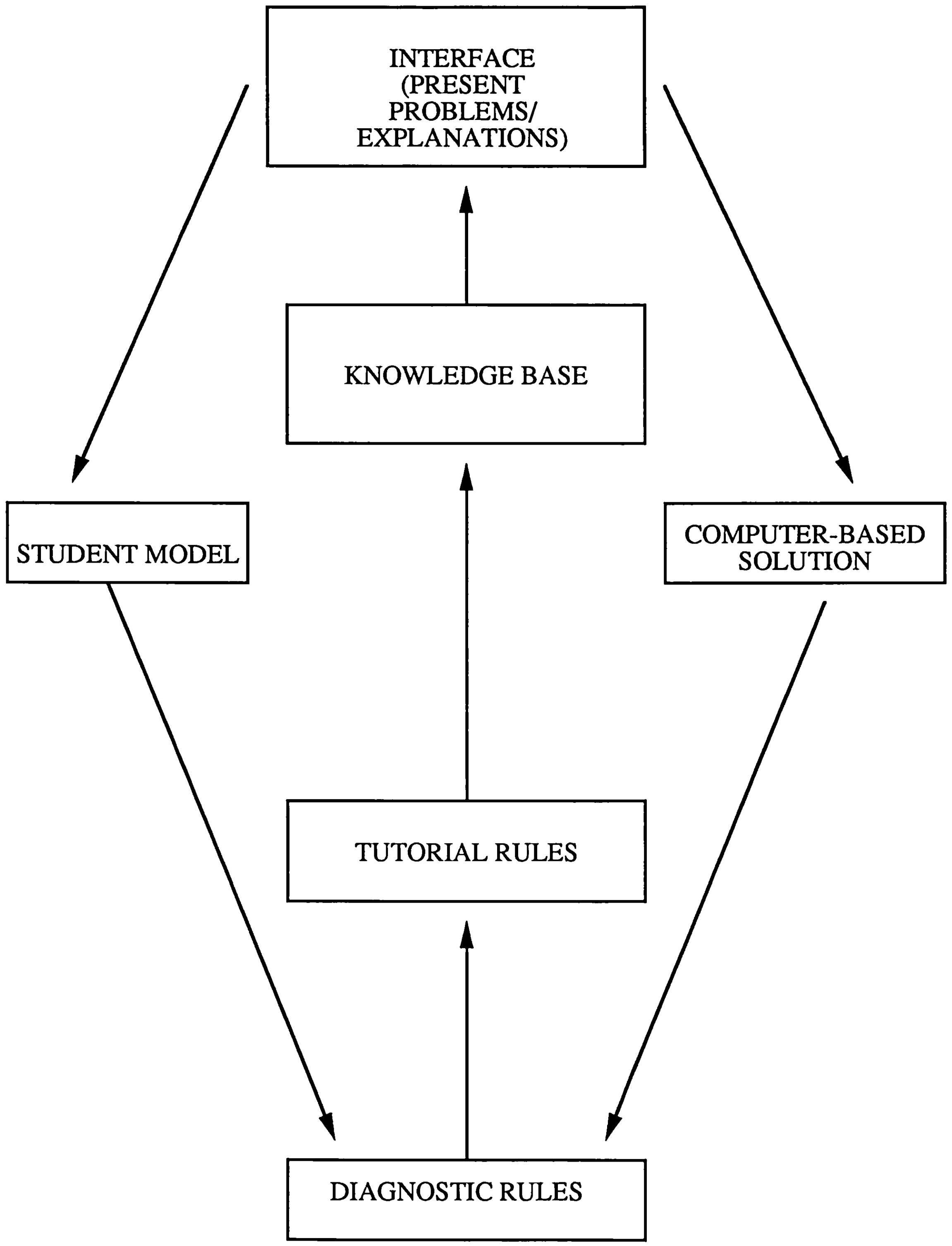

Figure 3.2
A General Model of an Intelligent Tutoring System
Adapted from (Kea 87)

parametric form. In current ITSs these parametric representations have been replaced by more complex methods which provide for prediction of the learning behaviour of individual users and diagnoses of the cause of errors. There is a number of ways in which this system representation of the user has been achieved. Amongst the more common methods of modelling are those which represent his knowledge state as a subset of that of an expert. The User Model is thus constructed by comparing the student's performance to the computer-based model of the expert's. Another common approach is to represent catalogues of 'bugs' which identify the misconceptions that students may have in solving a problem. User's replies are then compared with this representation (DEBUGGY) (Bro82). Alternatively domain knowledge may be represented as rules, and potential errors represented as variants of those rules (mal-rules) (LMS)(Sle82).

3.6.3. The Tutorial Module

This module determines and regulates the instructional interactions between the system and user. It provides decisions as to which teaching instructions should be presented to the user as well as how and when to present them. These pedagogic decisions are made with reference to the User Model and the knowledge domain. Common examples of teaching strategies incorporated in ITS design are those of the Socratic tutor and coaching methods.

3.6.4. The Interface

This module processes the flow of information in and out of the system. It provides the interface between the system and the user and as the determiner of the final form in which the user will receive the system's information, plays a vital role in the knowledge communication process. The interface can make or break the effectiveness of a tutor, regardless of the quality of the underlying design. There are not many communications formats currently available and these tend to reduce educational effectiveness by undermining user motivation and restricting the inferences which the User Model can make. However, communications issues in Intelligent Tutoring Systems are similar to those followed in the field of natural language comprehension and generation, which is a major area of research within artificial intelligence. A synthesis of this area is beyond the scope of this thesis.

3.7 A STATEMENT OF THE AIMS OF INTELLIGENT TUTORING SYSTEMS.

The aim of an Intelligent Tutoring System is to provide individualised, adaptive instruction dynamically (Yaz87).

A prerequisite of 'individualised' instruction is the ability to react to the cognitive level of an individual user. In order to achieve this, a system must be 'adaptive', that is it must be capable of adopting diverse teaching strategies for the selective presentation of instructional material. The instructional material in turn must be tailored to the user's abilities. In addition, the instruction must be dynamic, by which is meant that pedagogic decisions are generated on-line with reference to explicitly represented knowledge rather than predetermined for all users, irrespective of how their skills develop.

3.8 A REVIEW OF THE ACHIEVEMENTS OF EXISTING ITS

The first important step was taken in the 1970's with the change in emphasis in the design of tutoring system from the construction of a curriculum of exercises (as was used in CAI),to the representation of knowledge for the purposes of problem-solving (Cla87). This shift in emphasis resulted in complex programs being devised with modelling and problem- generation capabilities as well as the incorporation of explanation facilities (WEST, WUSOR, GUIDON) (Bur82,Gol77,Cla79).

Subsequently, attention was focussed on the modelling of errors (WHY, BUGGY, MENO) (Ste82,Bro78,Woo84) and their cause as well as on the construction of models of knowledge and reasoning (SOPHIE- III, PROUST) (Bro82,Joh84).

Since the early 1980's the incorporation of findings from the fields of knowledge representation, psychology and pedagogics have led to increased progress in the subject, for example the ACTP TUTORS (And85) (Cla87).

3.9 A REVIEW OF THE PROBLEMS IN EXISTING ITS

Despite the advances made in recent years in knowledge-based tutors, the subject is still in its infancy (Cla87). This is not surprising when the demanding objectives of providing individualised, dynamic instruction aspired to by these systems are considered. For example, in order to provide individualised instruction the system must respond to the particular cognitive level of the individual user. As this changes throughout the course of tuition, it has to be represented dynamically within the system. However, the problems of understanding how people learn are complex and current theories of learning are inadequate, with the result that the research basis on which to build intelligent tutors is in itself lacking (Kea87).

Fulfilling the objective of providing dynamic interaction between system and user also presents difficulties. Dynamic interaction requires that the response time of the system be virtually immediate, placing demands on both system hardware and software design. Further, if dynamic interaction is to be achieved, not only is the selection of the teaching material important but also the form in which it is presented. This places demands on the interface design which in some instances may take longer to produce than the underlying representation (Cla87).

These and other difficulties have lead to the limited success of current systems in achieving their aims. Amongst the major unresolved problems are the following (Sle82):

1. The teaching material provided by the system is frequently inappropriate for the user's needs.

This problem results from the system making incorrect assumptions about the user's current state of cognitive functioning. This incorrect assumption originates from an inadequate initial representation of the user within the system. This problem of inadequate representation of the user's level of cognitive functioning is also manifest when the task to be performed requires the application of multiple skills. The question the system has to answer in these circumstances is which particular skill is responsible for the user's correct/incorrect response? The system may incorrectly assign blame/credit to a skill and produce erroneous instructional material accordingly.

2. The system is unable to view the knowledge-domain from the user's perspective and accordingly it attempts to coerce the student into its own conceptual framework.

This problem also results from inadequate representation of the user within the system. It may arise, for example, from the difficulties caused by 'combinatorial explosion'. This occurs when there is a plethora of possible cognitive functions and reasoning paths which may be selected by the user when solving a given problem. It becomes extremely difficult in these instances to represent within the system all possible paths which the user might select. The system is accordingly forced to select those strategies which it considers the most likely to be chosen. The result of this is that the system represents a limited number of conceptual frameworks and accordingly, regardless of the user's strategy, tries to coerce the user into one of the representations known to it.

3. There is no consistent body of principles for tutoring strategies in the design of IT systems.

This problem results from the difficulty of abstracting and making accessible the knowledge that human tutors possess. Furthermore, it is difficult to make this knowledge explicit for incorporation into a computerised system.

4. Restricted user interfaces

User interaction, as in other areas of computing, is restricted by interface constraints. For example, those imposed by difficulties in natural language understanding and restricted input and output facilities.

3.10 CONCLUSION

Despite the opportunities presented to computer-based instruction over the last fifteen years by the advent of artificial intelligence techniques and hardware/software improvements, the field of Intelligent Tutoring System design has still not reached maturity. This is because of the difficulties involved in providing dynamic and individualised instruction.

In recent years a relatively common approach to the design of the architecture of these systems has emerged. However, the problems of how best to represent the user within the system, what teaching strategy to adopt and how to represent the teaching material remain unresolved. It is these issues which will be addressed in this thesis.

Chapter 4

AN INTRODUCTION TO THE EXPLANATION-DRIVEN, UNDERSTANDING-DIRECTED APPROACH TO THE PROBLEMS OF ITS.

4.1 INTRODUCTION

It is proposed in this thesis that a contribution towards the solution of those problems resulting from inadequate representation of the user, knowledge domain and teaching strategy described in Chapter 3 can be made by designing the tutoring system with reference to explicit, generic theoretical principles.

4.2 ASSUMPTIONS OF THE EDUD APPROACH

The principles outlined in this work for the improvement of ITS design are based on the following assumptions:

ASSUMPTION 1. For any intelligent communication to succeed, the information-communicator requires some understanding of the intended recipient of that communication. If effective knowledge communication is to take place between system and user , then there is a need for the system to have some form of internal representation of the recipient of that communication.

The basic argument of this research is that any system designed for the purpose of enhancing the processes of learning and understanding in the user must be equiped with an adequate, dynamic representation of the user's level of cognitive functioning. This representation must be based on a clear conception of the relevant psychological processes occuring in the user, as well as a representation of the ways in which these processes may be enhanced.

If such a conception is achieved then the problem outlined in chapter 3 of determining the level of detail of instructional material that needs to be communicated to the user is ameliorated. Once the user's level of cognitive functioning is assessed and represented within the system then the selection of the instructional material which will serve to drive the user from a particular level of cognitive functioning to a higher level of functioning may be based on this assessment.

An objective of the research has been, therefore, to provide a cognitive theory of the learning/understanding process taking place in the user and to specify a set of explicit principles from which a clear diagnosis of the level and extent of user functioning could be derived and implemented in a User Model. This objective is in contrast to the approach commonly taken to current intelligent tutoring system design (with the exception of the ACTP tutors)(And85) where the basic design strategy has originated either from task-dependent educational needs or as an attempt to improve existing Artificial Intelligence techniques but not from an attempt to define an explicit theory of learning.

ASSUMPTION 2. No effective learning can take place in the absence of understanding.

Throughout this research the importance of the role of understanding as an integral part of the learning process is stressed. Whilst the possibility of achieving some understanding of a subject through rote learning is not denied, the focus of this research has been on the ability of computer-based instructional systems to communicate knowledge about some selected domain leading to an expert level of performance. This approach is based on the belief that acquiring knowledge at an expert level cannot be achieved by rote learning but requires understanding of the meaning, capabilities and relations of and between a diverse number of domain concepts.

An objective of this research has been therefore to devise a systematic method for assessing and increasing the user's understanding of a given body of domain knowledge by providing him with appropriate explanation facilities. The justification for the selection of the explanation framework has been made with reference to scientific theory (Ach83).

ASSUMPTION 3 - The identification of consistent principles on which to base the design of tutoring systems provides both tutoring and critiquing strategies and a method for knowledge decomposition (Sle82).

Once a clear representation of the teaching principles to be employed has been achieved, analysis of the target-knowledge may be undertaken in such a way as to reflect these teaching principles.

Achievement of this objective constitutes a contribution to the solution of the problem of ad hoc teaching strategies adopted in the current generation of ITS design. The advantage of common principles for a teaching/learning strategy is that they may be incorporated in an ITS design for a number of domain applications (Nec87). The framework described in this research is not constructed, therefore, from the examination of a particular domain but considers principles which may be incorporated in the teaching of a number of domains.

4.3 THE EXPLANATION-DRIVEN, UNDERSTANDING-DIRECTED MODEL

Whilst the mutual dependence between ITSs and cognitive science has generally been acknowledged, little has been done in the design of existing tutoring systems to exploit this dependence.

An area where this is apparent in many existing tutoring systems is in their failure to incorporate an explicit learning theory in their design. It may be argued that the existence of an explicit user model and teaching strategy within the design of these systems is indicative of an underlying implicit theory of learning. However, to date, no attempt has been made to formalise these theories so that their commonalities and generalities may be identified. (A notable exception to this pattern is the work of Anderson, which examines a general theory of cognition with particular emphasis on skill acquisition) (And83).

The approach commonly used in existing ITS design emphasises the acquisition of problem-solving skills rather than the learning process occuring in the user. Furthermore, these problem-solving skills are viewed as idiosyncratic to a specific task. Present system design, therefore, examines a task and infers the underlying psychological processes necessary for its problem solution. Specific problem-solving skills are thus firmly linked to a specific task and it is these skills which are tutored with the help of the system. Psychological domain-independent arbitrary rules that underlie the reasoning process during learning are assumed but not explicated even though the user's ability to perform a given task is seen to result from the application of such rules.

It is felt, that as an Intelligent Tutoring System is to present facts in a way that the user can recall and apply, then these systems should be designed with greater emphasis on the user's understanding of these facts rather than improving particular problem-solving capabilities.

The system must therefore not only be capable of justifying its reasoning processes in terms of the problem-solving knowledge utilized at each step but also of explaining the underlying knowledge in a way which facilitates understanding in the user.

4.3.1 Explanation-Driven Understanding-Directed Learning

In the EDUD model, understanding is achieved by the development and restructuring of cognitive structures using appropriate explanation facilities. The goal is to obtain a hierarchically organized set of functionally related information structures relevant to a particular problem domain. There are several stages of transformation through which the relative cognitive structures must pass before an understanding of the subject can be claimed. It is held, that the order in which these stages is traversed is loosely sequential. It is the evolution of such cognitive structures which is emulated in the EDUD model. Whilst the model does not claim that the evolution of the stages it identifies are necessarily correct, it nevertheless claims to approximate it sufficiently and in terms amenable to computer implementation.

Each stage represents an essential level in the hierarchical structure aimed at. The criterion behind this is simple, namely , the user cannot fully understand stage x before he understands something of stage y. The model identifies the following stages of understanding through which the individual passes in the problem-solving process. Each stage is identified with a particular abstract, domain-independent or domain-dependent rule.

1. The Stage of Figurative Knowing
2. The Functional Stage
3. The Cause/Effect Stage
4. The Complex Derivational Stage
5. The Stage of Hypothetico-Deductive Reasoning

Such a hierarchical goal structure has particular benefits for tutoring purposes and system design. Once the level of understanding at which the user is functioning is identified the system can offer the necessary explanations which will drive him to the next level in the hierarchy. Explanation and understanding are seen as two necessary and interconnected components of knowledge communication.

4.3.2 Pragmatic Rules of Inference

The difficulty of establishing a single domain-independent theory of learning is acknowledged. However, recent research into how people reason has produced results which are of interest for Intelligent Tutoring System design (Nis87). It has been shown that in addition to the domain-dependent rules used in solving a particular task, people also use abstract, domain-independent inferential rules to think about everyday events. These rule systems are abstract inasmuch as they can be used in a wide variety of content domains. (However, their use is confined to certain types of problem goals and relations between events).

The EDUD model identifies some of these abstract domain-independent rules and demonstrates how they can be incorporated into the design of the Tutoring Module of an ITS.

4.4 IMPORTANCE OF THE EDUD MODEL FOR THE DESIGN OF THE USER MODEL IN ITSs.

In the EDUD model the user's understanding of the material presented to him is assessed at each level in the hierarchy. Based on this assessment a decision is made as to what material should be explained to him next. The goal is to pinpoint the abstract relations which are understood at a specific phase of understanding and consequently 'predict' those relations which still need to be tutored and those which must have already been mastered in order to achieve the present level of understanding.

Because the process of understanding is viewed as sequential, if at any level in the hierarchy the user is functioning successfully all previous levels may be assumed as successfully completed. The levels in the hierarchy thus form the basic framework for monitoring the progress of the user's understanding.

4.5. IMPORTANCE OF THE EDUD MODEL FOR THE ORGANISATION OF DOMAIN KNOWLEDGE FOR ITSs

The typology of explanations provided in the EDUD model affords a framework for the sub-division of the body of target knowledge. The target knowledge is decomposed so as to fill each explanation category by applying the pragmatic rules of inference to the task domain as follows:
What is X?
What does X do?
How does X achieve this function?
Why does X achieve this function?
If X then.....?
The information supplied by responding to these queries is then placed in the taxonomical framework and addressed by the node name.

4.6 IMPORTANCE OF THE EDUD MODEL FOR TEACHING STRATEGY

The EDUD model is based on the assumption that the most realistic approach to ITS design is that it be based on a clear hierarchy of levels of understanding, rather than on problem-solving skills specific to a particular task. By providing a hierarchy of understanding, boundaries for the user's knowledge are provided and a focus for pedagogical activities is suggested. Accordingly, Intelligent Tutoring Systems should be capable of operating in distinct phases, each phase designed to support a level, or levels , in the process of understanding.

In the first phase the objective is to teach the user the necessary declarative knowledge to provide a conceptual framework for subsequent problem-solving. In the EDUD model declarative knowledge is acquired during the initial three stages of understanding. Accordingly,the system should operate in an instructional way with a directed learning strategy in which it takes the initiative. In the second phase the object is to assist the user in converting and re-organising his declarative knowledge in the context of procedures by setting problems to be solved. The system should therefore provide a set of appropriate problems for solution. Finally, support should be provided for an articulate expert system, to provide a model of heuristic problem-solving and decision making skills that are acquired in the final stage of the understanding hierarchy.

4.7 CONCLUSION

As a solution to the stated requirements of an effective tutoring system the Explanation-Driven, Understanding-Directed (EDUD) model has been developed. This model incorporates a set of explicit, domain-independent concepts and emphasises the importance of the indissociable concepts of understanding and explanation in the design of Intelligent Tutoring Systems.

The practical result of the incorporation of domain-independent concepts in the design of the User Model, Knowledge Base and Tutoring Module is that they may be used as the framework for an ITS shell.

Chapter 5

A GENERIC APPROACH TO AN INSTRUCTIONAL SYSTEM DESIGN

5.1 INTRODUCTION

The objective of this work has been to develop a set of independent principles for ITS design which would serve as a framework for a shell.
This chapter of the thesis considers:

i. argument for the inclusion of domain-independent principles in the design of Intelligent Tutoring Systems,
ii. existing Artificial Intelligence approaches to the design of generic tools,
iii. generic techniques and tools provided for the design of Computer-Aided Instruction,
iv. existing ITS approaches to providing generic principles,
v. the User Modelling Front-End System (UMFE) (Sle85) which attempts to build a domain-independent modelling subsystem for an ITS and
vi. the role of domain-independence in the EDUD model

5.2 ARGUMENT FOR THE INCLUSION OF GENERIC PRINCIPLES IN ITS DESIGN

It is generally accepted that tutoring system design needs to move away from individually 'hand-crafted' approaches suited only to a particular application to the use of general purpose tools. From a computational perspective such generalisation of principles would curtail the number of procedures that have to be individually coded (Cla87).

If the design of general purpose tools is to be achieved then questions concerned with basic principles such as those of the nature of knowledge, communication, learning and understanding need to be addressed (Ohl88 , Law87, Wen87). Explication of these topics will lead to domain-independent 'frameworks' from which general purpose tools for constructing tutoring systems may be developed (Wen87). In this thesis the basic principles on which design is based are grounded in the disciplines of pedagogy, psychology and artificial intelligence. Argument for the inclusion of principles from these particular disciplines is produced in Chapter 6.

5.3 THE AI APPROACH TO PROVIDING GENERIC TECHNIQUES

Artificial intelligence provides a number of domain-independent tools to facilitate the construction of expert systems. These include, inter alia, knowledge acquisition techniques as well as formalisms for knowledge representation and reasoning. In addition, expert system 'shells' are provided which are static, domain-independent frameworks which provide syntactic procedures for the insertion of application-specific information. The inserted information is then manipulated by the system in accordance with pre-specified reasoning methods. The advantage of these shells is that they allow the user to concentrate on what the system needs to know, in terms of the content and structuring of the prerequisite knowledge, rather than on how the system will handle that information internally.

5.4 CAI APPROACHES TO GENERIC SYSTEM DESIGN

CAI provides domain-independent tools in the form of 'authoring languages, for example, TICCIT (Ald79). An authoring language is a framework within which teachers, who do not have complex programming skills, can compose instructional interactions.

5.5 ITS APPROACHES TO GENERIC SYSTEM DESIGN

The objective of ITSs, namely, to provide a framework which captures the knowledge that allows the teacher to compose an instructional interaction in the first place is, by definition, domain-independent (Wen87). Clearly, the difficulties of achieving such an aim are formidable. However, despite the fact that, to date, few domain-independent IT tools have emerged to facilitate the achievement of this objective, improved understanding of model building is beginning to provide the skills which will enable knowledge bases to be created for several domains (Sle87b). Evidence of this is provided by a group at Xerox Palo Alto Research Centre who are working on an authoring aid, the Instructional Design Environment (IDE), which will provide a collection of tools to aid the design of instructional material (Bur88).

The fact that domain-independent tools for ITSs remain scarce, may be seen as a result of the number of constraints that must be observed in designing domain-independent tutoring tools. A principle constraint is that in defining the principles on which to base these tools the goals of the educational interaction must be clearly set, as principles arising out of a single viewpoint cannot necessarily be generalised to all educational settings (Wen87). Once the goals have been set, and the principles identified, it is further necessary to stipulate which components of the tutoring system these generic principles are applicable to. Whilst ITS architecture commonly distinguishes four basic components in system design - the Tutoring Module, the User Model, Interface and Domain Knowledge, generic analysis is usually undertaken on one of these components. For example, whilst a number of domain-independent teaching strategies have been incorporated in the design of Tutoring Modules, few generic principles for discourse procedures have emerged. Significant attempts to provide principles for domain-independent User Models have been made in the design of PIXIE (Sle87b) and UMFE (User Modelling Front-End Subsystem) (Sle85). UMFE is discussed in detail in Section 5.6 of this chapter.

A recent attempt to provide a full ITS which is domain-independent is Challenger (Siu89). However, whilst the domain knowledge for this system can be changed to teach a number of subjects, it is limited to equation-oriented domains (calculus and physics, for example).

5.5.1 EXAMPLES OF EXISTING GENERIC MECHANISMS IN ITS DESIGN

In general, where attempts have been made to incorporate domain-independent principles in the development of an IT system, these have taken the form of specialised 'mechanisms' incorporated to achieve a specific purpose. For example, domain-independent rules for determining which set of problems should be generated for diagnostic purposes are found in LEEDS Modelling System (LMS)(Sle82).

A set of domain-independent heuristics - the generative principles for 'repair mechanisms' are found in REPAIR theory (Bro80). (A repair mechanism may be seen as a 'unit' of behaviour found in a general purpose reasoning procedure. The philosophy behind the theory is that when an individual comes to a point in a reasoning procedure where they are at an 'impasse', they resort to a 'repair'. Examples of repairs are skipping a step, or replacing an operation by an analogous one).

A set of domain independent discourse procedures for guiding the instructional dialogue are found in the GUIDON program (Cla87). In this system, it is the teaching knowledge which is viewed as a domain-independent form of expertise that can be codified in a knowledge base in its own right. A discourse procedure represents a 'recurrent conversation' in teaching, about how to interpret student behaviour and how to respond to a student. This network of procedures provides a representation for organising heuristics for carrying on a structured dialogue.

A domain-independent discourse strategy which is compiled with a domain-specific language-generator and attempts to formalise the type of discourse procedures developed in GUIDON is found in MENO-TUTOR (Woo84) which provides a general framework within which tutoring rules can be defined and tested.

5.6 UMFE:A User Modelling Front-End Subsystem

UMFE is a significant attempt to write a portable domain-independent modelling system (subsystem) which infers overlay models for users. (Sle85).

The intention in this system is to serve as a front-end to a knowledge-base by filtering responses to user's queries in terms of those concepts which UMFE believes the user understands.

The system is domain-independent in that once difficulty and importance ratings have been assigned to concepts, the inference rules to be applied are algorithmically determined. Thus, for each concept UMFE determines the mean difficulty and then asks the user whether he understands the significance of this concept. If the user states he does then the algorithm continues the search in the range of the most difficult concepts, alternatively if he does not understand the concept the search is performed in the lower range. The system's belief as to the level at which the user is functioning is arrived at by comparing the user's response on a particular domain-concept with an expert's predetermined rating, associated with that concept, in terms of its difficulty or importance on a 1-10 scale, for a particular population. Because the user's competence is represented as a subset of the expert's this system may be viewed as an 'overlay model'.

A number of limitations of this model have been observed (Wen87). These limitations include the fact that the knowledge base requires a new set of rules to be defined for each domain and each class of users. In addition, the system imposes a uniform conceptualisation on modelling by utilising a fixed set of inference rules.

The view taken in this thesis and which is addressed in the design of the EDUD model is that the major shortcoming of UMFE is that it bases its assessment of concepts on a 'difficulty factor'. 'Difficulty', is an inherently relative concept and what may be considered to be difficult by one user may not be so for another. It is probably impossible to devise a universally acceptable classification of concepts according to some scale of difficulty.

5.7 THE EDUD APPROACH TO DOMAIN-INDEPENDENT SYSTEM DESIGN

In the design of the EDUD model the assignment of difficulty measures to concepts has been avoided on the grounds that 'difficulty' is a subjective concept. Instead of reflecting the difficulty level at which the individual is functioning, it is primarily concerned with demonstrating the user's level of understanding of a particular subject-domain. The purpose of the EDUD User Model is to identify the 'understanding' processes taking place in the user, and on the basis of this identification to determine the sequence of the teaching action required by the Tutoring Module to move the user from one level of understanding to the next.

Accordingly, the EDUD model specifies content-free layers of understanding, arranged in a hierarchical structure, which is accessed through the application of abstract pragmatic rules of inference (These rules are described in detail in Chapter 10).

An advantage of using this domain-independent structure is that a learning objective which describes the level of proficiency a user will be expected to perform and under what conditions, may be associated with each level in the hierarchy. In this way the hierarchy of learning objectives,in essence, defines the domain.

The practical benefit of specifying the hierarchical structure of understanding and pragmatic rules at an abstract level is that it allows these elements to be used as the framework for an ITS shell.

However, it must be emphasised that there are constraints on the extent to which the EDUD model may be considered domain- independent. It is perhaps more accurate to describe the design as being 'somewhat' domain-independent as it is only applicable to what Clancey describes as those areas which use non-numeric models and combine them with heuristic procedures for solving practical problems, for example, medicine, engineering and business (Cla87). In other words, it is suitable for application to those areas which require an analysis and synthesis of a real world system. It is not suitable for teaching domains such as language or mathematics.

In this research medicine and engineering have been selected as the teaching domains. The choice of these two 'domain areas', from which to draw illustrative problems for analysis in this work is intentional because these areas provide tasks which may be represented at a level of abstraction which is neither too abstruse nor too concrete. The task-requirements in these domains may be considered as 'relatively difficult' to analyse. They are not so constrained as is, for example, arithmetic, where it is not difficult to specify certain basic arithmetic operations (such as addition, subtraction, multiplication, division) and delimit the range of problems to which the operations applied. On the other hand they are not as complex as domains such as language.

In addition, the domains of engineering and medicine are amenable to analysis in terms of the intellectual skills required to perform their objectives and a primary aim of this research is to specify generic principles rather than performing task analysis. It therefore moves away from domains defined strictly by content into domains defined by intellectual skills. This complicates the picture because the intellectual skills must be defined first and then exercises developed to reflect these skills.

5.9 CONCLUSION

In order to achieve any measure of generality the output of research into computerised teaching should consist not of particular systems and instructional artifacts but of principles which enable specifications to be constructed in terms of instructional content and student characteristics. These specifications may then be used as frameworks for the design of ITS shells.

Accordingly, in this work a set of generic principles on which the design for an ITS shell may be built is provided. These generic principles provide the framework on which the EDUD User Model is based which allows a profile of the user's level of functioning to be constructed and is applicable to a number of domains; a generalised method for the division of instructional content into assimilable units which are accessed through the application of abstract pragmatic rules of inference; and a tutoring strategy for the presentation of the selected instructional material.

Chapter 6

THE NEED FOR A MULTI-DISCIPLINARY APPROACH TO INTELLIGENT TUTORING SYSTEM DESIGN

6.1 INTRODUCTION

The design of general purpose tools for ITSs requires the explication of basic principles from salient disciplines which will provide frameworks from which these tools may be constructed. The view taken in this thesis is that of these disciplines, psychology, pedagogy and artificial intelligence are the most significant to ITS design. This view is based on the assumption that psychology provides insight into how individuals learn; pedagogy provides guidelines for applying and evaluating these insights in a teaching situation and finally artificial intelligence provides the means for accommodating the findings and lessons gained from the latter two areas in a purpose-built teaching-learning environment.

In this chapter more attention will be paid to the need to incorporate psychological theory in ITS design, than the other two disciplines. This decision to stress psychological theory is on the basis that pedagogics is linked to psychological theory in an indissociable way, that is, once a clear idea of the learning processes taking place in the individual has been formed, decisions as to the most effective teaching strategy may be linked to these processes. Artificial intelligence requires adequate psychological theory, on which to base human-emulating system design.

6.2 CURRENT APPROACHES TO INTELLIGENT TUTORING SYSTEM DESIGN

The desirability of incorporating psychological theory into the design of Intelligent Tutoring Systems is generally acknowledged. Despite this acknowledgement, to date, only the ACTP Tutors, have been based explicitly on an underlying cognitive theory, namely, the Adaptive Control of Thought theory (ACT) (And85).

It may be argued that in those systems where there is no explicit identification of an underlying cognitive theory, that the existence of an explicit User Model and teaching strategy within the design of these systems is indicative of an underlying implicit theory of learning. The view expressed in this work is that such an approach is insufficient and that it would be beneficial to the field of ITS if an attempt were made to formalise these theories so that generalities in the learning process may be identified.

Some of the more important reasons for the lack of inclusion of explicit psychological principles in the design of existing Intelligent Tutoring Systems become clear when the original motivation of the designers of these systems is considered. Amongst the more common current reasons for undertaking to build intelligent tutoring systems are:

1. interest in the application of artificial intelligence techniques to teaching,
2. perceived task-specific educational needs and
3. research on human problem-solving skills

6.2.1 The Influence of Artificial Intelligence on the Design of Current ITS

Issues specific to human cognition which are essential ingredients of successful communication with people are often absent from AI work (Wen87). The absence from AI work of these issues, which are the subject of psychological studies, can be explained not so much as a reluctance on the part of the artificial intelligence community to accept psychology as a valid source of wisdom, but more as the result of the fact that the designers of AI systems tend to emanate from computer-based disciplines. This educational background in computer science leads to an approach to system design which is primarily computer-orientated and therefore focuses on aspects of design such as programing techniques, rather than a more cognitive-orientated approach which would be more inclined to stress and incorporate psychological considerations in the system design.

It follows, therefore, that in those cases where the motivation for the design of Intelligent Tutoring Systems has been governed by a desire to test and improve existing Artificial Intelligence principles, the result has been that psychological principles have typically been absent from these systems.

6.2.2. Intelligent Tutoring Systems originating from Perceived Task-Specific Educational Needs

Another motivating factor for the design of IT systems is the perception of a task-specific instructional need that can be well served through computer-based teaching systems. In this case the approach to system-design is to examine the specific task and decompose it into a number of problem-solving units. The psychological processes necessary for the solution of each unit are inferred and it is these (inferred) processes which are then taught with the help of the system. Whilst this is certainly a worthwhile approach to system design, when followed too rigorously and exclusively it leads to unwarranted emphasis being placed on task decomposition and insufficient attention being placed on the underlying psychological processes which promote the understanding of the task. It is the view expressed in this work that the reverse of this strategy should be adopted. Initially, an examination of the cognitive processes taking place in the user should be undertaken and it is in the light of these processes that the task should be decomposed. In other words, the emphasis in tutoring system design should be as much on psychological processes as their product.

6.2.3. The Influence of Research on Human Problem-Solving Skills

Research into human problem solving skills, in particular the work instigated by Newell & Simon (New72), has had widespread influence on and provided valuable information for both artificial intelligence and ITS design. This work stimulated substantial research efforts which concentrated on methods for implementing computer applications, the objective of which was to emulate the human problem-solving process.

Whilst the importance of teaching problem-solving skills through computer systems is acknowledged , it is felt that equal attention should be given in designing these systems to the teaching of the basic understanding of the target-knowledge. This view will be discussed more fully in Chapter 8.

6.3 THE ROLE OF PSYCHOLOGICAL THEORY IN INTELLIGENT TUTORING SYSTEM DESIGN

It is held here that there are three major reasons for the incorporation of psychological theory in ITS design. First, psychological theory should be incorporated in any computer system design which purports to emulate human reasoning processes. The aim of the User Model in an ITS is to reflect the learning processes experienced by the user. Accordingly, it is these human processes that need to be clarified and emphasised in its design.

Second an explicit psychological theory is also important to curriculum design. Once psychological principles have been formulated, they provide clear directions for the content and sequencing of instruction (Gag62). To date ITS have tended to represent the target knowledge of the instruction explicitly but not to represent explicitly that body of knowledge which specifies the goal structure for instruction.

Third, it is by examining psychological processes that the possibility of describing a generic body of knowledge which identifies both consistent and first principles for computer- based instruction is facilitated. The argument for providing generic principles in system design is described fully in Chapter 5.

In practice,however, there are difficulties in including findings from psychological research in ITS design. The reasons are, briefly, these.

The research findings from psychology are directed towards providing a measure of the observable changes in behaviour consequent to the learning process. To date, however, no measurable manifestations of cognition have proved sufficient to provide evidence of complex changes in cognition such as result, for example, from increased understanding. It therefore becomes necessary to incorporate those findings which psychology has to offer but to look elsewhere for those answers which it cannot provide. It is useful, for example, to turn to philosophy to provide guidelines for the definitions of concepts.

These definitions may then be incorporated into a plausible theory of a process of learning which is amenable to computer implementation. This is the aim of the EDUD model.

6.4 ARGUMENT FOR THE INCORPORATION OF PSYCHOLOGICAL THEORY IN THE DESIGN OF THE USER MODEL

As evidence for the need for the explicit incorporation of psychological theory in the original formulation of the User Model the following section considers difficulties experienced by the DEBUGGY (Bur82), REPAIR (Bro80) and PIXIE (Sle87b), systems which document problems encountered as a result of the omission of such theory from their initial design.

The DEBUGGY system was designed to extend the ability of the BUGGY (Bro78) system from that of accounting for student errors in simple procedural skills associated with number subtraction, to include diagnosis of these errors as well. Although DEBUGGY can claim success in its ability to diagnose bugs it cannot explain the origin of these bugs within the user's conceptual framework (Wen87). This inability stems from the absence of an explicit underlying cognitive theory. Without such theory the results of the diagnostic procedures are without a yardstick with which they can be compared and explained in any meaningful sense. Any explanation provided by the system of the origins of a bug must, therefore, remain syntactic.

In order to enhance DEBUGGY's explanatory power as to the genesis of bugs REPAIR theory was proposed. This theory although syntactic in nature, does view errors in a student's procedure as arising from the incorrect application of a 'core procedure'. The authors do not however attempt to specify the semantic nature of the core procedure by explaining its genesis in cognitive terms, but instead they provide what they call a model of the 'rational genesis' of bugs. This model comprises a system of procedures, formalised in a graphical representation which, as in DEBUGGY, attempts a syntactic explanation of bugs. This solution to the problem of the genesis of bugs is, however, unsatisfactory. The authors themselves state that their dissatisfaction with the system stems from the use of syntactic procedures without reference to any underlying psychological model. In addition, it has been pointed out that these procedures lead to results which are psychologically implausible, and give rise to unbounded search spaces as a result of their limitations not being defined by an underlying framework (Wen87).

A further example of a User Model which encountered difficulties as a result of the omission of an underlying psychological theory is that incorporated in the PIXIE system. PIXIE is an algebra tutor which can generate new mal-rules to explain behaviour which cannot be described by existing mal-rules. (The term mal-rule refers to a variant on a correct rule). Once again the exclusion of an explicit cognitive model has resulted in a number of limitations (Wen87). In this instance the system experienced difficulties when it attempted to produce generative mechanisms for 'mal-rules'. In an attempt to correct these problems syntactic solutions were proposed. However, the lack of an underlying framework resulted in mechanisms that were underspecified and it was not clear how they should be implemented. Further,the program did not seem to have any way of preventing the inference of transformations that were at the wrong level of granularity(Wen87).

6.5 ARGUMENT FOR INCLUDING PSYCHOLOGICAL PRINCIPLES IN THE DESIGN OF THE TUTORING MODULE

The need to base teaching strategy on psychological principles has been acknowledged (Blo56). Omission of such theory leads to systems which are inflexible in their presentation of the teaching material and have limited capabilities for adaptive diagnosis and feedback (Kea87). This is a criticism of Computer Aided Instruction systems which ignore the implicit processes taking place in the user and thus provide no basis for diagnosing learning difficulties or for remedial instruction. To provide flexible and adaptive instruction requires the user to be viewed as an 'information processing system' in which he plays an active role in the teaching process.

6.6 ACT* - An Example of a System Based on an Explicit Cognitive Theory

So far only one group of researchers have claimed there is a close linking between a psychological theory (ACT*) (And83) and the design and implementation of their ITS (ACTP tutors) (Yaz87). The ACTP tutors (GEOMETRY and LISP), are the implementation of a cognitive theory which was developed independently of tutoring system design. The rationale for implementation of ACT* theory into a tutoring system was on the basis that a computer system would provide a good medium for testing the theory. It is not surprising, therefore, that in the light of the initial attention paid to an explicit cognitive theory, that the results of this work are impressive (Wen87).

However, the implementation of ACT* theory has been criticised on the grounds that it results in a directiveness which is best suited to novices in well-structured domains (Wen87). The view taken in this work is that this shortcoming results from an assumption made by the theory that skill acquisition should be for the most part given in a problem-solving context. Whilst this assumption may be true for instruction at novice level, and suitable for a 'learning-by-doing' environment, it is not appropriate for teaching aimed at encouraging expertise. If it is expertise that is ultimately desired then emphasis should be placed on increasing the user's understanding of the domain knowledge and its associated inference strategies and not problem solving skills.

A further criticism of ACT* is on the grounds that any claims it may have to universal applicability may only be assumed (Sle87b). This criticism is one which all theories based on universal psychological principles are prone to and to date no research has adequately refuted this criticism.

6.7 DEFINITION OF COGNITIVE DIAGNOSIS

It has been stated that one of the aims of this research has been to provide an explicit cognitive theory on which to base, inter alia, the design of a User Model for ITS. A purpose of the theory is, therefore, to provide a framework to serve as a methodology for the 'diagnosis' of the user's level of cognitive functioning which may then be implemented as an effective and accurate User Model within the system.

Throughout this work the concept of 'cognitive diagnosis', refers to the description of specific mental processes occurring in a particular individual with respect to a particular task (Ohl88). This definition of cognitive diagnosis differs from the more common use of the term found in the ITS literature where it refers to those methods of diagnosis which are concerned with discovering how a particular incorrect answer was produced by a user on a particular problem-solving task. The cognitive processes this research is concerned with do not appertain to how incorrect responses are generated in a problem-solving situation, but rather with how the individual reasons about his environment in a non problem-solving way and how this leads to his increased understanding of a domain.

6.8 INCORPORATION OF PSYCHOLOGICAL THEORY IN THE EDUD USER MODEL

The examples sited above make it clear that if a teaching system intends to provide diagnostic capabilities in such a way as to afford guidelines for teaching and remedial teaching purposes then it needs to be capable of generating a description of the cognitive genesis of learning procedures. The EDUD model described in this work provides such a description. It details cognitive structures which are internal representations of external reality and which are, in turn, generated by a set of abstract inferential rules that underlie the reasoning process during learning and serve as triggering mechanisms to elicit information from the environment. These mechanisms are psychological in nature and are mentally formulated to seek explanations that will increase the individual's understanding of a domain.

6.8.1. Advantages of the EDUD User Model over those Based on Bug Diagnosis.

Because the objective of the EDUD User Model is the interpretation of the user's responses as a reconstruction of his underlying reasoning processes as defined in the EDUD hierarchy of understanding, it is not directly linked to bug generation and, therefore, avoids the problems inherent in enumerating bugs encountered by systems such as REPAIR and PIXIE. By providing an explicit theory of learning rather than bug generation the EDUD view is that any bugs generated are the result of failures of the learning process. Once the learning process is described, errors may be directly linked to a stage in the learning/understanding process rather than attempting what is the almost impossible task of enumerating every conceivable misconception the user might have about a particular problem. Thus, any diagnosis of the user's observed behaviour is based on a reconstruction of his reasoning processes and therefore provides a semantic rather than a syntactic interpretation of his responses. For example, the system may infer that a user offering the suggestion that "carburettor malfunctioning is a possible problem with the fuel system", has some knowledge of what a carburettor is and what function it performs.

6.9 THE ROLE OF PEDAGOGIC THEORY IN ITS DESIGN

The aim of a teaching system is to provide appropriate instruction which will serve to increase a user's understanding of a particular body of target-knowledge. Accordingly, some method for curriculum design is required which achieves internal coherence and linearity in instructional sequencing. Omission of such theory leads to difficulties of lack of coherence in the instructional interaction. An example of a system which has experienced such difficulties is WEST (Bur82). The WEST tutor contains a method for determining how close to optimal a player's performance is and a set of issues to be considered in the light thereof. These issues constitute part of a curriculum knowledge structure but fail to have any relational structure tying them to each other or to a representation of target knowledge (Les88).

6.9.1 Incorporation of Pedagogic Theory in the EDUD Tutoring Module

In the EDUD Tutoring Module the teaching strategy is based on principles of psychology, epistemology and philosophy. Psychology and epistemology specify principles of learning which describe the evolution of cognitive structures into an organised conceptual hierarchy. The content of the levels in the structure are derived from philosophy of science which specifies categories of explanation.

The evolutionary development of the cognitive structures, progress in a more or less sequential 'linear' order. Instructional sequencing is then based on this hierarchy which purports to emulate the human process of understanding. For example, teaching the definition of what an object is,precedes teaching what it does, and what something does is taught before teaching how it does it.

In addition, EDUD theory describes 'triggering mechanisms' which are conceptual devices used by the individual in reasoning about his environment. These devices describe the questions asked by the individual in his attempts to increase his understanding of the domain. These triggering mechanisms are the 'glue' that tie the relational structure together, thus providing internal coherence within the structure and avoiding the difficulties of lack of coherence, described earlier, experienced by systems such as WEST.

6.10 THE ROLE OF ARTIFICIAL INTELLIGENCE IN ITS DESIGN

The emergence of the techniques of artificial intelligence provided new methods for modelling processes for computer implementations. Earlier CAI systems designed without the benefit of artificial intelligence techniques, provided instructional interactions which represented their decisions in the form of programs. By using the techniques of AI, instead of decisions resulting from some knowledge, it is the knowledge itself that is explicitly represented so that it can be used in computer-based systems (Wen87). In addition, an important principle of expert system design, that of representing separately the various modules within a system, was of importance to ITS design, as it allowed the subject knowledge to be represented separately from the strategy by which it was to be taught.

6.10.1 Incorporation of AI techniques in the EDUD model

In the EDUD model, the artificial intelligence techniques of logic programming have been used to implement a small tutoring system which teaches about a motor car fuel system. This knowledge has been represented using semantic networks, frame- like structures and production rules.

However, by providing an explicit psychological theory of reasoning the EDUD model also contributes to the field of artificial intelligence. Because it focuses on the properties of the body of knowledge itself, by decomposing it with reference to principles of learning and teaching, it defines information which will be intuitive in the human teacher but needs to be made explicit for a machine.

6.11 CONCLUSION

The purpose of this chapter has been to present evidence in support of the inclusion of psychological, pedagogic and artificial intelligence theories in the design of the EDUD model.

The view has been expressed that it is crucial that the design of Tutoring Systems should originate not only from a machine-orientated perspective but that the initiative for the design of these systems should originate from other disciplines as well. For example, whilst artificial intelligence provides effective techniques for the representation of knowledge and an effective teaching system requires the explicit representation of such knowledge, the content and form of such representation and its communication to the user will be provided from other disciplines. It is as important to initiate system design from the question of 'what' it is that needs to be represented as it is to initiate it from the question of 'how' it is to be represented. Information on the requisite contents and communication strategies of the represented knowledge will be provided from non computer-based disciplines.

Chapter 7

THE IMPORTANCE OF EXPLANATION IN KNOWLEDGE COMMUNICATION

7.1 INTRODUCTION

The way in which an ITS explains knowledge is crucial to the achievement of educational objectives. In this chapter the importance of explanation in knowledge communication will be discussed. Explanation is a complex topic and the discussion which follows is not meant to be exhaustive. It does intend, however, to consider the following points:-

i. why current approaches to the design of explanation facilities used in expert systems are not suitable for intelligent tutoring systems and
ii. why philosophy of science provides plausible categories of explanation suitable for incorporation into ITS design and a description of how this is done in the EDUD model.

7.2 THE EXPERT SYSTEM APPROACH TO EXPLANATION

Early expert system design provided explanation facilities by demonstrating a simple natural language translation of a trace of the system's reasoning. This trace was augmented with the ability to respond, during the execution of the program, to 'HOW?' and 'WHY?' questions, posed by the user (MYCIN) (Sho76). These facilities were later expanded by the provision of additional explanation capabilities for how current values for various parameters were derived (EMYCIN) (Van81). In later work explanation (justification) was provided by the system for the conclusions drawn in the problem-solving process by displaying each rule that contributed to a particular conclusion together with the certainty factors associated with the successful application of that rule (EXPLAIN) (Swa81).

However, the adequacy of explanation facilities in expert systems is limited. This limitation results from the almost exclusive focus of current systems on syntactic rather than semantic considerations. An explanation associated with some form of tracing of rules that fire during the course of a system problem-solving session is insufficient to provide 'meaning' for the knowledge used. It can only provide a satisfactory explanation of the way a program has reached a conclusion. Explaining domain knowledge for human purposes requires an ability not only to demonstrate the connections between the inference steps in the problem-solving process but also to connect these steps with fundamental domain principles as justifications. This ability is more akin to general perception than to rule tracing(Hay83).

An early attempt to provide semantic rather than syntactic-based explanatory facilities was the NEOMYCIN system (Cla83). The initial design of this system called for a more psychological approach to representing expert-level problem solving which involved modelling the diagnostic strategies that clinicians use separately to modelling domain rules. The effect of this approach was that by separating out and explicitly representing problem-solving processes from domain rules, the system could show the plans and methods used by the expert in attaining a goal rather than merely providing a trace of the rules fired (Jac86).

Research into how to provide adequate explanation facilities within expert systems continues. However, in the next section consideration is given to why this research is not directly applicable to research in the design of explanation facilities for ITSs.

7.2.1 Expert System and ITS Approaches to Explanation

It is contended in this thesis that there is a fundamental difference in the purpose for which explanation facilities are provided in the design of expert and Intelligent Tutoring Systems.

In an expert system the goal of the explanation is to explicitly provide justification that the knowledge used in the reasoning process as well as the process itself is correct and appropriate. It is, therefore, concerned with communicating to the user what the system itself 'knows' in terms of how and why a particular conclusion is reached on a particular problem-solving task. Because the emphasis is on justifying conclusions arrived at by the system at expert level, the issues with which research in this area are concerned include how an expert decomposes and integrates his reasoning in a problem-solving context and how this may be articulated in system design.

In an ITS the goal of the explanation is different to that described above. In an ITS the aim is to increase the (non-expert) user's understanding of a particular domain, and to do so in such a way as to ensure that the user may recall and apply that knowledge at a later date. The explanation process is therefore part of an act of teaching which must be related to the conclusions that the user expects the system to draw and the way in which he would have drawn those conclusions himself. The basic emphasis is not, therefore, on how and why a particular conclusion was reached by the system but on what, when and how to explain the given body of target-knowledge. Thus, the primary concern of explanation in an ITS is to enhance the user's learning/understanding of the target-knowledge, a task which requires knowledge of the current epistemological and psychological characteristics of the user.

This requirement has important implications for the design of the architecture and behaviour of tutoring systems. In expert systems it is assumed that explanations will be given in response to some content-seeking question initiated by the user and that the user knows the correct question to ask. The explanation facilities in expert systems may therefore be described as playing a 'passive' role.

In an ITS on the other hand explanations must play an active role since it cannot be assumed that the user will request an explanation at the appropriate time or level, nor that the explanation-seeking question asked will be the correct one. The system must be able to determine the appropriate content and timing of the explanation which necessitates the explanation facilities being extended to select appropriate material, adapt this material to the user's level of functioning and to conduct an efficient interaction. In those instances where a question is initiated by the user, the system is required to consider, inter alia, what inferences the user is trying to draw, what his problem-solving procedure is and what representation language he is using to accomplish what tasks. The initiative for providing explanations is therefore shared between the system and the user.

Whilst these differences make it impractical to apply methods of explanation used for expert system design in ITSs it is believed that the explanation methods incorporated in the EDUD model have implications for the former since they are concerned with fundamental issues of the explanation act.

7.2.2 Expert Systems as Tutors

Many of the above mentioned inadequacies of the explanation facilities of expert systems were revealed by the GUIDON (Cla83) program which attempted to adapt an existing expert system, the MYCIN system (Sho76), (a system designed to address the problem of diagnosing and treating infectious blood diseases), for tutoring purposes. In designing GUIDON it was found that whilst rule-based expert systems provided a good basis for tutorial programs, they were insufficient in themselves for making knowledge accessible to a student. This insufficiency arose partly because the explanations provided by the program were based on the problem-solving rules and goals of the system and did not provide justification as to why a particular rule was included in the knowledge base. In addition, neither the basis on which conclusions were assumed to be correct nor the strategy behind the given goal structures was stated explicitly. In the light of these omissions it was decided that if existing expert systems were to be adapted to act as teaching systems they needed to be augmented with additional information about the levels of problem-solving strategies, the structure of domain-concepts and supporting data. The function of the levels would be to justify individual rules within the knowledge base and provide abstraction levels to organise rules into patterns.

Whilst it is important to mention the difficulties encountered in adapting expert systems to teaching systems, this thesis is not concerned with the difficulties of adapting an existing expert system for the purposes of teaching, but more specifically with the differences in the approach to the explanation facilities provided by the two types of systems.

7.3 THE RELATIONSHIP BETWEEN CATEGORIES OF QUESTIONS AND EXPLANATION

Before describing in detail how explanation is treated in the EDUD model, mention must be made of important research carried out on the classification of questions (Leh77). The work in question attempted to categorise questions in terms of their interrogative pronouns. Examples of such pronouns (keywords) are 'why', 'what', or 'where'. The posing of one of these questions by the user resulted in system identification of the requisite entity of domain knowledge being sought and offered to the user.

This approach was seen initially as promising to the goals of this research. However, a closer examination of the question categories provided revealed several inadequacies in this approach which made it unsuitable for incorporation in the EDUD model. A particular shortcoming of question categorisation is that it uses a keyword classification system which leads to the problem of 'near-equivalence' (Har88). The notion of 'near equivalence' is best demonstrated through giving an example, for instance, by considering the following three sentences where different pronouns evoke a similar response - 'WHY did you mention that?', 'HOW is it that you mentioned that?',' WHAT on earth made you mention that?' A further shortcoming of this approach is that it can only provide a set of syntactically determined responses to questions. Thus, no consideration is given to the user's level of understanding, nor to the semantic meaning behind the 'wh-format' of the question.

Accordingly, in designing the EDUD model of explanation the view was taken that it was impractical to attempt to provide a comprehensive classification of questions which the user might ask, but that the problem which needed to be addressed was what possible explanations could be given in response to a diverse number of questions. Clearly questions such as 'why is it the case that p?' differ from context to context and such context-dependent aspects must be provided for. Thus some 'theory' of explanation and data relative to which a question/answer could be evaluated was needed. This theory was provided by philosophy of science (Ach83) (Van80) and the data was context-dependent.

7.4 THE EDUD APPROACH TO EXPLANATION

The views expressed above led to the decision to take a practical yet philosophically plausible view of the role and types of explanations suitable for computer implementation which would serve to increase a user's understanding of a particular domain.

In considering explanation facilities the primary concern has been to provide methods of explanation which would enhance the user's understanding of instructional material. The notion of the explanation process adopted for this purpose is one that : enables the receiver of the explanation to be in a complete knowledge state with respect to a body of information and some appropriate set of instructions (Ach83). The concept 'in a complete knowledge state' is to be taken in this context as being equivalent to the concept of understanding. This means that in order to achieve understanding there must be explanation and explanation exists in order to increase understanding. Explanation and understanding may therefore be seen as interconnected processes. A full definition of understanding is provided in Chapter 8.

The notion of an appropriate set of instructions is introduced for those cases which arise where a question can be correctly answered in different ways by providing various kinds and amounts of information. For instance,a person might be said to understand a reply in one way but not another. Instructions are therefore provided as rules imposing conditions on answers to a question. In the EDUD model this set of instructions is based on epistemological consideration and is described in terms of the EDUD structure discussed fully in Chapter 14.

7.5 CATEGORIES OF EXPLANATION IN THE EDUD MODEL

The theory on which EDUD's explanation facilities are based derives from philosophy of science and identifies a typology of five basic types of possible explanation (Ach83). These are:

1. identity explanations
2. functional explanations
3. causal explanations
4. derivational/complex derivational explanations
5. hypothetico-deductive explanations

7.5.1 Identity Explanations

Identity explanations are non-causal reason-giving explanations which describe what something is. They may be considered as justifications for a claim rather than explanations of why such a claim is true. Examples of such explanations are 'x=y', or 'this is a book'. This type of explanation is commonly evoked in response to 'ultimate questions', that is questions which call for responses which are at the limit of explanation, such as 'what is it?', 'what colour is it?'.

Included in this type of explanation are classification explanations. Classification explanations are those in which the properties of the objects/concepts concerned are identical by virtue of definition, that is, where the property of being a p1 is identical with the property of being a p2. For example the explanation 'it is a flower', means that the object has the requisite properties which classify it as a flower.

7.5.2 Functional Explanations

Functional explanations describe the doing of a thing with regard to the purpose in hand (Wen87). They are associated with the goal or intended purpose behind the design, use or service of an object or concept. This emphasis on intention is important as it differentiates a functional explanation from a causal one. A causal explanation justifies the result of a process, whereas a functional explanation justifies the means to an end. For example, the function of a battery is independent of its role in a car or camera.

7.5.3 Causal Explanations

Causal explanations are characterised by treating a prior event as an explanation for a subsequent event. Accounts of causality are central to justification in terms of first principles and are seen to pervade human reasoning about processes. The causal relationship between two events may be intentional or unintentional. Thus whilst the falling of a stone may cause the glass to break, this effect may be unintentional.

7.5.4 Complex Derivational Explanations

Complex Derivational explanations contain a derivation which is somewhat different from that of causal explanations. The distinction between these two types of explanation is best illustrated by considering the different content seeking questions by which they are evoked. In the case of a causal explanation the question is 'how is proposition p1 derivable from p2?', whereas in the case of a complex-derivational explanation the question is 'why is it the case that p1?' (Ach83). Thus, in the second case, nothing causes one proposition to be derivable from others. Thus, this type of explanation may make reference to entities or properties that are not an intrinsic part of its state description (Pyl84). Accordingly the point of the explanation is not to provide a premise from which a conclusion follows, but to provide derivations which explain how a conclusion follows from a set of premises. The possibility therefore arises for a complex derivational explanation to be construed as two explanations rather than one. In this case, the two parts of the explanation are expressed in terms of the laws governing their relationship rather than in terms of any cause/effect relationship between the two parts. For example, electrical connections may be represented using general causal knowledge. However to represent such knowledge at a more generic level requires understanding of more general principles, such as those of Kirchoff's law. Thus, whilst device 1 causes electricity to be applied to device 2, this is achieved as-per Kirchoff's law.

7.5.5 Hypothetico-Deductive Explanations

These explanations refer to the content of anticipations or goals which are possible future states of affairs; none can enter directly as causes in the usual sense, but define how new potential can be realised from existing knowledge (Pyl84).

7.6 CONCLUSION

This chapter has discussed how the explanation facilities provided by Expert Systems for justifying problem-solving steps are inadequate for the purposes of an ITS. In particular, it has been stressed that in an ITS the explanation facilities must be enhanced so as to increase the user's understanding of the target-knowledge. Such a need requires a degree of articulateness which goes beyond that required for explaining problem-solving steps.

The five categories of explanation on which EDUD's explanation facilities are based have been described, namely, identity, functional, cause/effect, complex-derivational and hypothetico-deductive explanations. The advantage of providing these discrete categories of explanation is that they provide a domain-independent framework for explanation facilities on which to build an ITS shell.

In addition the categories provide a framework for the decomposition of the target knowledge which is to be explained by the system. Details of knowledge decompositions based on these categories are given in Chapter 13.

Chapter 8

THE ROLE OF UNDERSTANDING IN INTELLIGENT TUTORING SYSTEMS

8.1 INTRODUCTION

The goal of the EDUD system is to provide effective tuition by increasing the user's understanding of a domain through appropriate explanation of the relevant instructional material. In the previous chapter the role of explanation in ITS design was discussed. In this chapter the focus is on the role of understanding.

Definition of the concept of 'understanding' can be viewed as a philosophical question. However, in the discussion which follows the following facets of understanding will be considered:

i. the artificial intelligence approach to understanding which is concerned with the notion of understanding as a process of inference,
ii. the pedagogic approach to understanding which is concerned with the testing of external expressions as evidence of the internal process of understanding,
iii. the psychological approach to understanding which is concerned with its cognitive aspects,
iv. argument as to why intelligent tutoring system design should be approached from the perspective of increasing the user's understanding of a domain rather than the teaching of problem- solving skills,
v. a definition of understanding, apposite for tutoring system design.

8.2 SOME ARTIFICIAL INTELLIGENCE APPROACHES TO UNDERSTANDING

Artificial intelligence has been reluctant to concern itself to any significant degree with defining the concept of understanding other than is necessary for defining its goals and assumptions. Whilst there are good and practical reasons for taking this approach, it nevertheless becomes necessary, when systems are designed with the purpose of providing procedures by means of which knowledge is communicated, to transfer this knowledge in such a way as to achieve epistemic continuity for the user (Wen87).

Epistemology is concerned with the nature of knowledge, that is, what we know and under what conditions we know it. It provides norms or principles and specifies what questions must be fulfilled if something is to be known. These norms, principles and questions therefore need to be taken into consideration in ITS design as they will restrict or expand the methods which are appropriate to achieving epistemic continuity. If epistemic continuity is to be achieved, then some definition of the notion of understanding must be provided.

Where definitions of understanding do exist in artificial intelligence they are machine-orientated, mechanical notions which ignore the cognitive/semantic aspects of the concept. For example, Charniak & McDermott (Cha85) define understanding as a particular kind of inference, that of abduction. In their terms something is said to be 'understood' when the correct explanatory facts have been abduced. Abduction is the process that generates explanation and has the paradigm:

From: b (if a b), infer:a

Whilst this form of explanation may provide syntactically adequate solutions to the problem of providing explanation facilities in expert systems, it is insufficient for human requirements. The human world is pervaded by 'meaning' in a way in which the physical world is not. 'Understanding' in human terms is more than syntactic analysis, it refers to human actions which are accompanied by consciousness and an appreciation of values (Ric67). It is, therefore, 'understanding' as a distinctive approach to human behaviour that is required in systems responsible for communicating knowledge, thus making the syntactic explanations provided by expert systems insufficient for increasing the human's understanding of a domain.

8.3 SOME PEDAGOGIC APPROACHES TO UNDERSTANDING

Pedagogic approaches to understanding have traditionally been concerned with the relationship between understanding and the attainment of particular educational objectives. The pedagogic point of view has concerned itself with testing for the presence/absence of abilities as indicators of underlying understanding. Examples of this approach are found in Bloom's work (Blo56). This work provides a taxonomy consisting of categories for six cognitive abilities, namely, Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation. Understanding is linked to these categories in such a way that the attainment of the educational objectives described in each category is indicative of a particular level of understanding. For example, 'some understanding' of a situation or phenomenon is demonstrated by the ability to 'translate' it, that is, by describing the initial phenomenon in terms slightly different from those originally used. Deeper understanding is reflected by the ability to interpret the information, that is to summarise and explain the phenomenon previously described. Bloom's taxonomy provides support for the view expressed in this thesis that a generic approach to ITS design is possible. Whilst it is true that there is very little that is absolute about understanding (Ric83), and that measures of understanding tend to be with respect to a particular task , this does not necessarily have to be the case. The criteria provided for assessment of the understanding educational objectives in Bloom's taxonomy are in no way task-specific but wholly domain-independent.

8.4 PSYCHOLOGY AND UNDERSTANDING

Psychology, like artificial intelligence, is reluctant to examine the notion of 'understanding' in any significant detail. This is possibly as a result of the view that understanding is a psychological concept that neither requires special psychological investigation to establish it nor generates any hypotheses for experimental test (Dee76). Alternatively, it may be that psychology views the concept of understanding as synonymous with learning.

8.5 THE RELATIONSHIP BETWEEN UNDERSTANDING AND PROBLEM SOLVING IN ITS DESIGN

The impact of research on problem-solving on the field of artificial intelligence has already been mentioned in chapter 5. This school of thought defined understanding as "the internal construction of a problem space appropriate for subsequent problem solving" (Hay74).

This view of understanding as inter-related primarily with problem-solving led to an approach, frequently incorporated in existing ITS design, which emphasised the teaching of the acquisition of the problem-solving skills, associated with a specific task. Successful problem-solving was seen as indicative of the ability to construct a problem-space for a particular problem, which was in turn indicative of a state of understanding. The effect of this view has led to an approach to present system design which examines a task and then infers the underlying psychological processes necessary for the problem's solution. Specific problem-solving skills are thus firmly linked to a specific task and it is these skills which are tutored with the help of the system. Psychological domain-independent rules that underlie the reasoning process during learning are assumed but not explicated even though the user's ability to perform the given task is seen to result from the application of such rules.

This concentration on the teaching of problem-solving tasks is reflected in the teleological approach, common in current IT system design. This approach is based on a doctrine of the final cause of things in which the structure of a procedure is justified in terms of the functions fulfilled by its components in accomplishing its purpose (Wen87). The effect of this teleological approach on ITS design has been that task-specific procedures are taught as a means to an end, through the sequencing of instructions in a ' first do this, then do that to get to the desired state' format. Examples, include STEP (Van83) and REPAIR (Bro80) theory in which the meaning of procedures is represented from a teleological perspective with the help of planning nets.

The view held in this research is that teaching procedures in this way omits the underlying reasoning as to why this or that should be done. Education is more than the assimilation of a set of skills, intended to perform some particular problem-solving task. Skills on a number of diverse tasks can be developed to a very high level without much understanding. For example, a student may be capable of calculating the Net Present Value of a sum of money by using the appropriate formula, but have no understanding of what is achieved in doing so.

Further, it is held here that the effect of viewing problem- skills as being idiosyncratic to a particular task has led toward a more restricted educational environment rather than an enriched one (Rid88). The ultimate goal of an Intelligent Tutoring System should be to present facts in a way that the user can recall and apply and to encourage the acquisition of expertise. Thus, whilst problem-solving skills should be taught with the aid of an ITS, these systems should not disregard the importance of increasing the user's understanding of facts and procedures, in such a way that he may later recall them.

Expertise, in particular, requires more than mere high problem-solving skill-performance. It involves an ability to view problems from various conceptual perspectives. This has important implications for Intelligent Tutoring System design where it is an intended ultimate outcome that users will demonstrate some degree of expertise in a selected task-domain.

Accordingly, the approach taken in this thesis is to emphasise the importance of understanding a given body of knowledge, rather than the acquisition of particular problem-solving skills relevant to that knowledge. Once understanding exists the user can construct his own problem-solving heuristics. Teaching problem-solving skills takes the reverse approach. There the user is encouraged to learn problem-solving heuristics and deduce the conceptual structures supporting them.

8.6 THE EDUD APPROACH TO UNDERSTANDING

EDUD theory approaches the concept of understanding from a psychological and epistemological perspective. In considering psychological aspects of understanding the objective is to provide a clear set of assumptions on which instructional system design may be based. Epistemology is examined in order to ensure that epistemic continuation is achieved in the statement of the goals of understanding as educational objectives.

8.7 PSYCHOLOGY AND UNDERSTANDING IN THE EDUD MODEL

The meaning of understanding is set out as rigorously and as precisely as possible, for the purposes of this work, in the following way: 'understanding exists when an individual has the concept of X in such a way as to know the correct question to ask and knows that a complete content-giving proposition has been given with respect to the question and is able to go beyond the information provided' (Ham87, Ach83, Bru66).

This definition of understanding presents the concept as a tripartite phenomenon. In the initial instance, understanding entails the perception of words, events and things. That is, it concerns the understanding of 'what' the concept(s) in question consists of. In the second instance, it is the comprehension of the meaning of events and things which is derived from their inter-relationships - this is the understanding of 'why', things are related to one another. It should be noted that these two instances of understanding involve two distinct cognitive processes. Consider, for example, if a state of affairs is verbalised as 'the stone falling broke the window'. In the first instance the cognitive process involved is understanding what is said and in the second instance it is concerned with understanding the effect of the stone on the window (Ric67). In the third instance understanding involves symbolisation.

Whilst three instances of understanding are distinguished here, it is not intended that these three instances be viewed as occurring in three quite distinct processes. Rather they take place within a continuous process of moving from elementary to higher and more complex acts of understanding (See Figure 8.1).

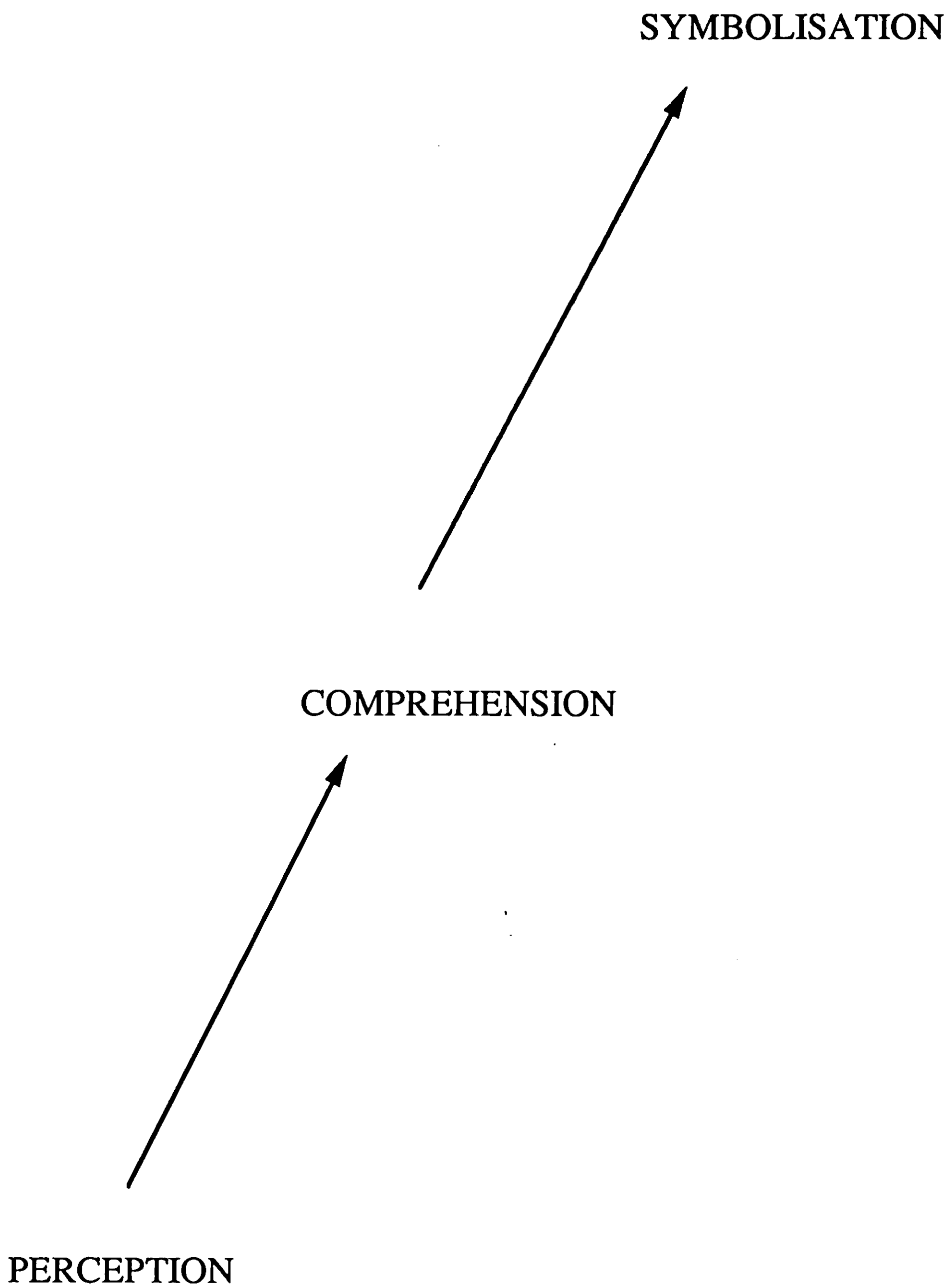

FIGURE 8.1
Understanding as a Tripartite Process

8.8 THE TRIPARTITE VIEW OF UNDERSTANDING AND IT SYSTEM DESIGN

This view of understanding as a tripartite phenomenon has important implications for the architecture of ITSs. It requires that these systems be designed with the capability of being used in three separate phases, each one aimed at increasing a particular 'instance' of understanding.

In the first phase the goal of the system is to ensure that the user has the concept of X. Concepts are acquired through definition. System design must, therefore, provide for objective definitions directed at the description of the concept and its extensional attributes. This entails conveying factual knowledge as well as principles. Procedural knowledge must also be conveyed so that both the parts of a procedure as well as how successive parts of the procedure combine to accomplish its purpose are understood.

In the second phase the goal of the system is to provide meaning to the concepts acquired in the first stage so as to enable the user to go beyond mere problem-solving performance. In most cases meaning is established through repeated experience with the associated items (Woo61). The system must therefore provide opportunities for exploration and practice in the domain by the user.

In the third stage the goal of the system is to encourage understanding through insight, more specifically the type of understanding which is inherent in creative thinking and achieved through symbolisation. To achieve this aim, however, it is essential that the first two stages are completed, as creative thinking does not take place unless there has been considerable preparation for it, with lots of research and study and repeated attempts to get the solution. Further, all the information necessary for the solution must already exist within the memory structures and must have been brought to some level of activation (Lin77). This type of understanding is necessary where the ultimate objective is the acquisition of expertise and implies an in-depth understanding of the domain that includes, but goes beyond mere skill performance.

SYSTEM FUNCTIONING	lst Stage of Tutorial Interaction	2nd Stage of Tutorial Interaction	3rd Stage of Tutorial Interaction
TEACHING STRATEGY	Expose User to Subject Matter	Provide User with Practice in Problem-Solving	Provide Facilities for Exploration of Expert Model
LEVEL OF UNDERSTANDING AIMED AT:	Perception	Comprehension	Symbolisation

Table 8.1
RELATIONSHIP BETWEEN SYSTEM FUNCTIONING, TEACHING STRATEGY AND UNDERSTANDING

8.8.1 THE TRIPARTITE VIEW OF UNDERSTANDING IN THE EDUD MODEL

To reflect the principle of understanding as a tripartite phenomenon in the EDUD model, system functioning is designed to operate in three separate phases (See Table 8.1). The first two phases of system functioning provide the user with the requisite pre-knowledge for creative thinking. In most ITS provision is made, in one form or another, for teaching aimed at increasing the first two levels of understanding. However, the knowledge required to increase understanding at the third level is often embedded in the system as expert knowledge and is presented as an inherent part of the teaching strategy followed in the first two stages. It is at this point that the architecture of EDUD departs from existing Intelligent Tutoring Systems, on the basis that teaching of expert knowledge requires, a separate environment.

The type of environment suitable for the teaching of expert knowledge is different to that of an expository, learning-by-doing or simulation type environment. In the case of an expository teaching strategy, the information presented to the user is at the perceptual level. It consists of factual information, heuristic knowledge typically being absent. In a learning-by-doing and simulation environment the user is exploring the environment in order to discover for himself its various facets. However, in the design of the EDUD model, a purely expert environment, in the form of an articulate expert system is included where the user observes, having already obtained all the prerequisite knowledge, how the expert explores the environment and how he uses his heuristic knowledge.

In this stage of interaction the system may be considered a 'passive' teacher, because it does not actively provide tuition. It is, therefore, not suitable to be used on its own without the support of the other two stages of system functioning. This is because if the user were to discover all of the knowledge considered in this phase he would have to ask an exhaustive series of questions. Some prior knowledge of the domain is therefore necessary before engaging the system in this phase of functioning. On its own this type of teaching system would be insufficient.

The intention behind the design of the EDUD model to support three phases of system functioning, is that the user will in the first two stages, reach a level of performance which may be considered as 'competent', that is, he may satisfactorily carry out the task at hand. However, he will not be considered at this stage as being expert. It has been estimated that true expertise requires about 10 years to achieve and the expert contains about 70,000 chunks of knowledge (Hay85). It follows, therefore, that whilst this level of expertise cannot at present be taught by using a computerised system, if indeed it can be taught at all, it does provide the user with a model of how an expert will apply heuristic knowledge to existing formulas, modify existing rules and reject particular laws. Further, by providing an articulate expert system all necessary aspects of expertise are open to observation, an overview of where the individual steps are leading to and what the articulating principles provided are, is made explicit. These principles may then be arranged in a teaching situation conducive to achieving insight.

The above discussion describes understanding as a tripartite phenomenon concerned with perception, comprehension and symbolism. In the discussion which follows a description of the process of epistemological understanding as the growth of knowledge in which these three instances are subsumed is provided.

8.9 EPISTEMOLOGICAL UNDERSTANDING

The development of understanding is a distinct cognitive process which is central to learning. It is a process concerned with those internal psychological mechanisms which afford external responses as expressions of the content of understanding. According to EDUD theory this process entails the development and restructuring of cognitive structures. The psychological aspects of this process are discussed fully in Chapter 9.

In this section the epistemic rather than psychological process of understanding is addressed. It concerns the question of what epistemically coherent process must be fulfilled in what mental context if understanding is to be achieved.

EDUD theory views epistemic understanding as an evolutionary process, in which external evidence and the internal operations that manipulate and reorganise the incoming information are combined. However, the acquisition of external evidence as new facts which serve to increase understanding, cannot be obtained by applying the knowledge that the individual already has and simply working out its implications or consequences (Ham78). Experience has to provide him with genuinely new information. According to EDUD theory the act of seeking such new information is through the conscious triggering of experience through the application of domain-independent pragmatic rules of inference to the external environment. These triggering mechanisms are discussed fully in Chapter 10. However, the fact which is stressed here and which is central to ITS design based on EDUD principles, is that elicited information cannot provide new experience unless it is somehow fitted by internal operations into the already existing web of understanding. For example, if an individual recognises that his bank account has changed to a new financial balance say, from debit to credit, this could not be a new fact for him unless he already knew what it was for something to be in credit. That is, understanding what being in credit is (having the concept of credit) logically presupposes understanding what a balance is, since being in credit entails having a financial balance. Thus, new knowledge it not implied by knowing what a financial balance is, but knowing what a financial balance is does presuppose new knowledge in some way. Accordingly, understanding is achieved procedurally through expansion of existing knowledge structures as a result of the incorporation of new knowledge, which is an epistemic continuation of existing knowledge.

The question then arises as to whether the process of understanding, depicted as an 'expansion' of cognitive structures is governed by any general 'law'. It is held in this work that it is. This law is demonstrated by the previous example, and states, briefly, that some kind of structure of understanding presupposes other kinds. However, it is important to emphasise that in claiming that having one concept logically presupposes having another does not necessarily have consequences for the temporal order of their acquisition, except that if A logically presupposes B one cannot be said to have A in the full sense before having B in the full sense. But it is possible for A and B to come together; alternatively since having a concept is not an all or nothing affair, one might still have A in some sense or way before having B in the full sense.

This description of the process of understanding appears to advocate that all humans learn in the same way. This is clearly not the case. Any such claim would have to ignore the role played by experience in the process of understanding. Experiences could only be identical if they followed exactly the same path of development, received identical inputs, in the identical order and used identical processes for organising them. This is extremely unlikely, particularly as it is seldom that an adult encounters an entirely novel event that is totally unrelated to his existing conceptual structure (Lin77).

Consequently the chances that two people will evolve exactly the same conceptual structure to represent the world they experience is remote. Thus, whilst it is not held here that there are universal principles of understanding which apply to all individuals, what is advocated is the identification of those invariant features of the human processing system, notwithstanding experience, which do apply. Since one of the goals of any science is to uncover invariance, the search for commonalities in the human cognitive architecture should be of central concern.

However, the identification of invariance in the process of understanding may only be based on the observation of convention. Understanding as a psychological concept does not generate any hypothesis for experimental test. It is, therefore, necessary to be content with accepting from a psychological standpoint that understanding "leads to no particular behaviour but is the inward sign of the potential for reacting appropriately to what we see and hear" (Hil77).

8.10 EPISTEMELOGICAL UNDERSTANDING AND IT SYSTEM DESIGN

The implication of this view of understanding as an epistemological process in which some understanding of A logically presupposes the full understanding of B has important implications for system design. In the EDUD model the target knowledge is subdivided into fundamental categories on the basis of the explanation categories described in Chapter 7. These categories are presented sequentially to the user during the first stage of system functioning with the objective of ensuring that the user is at no time confronted with concept B, without having been exposed to concept A thus encouraging epistemic continuity. In this way something of the meaning of each part is grasped with the presentation of each category of explanation, so that the whole derives its meaning from the meaning and ordering of the parts it contains. Thus, understanding is achieved in a controlled step-by-step way.

8.11 THE EDUD HIERARCHY OF THE UNDERSTANDING PROCESS

In the previous paragraphs, understanding has been described as an epistemic process, in which the full understanding of a concept logically implies some understanding of another. In Chapter 9 understanding is described as a psychological process in which cognitive structures evolve as they pass through hierarchical levels of understanding. The principles derived from these two processes are now amalgamated in the provision of the definition of the EDUD hierarchy of understanding. The sequencing of these stages is described from the lowest most rudimentary form of understanding to the highest, most abstract level as follows:

1. The Stage of Figurative Knowing
2. The Functional Stage
3. The Cause/Effect Stage
4. The Complex Derivational Stage
5. The Hypothetico-Deductive Stage

The level of understanding achieved in each of these stages is as follows.

8.11.1 The Stage of Figurative Knowing

In the first stage of understanding the user seeks to gain an intuitive grasp of a concept. The perceiver only takes cognizance of what is immediately apparent and obvious about a situation or concept, attention is paid to the surface characteristics. Unless he can achieve some degree of figurative knowing at this stage he will have difficulty in understanding relationships at a later stage. It would not be possible, for example, for the user to understand what causes syzygy (that is, to understand the causal stage) before the user has some understanding of what 'syzygy' means (figurative knowing stage).

8.11.2 The Functional Stage

In the next stage of understanding the user understands what the purpose or function of a concept is. The positioning of the functional stage at this level in the hierarchy is important. Functional understanding is seen as taking place subsequent to figurative knowing. This signifies that the individual understands what something is before he knows what it does. However, functional understanding precedes causal understanding. Thus the individual understands, what something does before he understand how it does it. For example, it may be understood that the function of starter motor is to start the car, but in order to understand how this is done, it is necessary to progress to understanding the causal relationships in the next level.

8.11.3 The Cause/Effect Stage

In the next stage, the user understands the pertinent cause/effect relationships between entities. Two or more independent schemas become coordinated within a new totality, one serving as instrument and another as goal. The result is that means-end relationships are understood in this stage. That is, he understands how function is achieved through cause/effect relationships. A distinction is made between direct causal relationships, that is, where there is no ambiguity about the cause/effect link and relative causal relationships, where the entity represents an object which may have causal influence under particular conditions. For example, if a user understands what a starter motor is he is then in a position to understand that its function is to start the car, but in order to know how this is done, he needs to understand the underlying electromechanical relationships.

8.11.4. Complex Derivational Stage

Once the prior definitional, functional and causal relationships have been understood, the user may understand a concept sufficiently to ensure that given an effect he may infer a cause which is not solely deductible by means of domain-specific information. Understanding at this stage could be considered as isomorphic with discovering causal relationship if the domain is considered independently of its environment. But more than this it is also the ability to understand how the domain relates to all other domains with which it is functionally linked. For example, consider a steam valve that opens and lets steam escape when the steam pressure goes over a certain limit. One designer may use this to make a high-pressure alarm by attaching it to a whistle, and another may use it as an explosion-preventer in a steam engine.

8.11.5 The Hypothetico-Deductive Stage

This is the highest level of understanding and incorporates reasoning at the expert level. At this level, given the knowledge with which he starts, the individual is capable in principle (although not necessarily in fact) of working out the conclusion from what he already knows. Thus, given an object, fully understood and a knowledge of how it links functionally with its environment, the expert understands how these links can be improved by improving the object itself. This stage is therefore characterised by the ability to make statements not only about all aspects, actual and potential of a given concept, but about the concept in general. It incorporates creative thinking.

8.12 CONCLUSION

In this chapter consideration has been given to the importance of the notion of understanding in Intelligent Tutoring System design as well as a description of the artificial intelligence, pedagogic and psychological approaches to the concept.

The view has been expressed that a clear distinction should be made between the concepts of problem-solving and understanding in approaching ITS design. The argument is that teaching problem-solving procedures omits to ensure that the underlying understanding necessary for generalising this knowledge to new situations is achieved. Thus, it is teaching aimed at increasing 'understanding' which is emphasised in this research.

The view taken has been that the process of understanding is both a psychological and an epistemological development, in which the individual moves from less to more advanced levels of understanding. It was proposed that this process is governed by a general 'law' - that some understanding of a concept presupposes other kinds of understanding. Thus, having one concept logically presupposes having another. This is not to imply that there exists a temporal order for the acquisition of conceptual structures. Rather it is proposed that A and B might 'come together' or there may be some partial understanding of A prior to having full understanding of B. In the latter case B logically presupposes A. Therefore, one cannot be said to have B in the full sense before having A in the full sense.

These descriptions of understanding as an epistemic and as a psychological process have been combined to produce the EDUD hierarchy of understanding , which is the basis of the design of the EDUD User Model . Details of this design are given in Chapter 12.

Chapter 9

THE COGNITIVE STRUCTURALIST APPROACH AND THE EDUD APPROACH TO ITS DESIGN

9.1 INTRODUCTION

The role of psychological maxims as a basis for the design of tutoring systems is emphasised throughout this research. In this chapter the importance of one such psychological notion, that of the 'cognitive structuralist' approach to learning is described (Gar78). In this work particular attention is focussed on the role which this approach plays in the design of the User Model and it is the foundation on which the design of the EDUD User Model is based.

9.2 DEFINITION OF A COGNITIVE STRUCTURE

Briefly, a cognitive structure is an internal mental representation of an external fact or phenomenon. Each cognitive structure contains a significant fragment of domain-specific information. However, these structures should not be considered as fixed, but rather as fluid entities capable of changing their shape, which they constantly do in the process of learning/understanding (Gar78). During the learning process actions and objects are assimilated to an existing structure causing it to evolve into a new structure. Thus, the existence of a new structure implies the development of a new level of understanding (See Figure 9.1). Accordingly these structures are constantly changing and expanding to reflect more complex levels of thought.

9.3 SEQUENTIAL EVOLUTION OF COGNITIVE STRUCTURES

The learning process is seen in the context of cognitive structuralist theory as a progression from less complex to more advanced states of cognitive functioning and understanding. This progression is reflected in the evolutionary growth of the cognitive structures. According to classical cognitive structuralist theory the order in which the development of these structures takes place is sequential and universal.

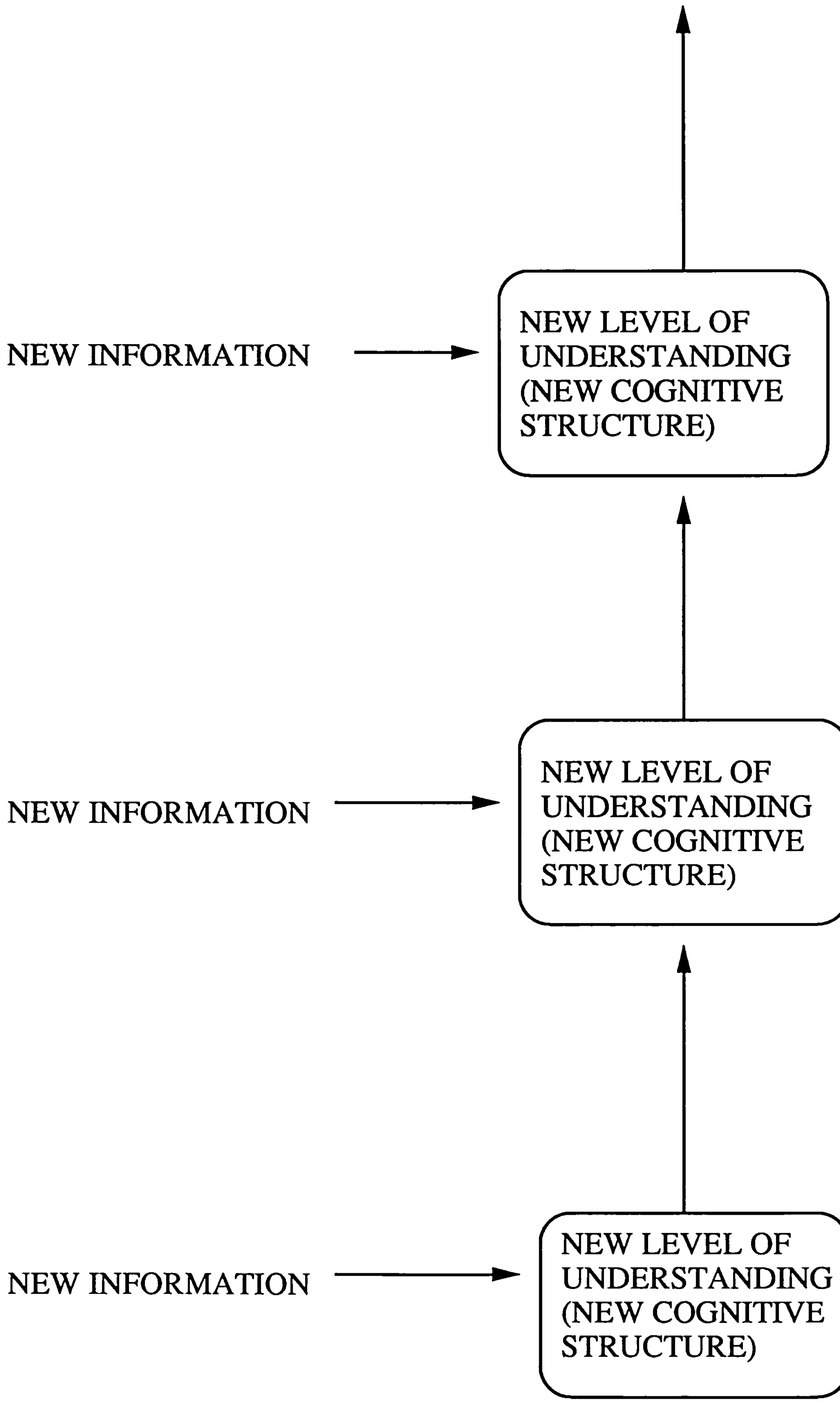

Figure 9.1
Development of Cognitive Structures

Each stage in the development process marks the completion of the preceding one and gives rise to the possibilities of the next stage. Thus, cognitive development is seen as a continuous process of the organisation and reorganisation of cognitive structures.

9.4 THE INCORPORATION OF COGNITIVE STRUCTURALIST THEORY INTO THE EDUD HIERARCHY OF UNDERSTANDING

In the EDUD hierarchy of understanding, five evolutionary stages through which cognitive structures pass are distinguished. Whilst it is not claimed that the evolution of the stages identified here is necessarily correct, it nevertheless is claimed that it approximates the human learning process sufficiently for practical utility and in terms that are amenable to computer implementation.

The five stages which the EDUD hierarchy of understanding refers to were identified and a detailed description of each of them was given in the discussion of the understanding process in Chapter 8.

9.5 CONCLUSION

The EDUD hierarchy of understanding is grounded in the cognitive structuralist theory of learning. In particular, it takes as given that the evolution of cognitive structures is indicative of an underlying process of learning/understanding. The sequencing of the developmental stages through which these structures pass is based on universal laws of understanding which give rise to this hierarchical formation. It is the emulation of this process of understanding which is encapsulated in the EDUD User Model.

The triggering mechanisms which bring about the evolutionary change in the cognitive structures are described by the application of pragmatic rules of inference. These rules are discussed in detail in Chapter 10.

The content of the cognitive structures is seen as the goal-state resulting from the application of the pragmatic rules of inference. This content is derived from scientific theory (Van80, Ach83) which provides categories of explanation.

Chapter 10

PRAGMATIC RULES OF INFERENCE

10.1 INTRODUCTION

In chapter 8 the importance of the definition and role of understanding for ITS design was discussed. In particular, emphasis was placed on the view of understanding as both a psychological and as an epistemological process.

In this chapter, the focus of attention is on the pragmatic rules of inference which initiate the psychological process of understanding and ensure that epistemic continuity is achieved.

The following issues are therefore addressed in this chapter:

i. definition of the pragmatic rules of inference
ii. the relationship between understanding and the pragmatic rules of inference
iii. the relationship between the pragmatic rules of inference and problem-solving
iv. the relationship between the pragmatic rules of inference and pedagogics and
v. the role of pragmatic rules of inference in the EDUD model.

10.2 PRAGMATIC RULES OF INFERENCE

The pragmatic rules of inference which are described in this work are 'rule systems' which are used by people in the reasoning process but are different from the formal rules of logic, commonly used to describe human-reasoning processes. Whilst logic has been used in the design of many knowledge-based systems, it is the view held here that people typically do not reason using the rules of formal logic. Although logical rules may be sufficient for the purposes of semantically tractable formalisation of knowledge and computational implementation, they are inadequate representations of the reality of human cognitive processes. Thus, whilst logic is intellectually appealing it is inadequate for 'real' purposes (in terms compatible enough to be constructive) (Wil87).

In everyday reasoning people use rule systems that are highly generalised and abstracted but nonetheless defined with respect to classes of goals and types of relationships (Nis87). These are the type of rules incorporated into the design of the EDUD model for the learning-teaching process. They are 'pragmatic' in that they resemble the more 'natural' type of logic reflecting the way people reason about everyday occurrences and whilst they are more specific than logical rules they are also 'abstract' in that they are not bound to any specific content domain, that is, they are domain-independent. Their function is thus not to model any domain-specific environment directly but to generate empirical rules, which are added to the individual's general knowledge base. It should be pointed out, however, that whilst these rules may be applied to a number of different domains, they do refer to certain types of problem goals and relations between events.

10.3 THE RELATIONSHIP BETWEEN UNDERSTANDING AND THE PRAGMATIC RULES OF INFERENCE

In the EDUD model a state of understanding is denoted by the existence of cognitive structures which represent statements about the content of individual knowledge states. The process of understanding, on the other hand, is the evolutionary combination of the external evidence (explanation) and internal operations that manipulate and reorganise the incoming information into these structures. However, before internal operations can be carried out on incoming information, this information must, in some way be elicited from the environment. This process of gathering information is initiated by the application of 'abstract pragmatic rules of inference' to the environment (see Figure 10.1).

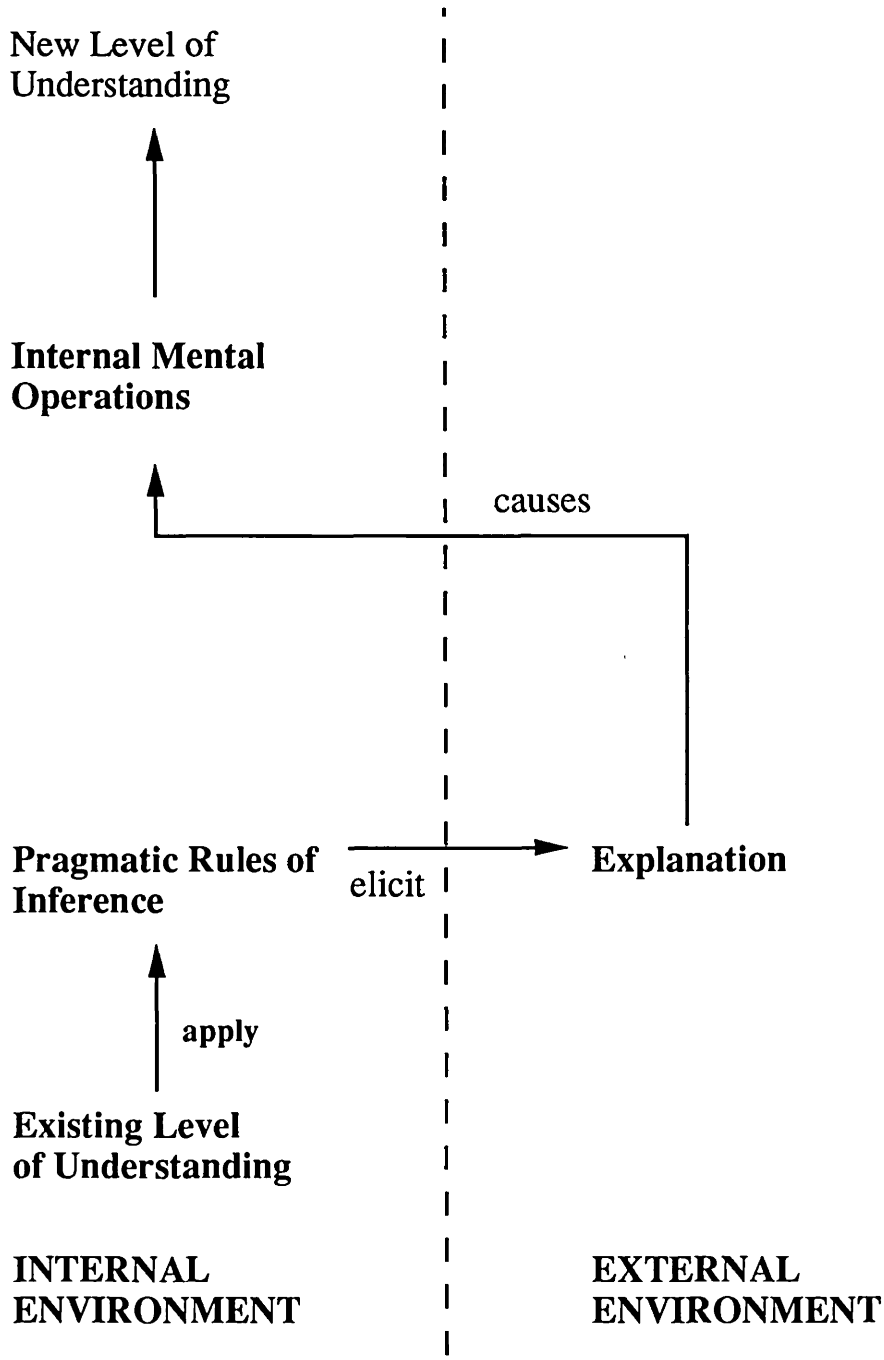

FIGURE 10.1
Schematic Representation of Relationship between Understanding and Pragmatic Rules of Inference

10.4 THE RELATIONSHIP BETWEEN PRAGMATIC RULES OF INFERENCE AND PROBLEM-SOLVING

The individual is not always a passive receiver of information, he frequently takes the initiative and actively interrogates the environment. It is held here that it is pragmatic rules of inference which serve as the psychological triggering mechanisms which initiate such interrogation. These rules constitute the propositional operations which the individual puts to his environment in order to elicit pertinent information and contextually correct explanations. If the right kind of questions are asked then meaningful information is obtained which may subsequently be used for problem-solving. Thus the reasoning process associated with learning the application of these rules takes place according to EDUD principles prior to the problem-solving process. Inference is seen as the process by which 'evidence' is gathered prior to weighing it (see Figure 10.2).

This view of abstract inferential rules of reasoning that are applied by the individual prior to problem solving differs from those views expressed in recent years by theorists concerned with problem-solving behaviour. Problem-solving theorists believe that there are no domain-independent inferential rules, but only highly domain-specific empirical rules dealing with concrete types of events. Newell stressing the role of problem-solving behaviour states that "learned problem-solving skills are, in general idiosyncratic to the task" (New80).

The view held in this research is that problem-solving skills are not necessarily specific to a task. Further, some theory of the methods employed by the user to focus and order the gathering of information and making of assertions about his environment prior to problem-solving is essential for Intelligent Tutoring System design.

In the design of an ITS, a theory of knowledge communication needs to be presented which will enable the acquisition of the knowledge represented by someone else. What is required therefore is not a computational representation of what people are capable of doing but a

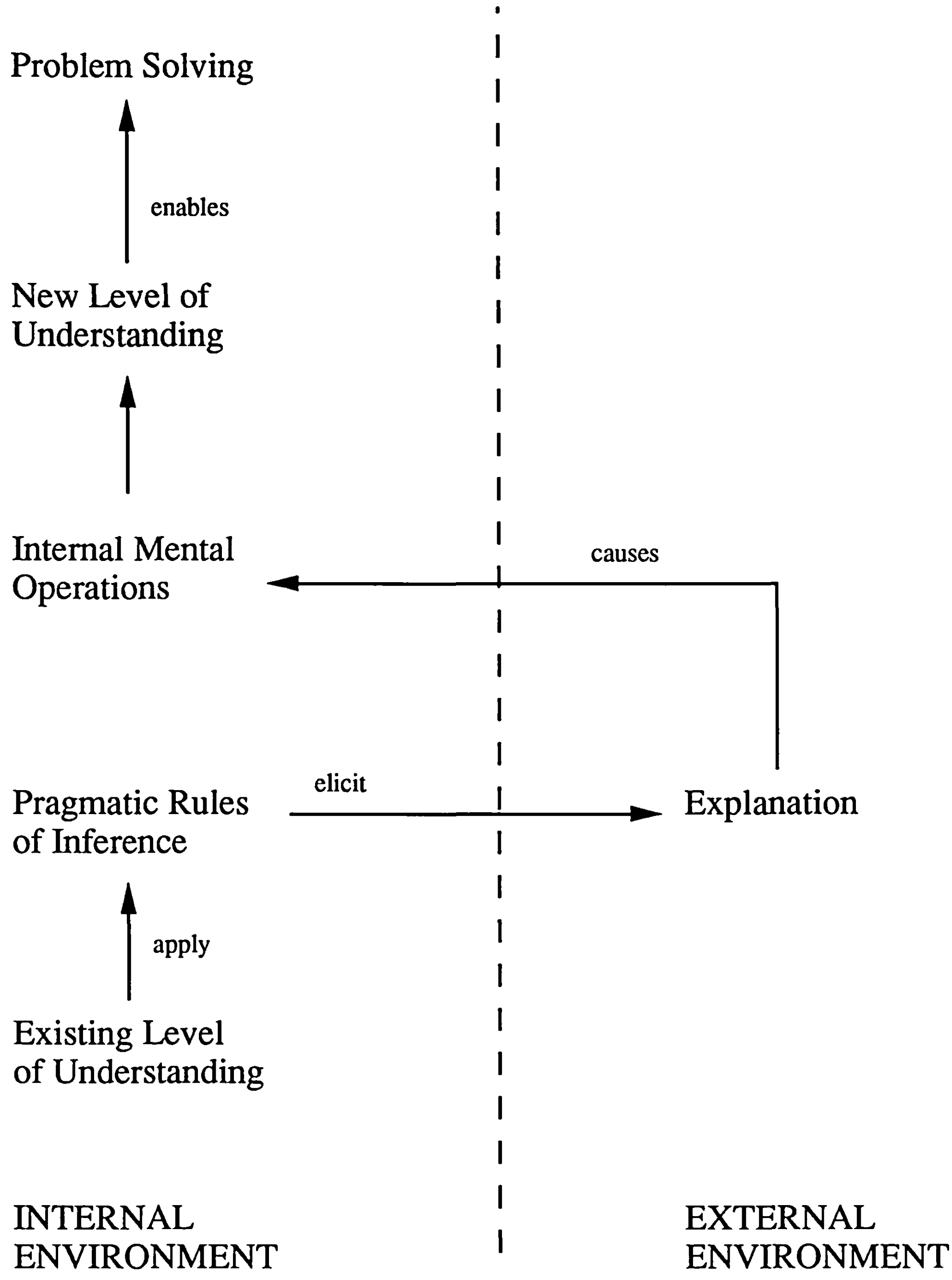

Figure 10.2
Schematic Representation of Relationship between
Pragmatic Rules of Inference and Problem-Solving

model of how reasoning is organised and activated. This model may then be used by the system to present the information and knowledge in such a way that its actions seem natural and self evident. Accordingly, an explicit and epistemically faithful representation of the inference procedure is crucial.

10.5 PRAGMATIC RULES OF INFERENCE AND PEDAGOGICS

Explicit representation of inferential procedures also has important implications for pedagogics. The purpose of teaching is to facilitate the student's traversal of a particular space of knowledge states. In order to provide adequate instruction, some knowledge of the general purpose reasoning mechanisms which enable the individual to progress from one state to another is required.

10.6 THE ROLE OF PRAGMATIC RULES OF INFERENCE IN THE EDUD MODEL

In the EDUD model the pragmatic rules of inference represent a set of psychologically plausible domain-independent rules which purport to emulate general reasoning processes. The temporal ordering of these rules reflects the path which the individual traverses in exploration of the knowledge state. Accordingly, if the individual knowledge states are considered as pieces of a curriculum, then the pragmatic rules of inference may be seen as the 'glue' which links these parts together.

The rules in question act as triggering mechanisms whose purpose is to initiate change, that is, a better understanding of the domain-knowledge, in the user's internal cognitive structures. Inferential competence (what has to be mentally computed) is linked to a specially designed typology of explanations. Inferential performance (the mental processes underlying these computations) is then characterised by the order in which these explanations are given. Each rule has an intended explanation eliciting function. For example, to obtain an identity explanation the appropriate rule is one which allows the user to 'define' what something is by establishing a valid set of its attributes.

10.6.1 The Content of the Pragmatic Rules of Inference in the EDUD Model

The first abstract inferential rule which is applied in the process of understanding is 'to discover the definitional attributes' of a particular concept in a specific domain together with their value. The propositional operators which are applied in this case, therefore, tend to take the form of questions such as 'what is it?', 'how much is it?'. The intention is to elicit an identity explanation, commonly associated with an 'it is' answer-giving type format.

Once the individual has some understanding of the basic meaning of the concept the next inferential rule to be applied is to 'discover the purpose or use of an object or action'. The explanation-seeking question in this case is 'what does it do?', 'what is its function?'. Explanations will provide information such as 'it exists for the purpose of....'

The next inferential rule to be applied, will then be to 'discover the causal relationships'. The propositions which will be put to the environment at this stage are terms concerning 'how....?','what causes x?', 'what effect does x have on....?' and explanation given by 'by......'.

In the next stage the individual attempts to 'discover why it is the case that....'. Explanations are induced by asking 'why...?' and given by using a 'because....' format.

The final type of pragmatic rule identified is to 'discover the various possibilities'. Propositions put to the environment are therefore of a 'what if...?' format and explanation takes the form of 'if.....then' (see Figure 10.3).

Pragmatic Rule Applied	Explanation Driven
What is it?	It is...
What does it do?	Its function is....
How does it do it?	By....
Why does it....?	Because ....
What if..?	If......then

Figure 10.3 Application of Pragmatic Rules of Inference and Explanations Elicited

10.7 CONCLUSION

In this chapter the need to incorporate a theory of reasoning into the design of Intelligent Tutoring Systems has been discussed. It has been stated that the design of computer systems whose objective is to emulate the human reasoning process in some way needs to emulate the pragmatic rules, representative of the type of rules used by humans in everyday reasoning processes.

In the EDUD model the pragmatic rules of inference enable the articulation of the learning process in such a way as to allow it to be understood in terms of general functions. These general functions in turn provide a descriptive vocabulary, the generality of this vocabulary is useful for eliciting explanation (Wen87). The result of this explicit representation of the inference procedure is that the EDUD model is functional in contrast with purely behavioral simulations where functions are only implicit in the behaviour simulated.

A major advantage to incorporating these rules in ITS design is that they are abstract and thus are not tied to a specific domain which makes them suitable for incorporation into a framework for an ITS shell.

Chapter 11

PEDAGOGIC ISSUES IN INTELLIGENT TUTORING SYSTEM DESIGN

11.1 INTRODUCTION

In designing the tutoring facilities for an ITS, it has been suggested (Ded86) that the following pertinent questions need to be asked:

1. when to intervene in the teaching interaction
2. what to discuss
3. which presentation strategy to use
4. how much to say

In this chapter these questions will be looked at in depth.

11.2 WHEN TO INTERVENE IN A TEACHING INTERACTION

The question of when to intervene in the teaching interaction and what effect such intervention has on the motivation of the user is an important consideration in tutoring system design. Consideration must be given both to how often the interaction should be interrupted and to who should take the initiative to interrupt, that is, the system or the user.

In deciding how often to intervene it must be kept in mind that too frequent intervention may lead to boredom or frustration on the part of the user. This in turn may, for example, prevent the development of the user's cognitive skill of detecting his own errors and accordingly any learning which may have resulted from his having made such errors is lost. On the other hand, too little intervention may lead to floundering.

In deciding whether the user should be allowed to explore the environment uninterrupted with system-intervention only occurring in response to requests for assistance or whether the system should take the initiative and intervene where it deems fit, it should be kept in mind that lack of system intervention may result in user fixation on one particular area. For example, it has been found that in a gaming situation lack of intervention may lead the user to fixate on a subset of the available moves and hence miss the potential richness of the game (Bur82).

11.3 WHAT TO DISCUSS AT INTERVENTION

Once the need for intervention has been established the question of what to discuss at such intervention needs to be determined. This decision determines which particular pieces of domain- knowledge are both relevant and memorable and takes into account the educational, epistemic and psychological needs of the user.

Education is concerned with identifying the content of the target knowledge and sequencing its presentation in accordance with a set of pre-specified educational objectives (Blo56). For example, in the work by Bloom a taxonomy is provided whereby educational objectives are classified and assessed on a cognitive continuum ranging from simple to complex intellectual skills.

The continuum begins with
1. the student's recall and recognition of KNOWLEDGE and
2. extends through his COMPREHENSION of such knowledge to
3. his SKILL in its application,
4. his ANALYSIS of situations involving this knowledge together with
5. his skill in its SYNTHESIS into new organisations and finally
6. his EVALUATION of that area of knowledge to judge the value of material and methods for given purposes.

Epistemology is concerned with selecting and sequencing the knowledge to be presented to the user in a way which will ensure 'epistemic continuity'.

Psychology is concerned with linking the content of the information to be presented with pre-specified learning hierarchies. For example, a pretest-post-test approach, based on a learning hierarchy is provided by Gagne (Gag62). In this approach a desired task which is the goal of the training is identified and the question is then asked as to what kind of capability an individual would have to possess if he were able to perform this task successfully. The content of the educational interaction is then designed to reflect this capability.

11.4 WHICH PRESENTATION STRATEGY TO USE

The teaching strategy adopted by the tutoring system is crucial to its successful functioning. To date, there are a number of teaching strategies which have been implemented in IT systems which will be discussed below. However, before discussing these strategies, mention must be made of the various forms of presentation that such strategies may assume and which are also of importance in the successful functioning of the system. For example, explanation may be manifest verbally, numerically, physically, conceptually or symbolically (Ach83). The question for ITS design is whether any one form is better suited for a specific task or individual user than another. For example, it has been found that instructional strategies with "visibility", that is, where programming processes are demonstrated step by step via a simulated, simplified machine, assist the teaching of novices in programming languages (duB81).

A form of presentation which is increasingly being utilised as a medium for explanation in ITS is that of graphics. An outstanding example of this type of presentation strategy is found in the STEAMER system (Hol84). This system is designed to train engineers to perform a collection of procedures associated with operating steam propulsion plants by means of an inspectable simulation based on computer graphics.

A further important consideration in providing explanation facilities, and one which is frequently ignored, is that of rhetoric. Explanation should be stated so that it will be understandable. Use of language in explanation should be neither too complex nor too simple.

There are a number of approaches to the design of the teaching strategy in current intelligent tutoring systems. Three of the most important of these strategies, namely, Socratic Tutoring, Discovery Learning and Simulation are discussed below. A number of others are identified in Table 11.1 (Kea87).

11.4.1 Socratic Teaching Strategies

In the Socratic teaching method, the tutor does not teach the user by directly exposing him to the teaching material but provides successive questions intended to lead the user to reveal his current knowledge level. By examining the validity of the conclusions he draws and identifying any paradoxical contradictions which emerge the user himself draws conclusions on the information which he already has and accordingly is able to formulate general principles based on particular cases. Inherent in this type of tutor, is provision for a mixed- initiative dialogue whereby either the user or the system may instigate an interaction.

Whilst on the surface this method of tutoring appears promising, true Socratic tutoring requires the ability to learn in conjunction with the student, and such powerful learning models are not yet available (Wen87).

11.4.2. Discovery Learning

In recent years researchers have focussed on learning environments intended to facilitate learning-by-doing. By providing unstructured exploration users are able to formulate and test their own ideas and learn from the consequences of their behaviour. The LOGO programming language is based on the view that mental models are developed through intellectual exploration in which strategies are developed for purposeful inquiry. The more one learns, the better the model of learning and the more able one

System	Subject Matter	Student Model	Tutor Model	Reference
SCHOLAR	Geography	Overlay with importance weights	Socratic dialogue management	(Car70)
WHY	Causes of rainfall	Misconception identifier	Socratic dialogue	(Ste82)
EXCHECK	Logic & set theory	Overlay	Reactive environment with adviser	(Sup81)
BIP	Programming in BASIC	Overlay	Reactive environment with curriculum net and adviser	(Bar76)
SPADE	Programming in LOGO	Overlay	Reactive environment with coaching	(Mil82)
GUIDON	Infectious diseases	Overlay	Reactive environment with structural interactions	(Cla83)
ALGEBRA	Applied algebra	Overlay	Reactive environment with coaching	(Lan83)
PROUST	Programming in Pascal	Misconception identifier	Reactive environment with adviser(no tutoring function)	(Joh84)

Table 11.1 Some Intelligent Instructional Systems (adapted from Kearsley, 1987)

becomes as a learner. In this learning situation the selection of a particular aspect of the world to explore is not considered to be important. Learning is seen as a process of familiarisation with some subject material, discovering its problems, and resolving them by proposing and testing simple hypotheses in which new problems resemble others already understood (Pap80). A shortcoming of this approach is the inefficient use of time as well as the fact that the user may get lost and the system may be unable to offer the necessary support.

11.4.3 Simulation

Simulation is a powerful method of teaching. It provides a model of expert problem solving which the student can access in a realistic context. It allows the learner to accumulate experience in a typical real world environment, thus preventing him from limiting his proficiency to problem sets based on isolated topics. The difference between a simulation and discovery environment is that the learning-by-doing environment is designed for developing process skills whilst in the simulation environment the tutor offers a more instructor-centered approach geared to providing a foundation of descriptive knowledge. The situation is thus more 'structured' than in discovery learning environments.

A criticism of these types of systems is that unless they specifically incorporate a teaching system that can reason about procedural tasks, the procedural knowledge which incorporates the reasoning used by the system to solve problems in the domain is typically missing. Although, to date, no general solutions to this question, based on adequately detailed pedagogical theory exist (Bre88) the STEAMER system does attempt to explain the qualitative processes behind the steam engine simulation (Woo88).

11.5 EXTENT OF INTERVENTION

In determining how much to say consideration must be given to the amount of knowledge that will be sufficient to increase a user's understanding of a particular topic, dependent upon his purpose and epistemic situation.

11.6 CONCLUSION

This chapter has examined some of the pedagogic issues relevant to Intelligent Tutoring System design. Examples of a number of approaches in existing tutoring system design have been given. In particular stress has been placed on when to intervene in the teaching interaction, what to discuss, which presentation strategy to use and how much to say.

Chapter 12

THE USER MODEL

12.1 INTRODUCTION

The rationale for including a User Model within Intelligent Tutoring System design is that no effective communication of knowledge can take place unless the communicator of information has some awareness of the information needs of the recipient. Accordingly, the function of the User Model is to provide a dynamic assessment of a user's current level of functioning, including aspects such as his history, capabilities, knowledge, goals and motivation (duB87).

Early CAI programs adapted to individual learning situations by using scalar models of the pupil. However, recent programs are more ambitious and varied methods of modelling have been used.

Of the four architectural modules common to ITS design described in Chapter 3, the User Model has received the most attention. (Although the desirability of this state of affairs is not universally accepted - some researchers believe that emphasis should be shifted from the representation of the user to improving teaching tactics - the rationale being that whilst a system may have a rich representation of the knowledge and skills it wishes to impart it will be impaired by impoverished teaching tactics) (duB87).

Focussing on the User Model results from the view that intelligent tutoring systems should be adaptive. It is through the system's ability to adapt to the individual user that its apparent 'intelligence' is demonstrated.

The User Model therefore functions to provide the system with a representation of the user's current state of cognitive functioning.On the basis of this assessment decisions are made as to what subsequent interaction would be appropriate, thus enabling the system to adapt to the individual user.

12.2 DIFFERENCE BETWEEN MENTAL MODELS AND EDUD USER MODEL

Reference is frequently made in current literature to the terms Mental Models and User/Student Models. In order to avoid confusion in the way in which these terms are used in this thesis, the following distinction is made between these two concepts.

Mental models are concerned with describing which information about a particular domain, the individual selects and represents internally. The priority is thus WHAT is selected and represented rather than HOW this is done. These models therefore refer to the individual's beliefs and theories rather than to any universal properties of cognition (Pay88). Accordingly, emphasis in the development of such models is placed on the analysis of specific content domains.

The EDUD User Model described here,on the other hand, is a cognitive model of how the user obtains and uses domain-independent knowledge. It is a simulation model of the human reasoning process and consists of both a general model and an inference procedure. Its function is therefore not just to provide a static description of the user's internal representations at a particular moment in time, but it includes a 'program', an inference procedure by which the user's process of understanding is defined and assessed. Accordingly, the primary characteristic of the EDUD User Model is that it not only explains the basis of the user's current behaviour but may be used to predict subsequent behaviour. It is thus, an executable model (Cla88).

12.3 EXISTING APPROACHES TO USER MODELS

The accuracy and effectiveness of existing User Models together with the underlying theories and methods for their construction varies from program to program.

Rich gives three dimensions for classifying User Models (Ric83):

a. A model of a single stereotype-user versus a collection of models of individual users.

b. Models specified by the user of the system versus models inferred by the system based on the user's behaviour

c. Models of long-term user characteristics versus models of the current task.

An alternative classification for User Models may be in terms of their capabilities (see Table 12.1).

12.4 A GENERIC APPROACH TO USER MODEL DESIGN

It is held here that the design of the User Model should be based on generic principles. The argument for including generic principles in system design has been discussed in Chapter 5. Accordingly, the design of the User Model should not be one in which a particular task is examined, decomposed in terms of the underlying psychological processes inferred to be necessary for its solution and the incorporation of a measure of the extent to which such processes have been achieved. Rather, the inverse of such a procedure should be followed. The universal psychological processes taking place in the development of understanding should be examined first and the task decomposed in terms of these principles. Such an approach is essential if User Models are to be capable, for example, of recognising and evaluating alternative models that exist within the user. This is in sharp contrast with specially-engineered programs.

The designs of the majority of existing User Models have not been based on domain-independent characteristics. The most notable exception to this rule and an interesting attempt to create a domain-independent User Model has been the User Modelling Front End Subsystem (UMFE) (Sle85). This system was discussed fully in Chapter 5.

MODELLING CAPABILITIES	EXAMPLES
Assess correctness of final response. For example in terms of a single number to represent the overall complexity of tasks to be performed (scalar models)	RULETUTOR
Judge consistency user's responses in relation to data presented incrementally (profile models)	SOPHIE
Articulate the 'issues' that arise out of the user's response (ad hoc models)	WEST
Analyse user's responses in relation to an expert's choices for same circumstance (differential models)	WEST BUGGY
Analyse user's responses as a subset of expert's knowledge (overlay models)	WUSOR
Perform inferential diagnosis on student's responses (diagnostic models)	LMS REPAIR

Table 12.1 Capabilities of User Models

The design of the User Model in EDUD attempts to overcome the shortcoming of UMFE of having to specify difficulty levels for each concept. Instead the EDUD User Model is constructed by examining levels of understanding, where universal principles may be identified, rather than levels of difficulty.

12.5 THE EDUD USER MODEL

An effective and accurate approach to the assessment of cognitive development both in terms of problem solving and learning processes on which to base the User Model is a diagnostic inferential one which has its foundation in a classical cognitive structural approach (Gar78),(Fla63). This approach which is described in Chapter 9 provides a framework which reflects the evolutionary nature of the learning process and describes cognitive development as a progression from less complex to more advanced states of cognitive functioning and understanding. Accordingly, the model is not concerned with representing the relative difficulty of a specific concept but attempts to represent the continual evolution of the user's understanding of it. The advantage of using such an approach is that it is possible to pinpoint the conceptual relations which are understood at a specific phase of development and consequently predict implicitly those relations which still need to be tutored and those which it is known must have already been mastered in order to achieve the present level of understanding. In this way the User Model serves the purpose of providing a frame of reference for monitoring the learning process as well as establishing a reference point from which to drive the explanation facilities. The level and content of the explanation material is therefore closely related to the User Model and the two may be seen as inseparable parts of the EDUD model.

Each step in the reasoning and learning process can therefore be identified and made explicit through the explanation facilities in the program. Further, because development is seen as passing through identifiable stages, such User Model accurately reflects the particular stage at which the user is functioning. Consequent explanations are then aimed at moving him from this level of functioning to the next. The objective is to ensure that the user has adequate knowledge and understanding of the principles and domain-processes underlying the problem-solving process. The focus is thus on demonstrating the responsiveness of the model to the user's level of understanding, that is, the form of understanding or knowledge possessed by the user at a given cognitive stage.

12.6 THE EDUD HIERARCHY OF UNDERSTANDING AND THE USER MODEL

The EDUD User Model is concerned with representing the user's level of understanding through the identification of the content of his cognitive structures. This representation then identifies the stage at which the user is functioning in terms of the development of the structure which is being formed. On the basis of this content the type of explanation necessary to move him from this level of cognitive functioning to the next may be identified. Once the contents of a particular cognitive structure have been identified, an attempt is made, through the medium of explanation, to develop and extend his understanding.

In Chapter 7 the stages of understanding through which the individual passes have been defined. These phases are:

1. The Stage of Figurative Knowing
2. The Functional Stage
3. The Cause/Effect Stage
4. The Complex Derivational Stage
5. The Stage of Hypothetico-Deductive Reasoning

The User Model functions by pinpointing the conceptual relations which are understood at each level in the hierarchy. This is achieved by the Tutoring Module initiating diagnostic questions to assess the user's present level of understanding. Once the user's current level of understanding has been determined, a decision is made as to what material should be explained to him next. This decision is based on the belief that the process of understanding is loosely sequential. Thus if at any level in the hierarchy the user is functioning successfully all previous levels may be assumed to have been successfully completed. Consequently, there is no need to provide further instruction at these levels. Conversely, if at any stage in the hierarchy the user is not functioning successfully it may be assumed that he will not have reached higher levels of understanding. Further interrogation is therefore required to discover at which level in the hierarchy he is functioning and the appropriate instruction offered accordingly. The levels in the hierarchy thus form the basic framework for monitoring the progress of the user's understanding.

This hierarchical representation of the user's level of understanding has important implications for system design. It allows direct comparison of the user's input with the steps in the hierarchical structure. In this way, detailed inference models of the processes taking place within the user are obviated, as are any cumbersome computation overheads that might be associated with them.

12.7 CONCLUSION

In this chapter the importance of the role of the User Model as the instrument which effects tutoring system adaptability has been discussed. In particular, it has been stressed, that any representation which purports to emulate human understanding/learning processes, must be initiated from a cognitive perspective and based on generic principles.

The view taken in this chapter has been that in order to be an effective tutor the system needs to adapt its performance to the user's developmental level at a given cognitive stage. In order to achieve this a profile of the user's current understanding/learning level needs to be constructed. This profile reflects the user's initial knowledge of the domain fragment of an application domain intended for teaching as well as his progress in a learning task brought about by his interaction with the system.

In the EDUD User Model the user's understanding of the material presented to him is assessed at each level in the understanding-hierarchy. Because the process of understanding is viewed as relatively sequential, if at any level in the hierarchy the user is functioning successfully all previous levels may be assumed to have been successfully completed. The levels in the hierarchy thus form the basic framework for a shell for monitoring the progress of the user's understanding.

Chapter 13

THE EDUD FRAMEWORK FOR KNOWLEDGE DECOMPOSITION

13.1 INTRODUCTION

A problem for both expert and Intelligent Tutoring System design is to provide efficient and appropriate methods for the sub- division of the body of target knowledge into topic-dependent autonomous, information-containing structures. A requirement of the structures is that they be amenable to representation, computation and human comprehension. However, the objectives of ITS in relation to knowledge decomposition are different to those of expert systems with the result that knowledge-decomposition techniques suitable for expert system design are not necessarily suitable for the design of Intelligent Tutoring Systems. The basis for this difference is that expert system design seeks to construct a knowledge-base capable of supporting the problem-solving methods of a human expert. In an ITS on the other hand, the system design is more concerned with supporting learning processes taking place in the non-expert user.

The objectives in this chapter are:

(i) to consider current artificial intelligence, intelligent tutoring, psychological and pedagogic approaches to knowledge decomposition and representation,

(ii) to describe a generic framework for the decomposition of knowledge (however, it must be noted that whilst the framework may be applied to a number of different domains, it does refer to certain types of problem goals and relations between events),

(iii) to illustrate the relative domain-independence of the framework for knowledge decomposition by providing examples from two disparate domains, namely, car mechanics and medicine, and

(iv) to demonstrate the operationality of the knowledge by sequencing it for incorporation into a User Model

13.2 DECOMPOSITION OF KNOWLEDGE FOR EXPERT SYSTEMS

Artificial intelligence has provided a number of domain- independent techniques for the decomposition and representation of a body of knowledge. Amongst the most widely recognised representation techniques are semantic networks, frames, discrimination trees and production rules. The major focus for the decomposition task within AI has been on representing and organising the target-knowledge for use with expert systems and has been concerned primarily with the separation of static information from that of dynamic information. Methods for the representation of static information have focussed on the classification and/or indexing of objects, and are concerned with describing what something is. In the case of dynamic information, attention has been paid to methods for representing knowledge about problem-solving and is concerned with describing what to do.

13.3 THE PEDAGOGIC APPROACH TO KNOWLEDGE DECOMPOSITION

Pedagogics focuses on the decomposition of the target knowledge for the purpose of providing an explicit curriculum. Decomposition aims to ensure that the 'chunks' of information constructed form a coherent whole and are of a size which is easily assimilable by the student. Units of information need to be small enough not to exceed the limits of the individual's static memory and coherent enough so that individual chunks may be incorporated into a meaningful whole within a single lesson.

In order to achieve this goal, conventional instruction provides an explicit curriculum. However, conventional instruction does not provide a detailed representation of the target knowledge. Intelligent instructional systems, on the other hand have tended to provide a detailed representation of the target knowledge without explicitly representing the curriculum (Les88).

In particular what is missing from both conventional and 'intelligent' instruction is the knowledge that represents the 'glue' connecting the parts of the content of any specific lesson (Les88).

In this thesis a contribution is made to solving this problem by describing the EDUD knowledge decomposition framework which provides a coherent structure for the decomposition of the target knowledge for teaching purposes by dividing its content into assimilable 'chunks' on the basis of the categories of explanation referred to in Chapter 7. The 'glue' that holds these chunks together and ensures coherence between instructional units is provided by the application of the pragmatic rules of inference described in Chapter 10.

13.4 COGNITIVE PSYCHOLOGY AND KNOWLEDGE DECOMPOSITION

For methodological reasons cognitive psychology has, for the most part, avoided dealing directly with knowledge and its use. This state of affairs may be seen as a result of the nature of psychological research which requires testable hypotheses and reproducible results. Accordingly, psychologists have been led to devise theories and experiments independently of 'what' the subjects in the experiments know. Most cognitive experiments have focussed on disclosing basic properties of human cognition, such as short-term memory and abstract abilities, independent of the knowledge involved in such activities.

However,the emergence of AI has posed interesting questions for cognitive psychology. These include how the human decomposes, stores, indexes and retrieves information and how the 'right' chunk of knowledge is found at the right time without irrelevant information getting in the way.

13.5 INTELLIGENT TUTORING SYSTEMS AND KNOWLEDGE DECOMPOSITION

In the introduction to this chapter it was stated that the methods of knowledge representation and decomposition used in the design of expert systems are not always suitable for the design of Intelligent Tutoring Systems. This inappropriateness stems from the fact that where the purpose of a system is to provide problem-solving capabilities as is the case in expert systems, emphasis is placed on performance and efficient system functioning and accordingly the knowledge in such systems is frequently represented in a 'compiled form'. What is meant here by 'compiled' refers to knowledge which has become so specialised to a specific use as to have lost transparency and generality (Wen87). Examples of this type of knowledge are ubiquitous in expert systems, particularly, in those instances where the knowledge base contains the relationships between symptoms and malfunction hypotheses in some form (Sem86). This is not to say that there are not instances in expert system design where it is necessary to 'articulate' the underlying knowledge. For example, such articulation may be required for the provision of explanation facilities. However, even in these instances, the purpose of articulation is to justify the system's problem-solving conclusions.

In the design of an ITS on the other hand, the goal is to support the communication of pertinent knowledge in such a way as to facilitate the (user's) learning through the teaching process. This requires that the target-knowledge be 'articulated'. By articulated is meant that the compiled knowledge is augmented with additional knowledge in the form of deeper beliefs or models of first principles, to which the compiled beliefs can be traced. The problem of representation is further complicated in the design of ITS's because it remains controversial as to which method of communication of knowledge is best for teaching purposes. There are a number of views which are adhered to in this regard, with the dominant view held that teaching of problem-solving skills should be the vehicle for communicating knowledge about a given domain. An alternative view and the one taken in this research holds that it is more rewarding (to the user) to emphasise understanding of the target-knowledge and therefore to teach the knowledge underlying problem-solving skills. Whilst in the former approach most of the knowledge engineering is concerned with the representation and monitoring of the problem-solving activities of the user, in the latter approach it is the properties of the body of knowledge itself which are focussed on. The objective of increasing the user's level of 'understanding' of a particular body of knowledge requires a deeper level representation than that required to teach problem-solving skills.

13.6 THE EDUD APPROACH TO KNOWLEDGE DECOMPOSITION

The EDUD approach is concerned with supporting, through an appropriately devised method of knowledge-representation, decomposition and sequencing, the learning processes taking place in the user, namely the development of new cognitive structures. In order to achieve this objective the EDUD knowledge-decomposition hierarchy is provided. This structure consists of hierarchically arranged levels, each of which is associated with an abstract node name. The content of the levels in the hierarchy represent the content of the pertinent cognitive structures. The abstract node names provide a method for accessing the content of these structures. Thus the framework reflects an assumed, but plausible, cognitive method for the decomposition, indexing and retrieval of knowledge which is amenable to computer implementation.

The composition of the framework is based on generic principles. This emphasis on providing a relatively domain-independent framework makes a contribution to the pragmatic need for tutoring systems to move away from individually hand-crafted applications to the use of general purpose tools.

13.7 THE EDUD FRAMEWORK FOR KNOWLEDGE DECOMPOSITION

The content of each node in the EDUD framework is based on the five categories of explanation described in Chapter 7. These are:

1. identity explanations (which are concerned with the classification of objects and fundamental facts);
2. functional explanations which are concerned with the intended functions of objects in the domain;
3. causal explanations which are concerned with causal links and
4. complex-derivational explanations which are concerned with the application of laws to explaining behaviour of domain entities,
5. hypothetico-deductive explanations which are concerned with how the potential may be derived from the actual.

Support for the decomposition of knowledge into these categories is found in the work of Sembugamoorthy (Sem86) and Rasmussen (Ras85) who have identified similar categories.

The appropriateness of these categories for the decomposition of a target body of knowledge is demonstrated in the schematic representations of knowledge decomposition according to these categories for the fuel system of a motor car. In addition, an anaesthetic procedure for controlling regulation of consciousness is decomposed according to the first four categories of explanation. The decompositions are shown in Figures 13.1 & 13.2 respectively.

However, knowledge which is to be used either for human or computational purposes must be represented and stored in a way which facilitates its subsequent access and retrieval. The question remains, therefore, as to how these categories of explanation should be indexed.

13.8 INDEXING OF INFORMATION IN THE EDUD FRAMEWORK

It is crucial if an ITS is to function effectively that it is provided with a fast and efficient method of indexing information. In order to provide such a method, the EDUD approach again emulates human cognitive functioning by the incorporation of cognitive principles in the design of its indexing and retrieval mechanisms.

In order for the individual to understand something, he must be able either to elicit such information from the environment or to find the

IDENTITY	FUNCTION	CAUSAL	DERIVATIONAL
What is regulation of consciousness?	What is its function?	How is this function achieved?	Why regulate?
An anaesthetic procedure	1. Control patient alertness	i. decrease: inhalation hypnotic agents parenteral hypnotic agents ii. restore: spontaneous recovery narcotic reversal agents	I To control patient awareness: because need to control patient awareness has been recognised and decision made to implement treatment to produce change to assure return to normal state
	2. Control patient memory	i. decrease: inhalation hypnotic agents parenteral hypnotic agents ii. restore: spontaneous recovery narcotic reversal agents	
	3. Control sensory input to patient	i. control noise in patient area ii. control lighting in patient areas iii. control surgical manipulation of patient iv. control passive movement of patient	
	4. Control sensitivity of sensory nerve endings	i. decrease: infiltration with local anaesthetic agents administration of analgesics	II To control perception of sensation: because need to control perception has been recognised and decision made to implement treatment to produce change to assure return to normal state
	5. Control conduction of action potentials in afferent nerve fibres	i. decrease: inhalation hypnotic agents parenteral hypnotic agents local anaesthetic nerve block	
	6. Control perception of sensory input	i. decrease: inhalation hypnotic agents parenteral hypnotic agents flooded sensory input suggestion ii. restore: spontaneous recovery narcotic reversal agents	
	7. Control reaction to sensory input	i. decrease: produce mild euphoria suggestion	
	8. Control voluntary muscle tone	i. decrease: inhalation hypnotic agents parenteral hypnotic agents ii. restore: spontaneous recovery	III To control muscle tone: because need to control muscle tone has been recognised and decision made to implement treatment to produce change to assure return to normal state
	9. Control reflex muscle tone	i. decrease: inhalation hypnotic agents parenteral hypnotic agents infiltration with local anaesthetics agents ii. restore: spontaneous recovery	
	10. Control background muscle tone	i. decrease: inhalation hypnotic agents parenteral hypnotic agents ii. restore: spontaneous recovery	

Figure 13.1
DECOMPOSITION OF KNOWLEDGE FOR REGULATION OF CONSCIOUSNESS (ADAPTED FROM JACK (1976))

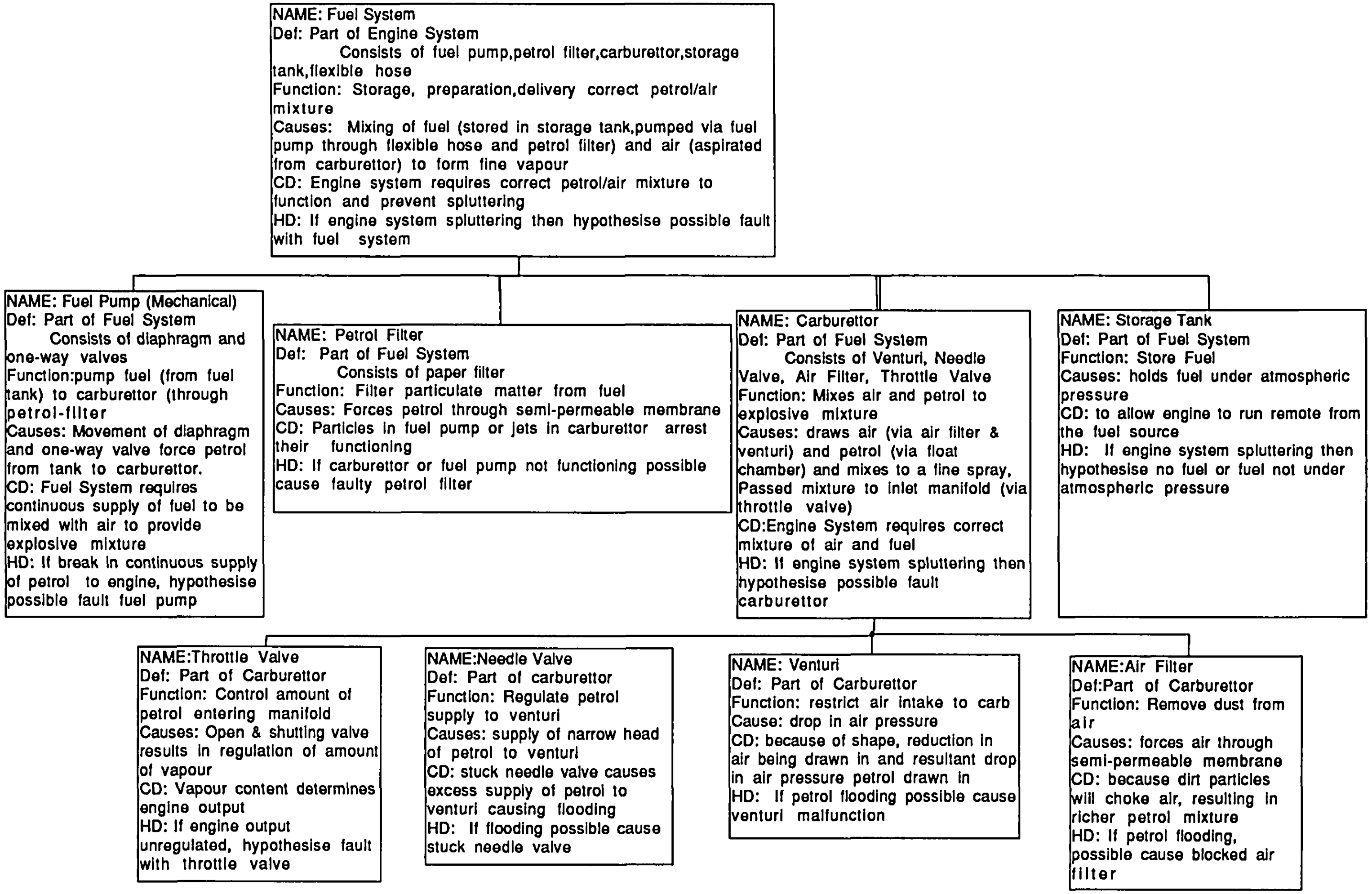

Figure 13.2
Partial Representation of Knowledge about a car fuel system

applicable cognitive structure in memory that describes it. According to the EDUD approach it is the pragmatic rules of inference which serve as indices and provide the retrieval cues, for the relevant cognitive structures. Each pragmatic rule of inference provides a set of primitives, which are then used as indices to interrogate the structures containing domain-specific fragments of target knowledge.

It is this method of triggering and retrieval of information by the application of the pragmatic rules of inference which is emulated in the EDUD model. An important aspect of this approach is that whilst the fragments of knowledge contained in the structures are domain-specific, the pragmatic rules of inference are abstract. A major advantage of this method of indexing information is that whilst the stored memory is richer than the set of indices pointing to it, by representing the knowledge at an abstract level the complexity of the representation is limited.

In seeking to retrieve information the following pragmatic rules of inference are therefore applied:
What is X?
What does X do?
How does X achieve this function?
Why does X achieve this function?
If X is altered.......Then.......

13.9 EFFECTING OPERATIONALITY OF KNOWLEDGE THROUGH REPRESENTATION

It is believed that the learning process is a relatively sequential evolutionary progression from less to more advanced understanding through a number of hierarchical stages. A more detailed description of this progression is found in Chapter 9. The stages of understanding identified are:

1. The stage of figurative knowing
2. The functional stage
3. The cause/effect stage

4. The complex-derivational stage
5. The stage of hypothetico-deductive reasoning

In learning each of the stages identified is traversed sequentially 'bottom up', that is, understanding proceeds from the stage of figurative knowing to the stage of hypothetico-deductive reasoning. These stages provide the foundation on which the representation of knowledge in the EDUD hierarchically-ordered frame structure is based. Each node in the structure has associated with it an information-seeking question in the form of a pragmatic rule of inference which in turn elicits and is associated with an application explanation (see Table 13.1).

PRAGMATIC RULE OF INFERENCE	EXPLANATION GIVEN
What is it?	Identity explanation
What does it do?	Functional explanation
How does it achieve its function?	Cause/Effect explanation
Why does it do this?	Complex-Derivational explanation
If X what then?	Hypothetico-Deductive explanation

Table 13.1
Pragmatic Rules of Inference associated with Explanation Categories

13.9.1 Operationality of Knowledge for Teaching Purposes

For teaching purposes, the knowledge in the EDUD framework is sequenced 'bottom up' in order to reflect the direction and development of the user's understanding. The information is therefore arranged and presented in a sequence which enables the user to understand how function results from definition and function, in turn, is achieved through behaviour. Further, it contains explicit pointers to generic domain knowledge and assumptions about behavioral alternatives. Thus each piece of information given to the user builds upon his understanding of the previous fragment which has been presented to him until the entire structure has been delivered (see Figure 13.4). The objective being that the user's understanding of the domain, will in its final form, reflect an integration of the knowledge obtained from this representation.

Definition: Fuel-pump
Is: part-of fuel system
Consists of: diaphragm & one way valve
Does : Deliver Fuel
By: Behaviour 1
Because: Fuel required by engine system
If:Fuel not supplied to engine system then
Symptom 1

Behaviour 1 Movement of diaphragm & one way valve
Symptom 1 Spluttering and choking

Figure 13.4
Representation and sequencing of information about a fuel pump for the purpose of increasing understanding of the domain

13.10 CONCLUSION

The representation, decomposition and presentation of knowledge in the design of knowledge-communication systems needs to reflect the general and particular teaching-learning needs of the individual as well as capture the peculiarities of the domain knowledge being taught.

In this chapter knowledge has been decomposed into a hierarchical structure according to EDUD principles and made operational through the interpretation of the hierarchy. The five evolutionary stages through which the individual progresses in seeking to understand a particular knowledge-domain have been used to provide a method for the sub-division of the body of target knowledge into autonomous topic-dependent information structures. Each stage in the process of understanding, has been linked to the previously defined explanation categories providing a set of structures amenable to both computation and human comprehension. The sequencing of the presentation of these categories of explanation is arranged so as to increase the user's understanding of the domain by its incorporation in the User Model.

An example of a body of target-knowledge about a car's fuel system is decomposed into a single framework to demonstrate how the purposes of explanation facilities for and instructional requirements may be met by the EDUD approach. In particular, an illustration of how sequencing of the target knowledge effects operationality was discussed together with the implications for the design of a dynamic User Model and as explanation facilities.

Chapter 14

THE EDUD TUTORING MODULE

14.1 INTRODUCTION

This chapter describes how the categories of explanation described in Chapter 7 together with salient principles of psychology (Chapter 8) and pedagogics (Chapter 11) may be unified in the design of the Tutoring Module for an Intelligent Tutoring System.

14.2 THE EDUD TUTORING MODULE

The EDUD Tutoring Module is designed with the ability to be used in three distinct phases, each of which supports different axioms for intervention. This design of the Tutoring Module reflects the principles of understanding outlined in Chapter 8. These stages include: the information-giving stage, the problem-solving stage and the observation stage.

The EDUD criteria for meeting the questions of how, when and what should be explained in the tutoring interaction outlined in Chapter 11 are, therefore, discussed in terms of these three separate stages.

14.2.1 EDUD criteria for what to say

In the EDUD model, the decision as to when intervention is appropriate is based on psychological and pedagogic considerations in the three stages as follows:

Stage 1 - The Information-Giving Stage

In the first phase of system functioning the objective is to teach the user the necessary declarative knowledge of the target-domain to provide him with a conceptual framework for subsequent problem-solving. It is believed that the individual cannot perform complex problem-solving operations unless both concepts and operations involved in the problem-solving task are built on a clear understanding of their meaning, capabilities and relationship with other concepts. A lack of declarative knowledge can impede effective learning and prevent an understanding of interrelationships linking different topics with an overall subject domain. Without fundamental understanding, skills are merely memorised. Consequently in this phase of system-functioning, intervention is system-determined and occurs at any point where the user indicates that he does not know the answer to a system initiated question (see Figure 14.1).

Stage 2 - The Problem-Solving Stage

In the second phase the objective is to assist the user in converting and reorganising his declarative knowledge in the context of procedures. Because the user now has the appropriate knowledge structure with which to explore the knowledge-network, procedural skills can be applied with full awareness of their causal meaning and a sense of their underlying theoretical justification. The user therefore is set problems to solve which are structured to encourage his own exploration of what he knows and doesn't know. Explanation is offered consequent to the user providing an incorrect answer to a question, and is offered with reference to the User Model and particular teaching strategy which pertains.

Stage 3 - The Observation Stage

In the third phase the objective is to provide the user with a model of heuristic problem-solving and decision making skills. Consequently an articulate expert system is provided. Intervention in this stage is user-determined. The view held is that the user has gained sufficient experience of the domain by this stage so as to be capable of judging when he requires further information. Accordingly, intervention occurs only at the user's request, explanations being offered in response to 'why?' type questions.

STAGE	ASSESSING UNDERSTANDING	EXPLANATION
Figurative Knowing	Do you know what a starter motor is?	It is an integral part of the engine - an electromechanical device
Functional Understanding	Do you know what it does?	Its function is to provide rotation to the crankshaft
Cause/ Effect	Do you know how it functions?	(By) turning ignition key to 'start' current flows to solenoid. Solenoid causes the drive of starter-motor to engage the ring-gear of the fly wheel and switches current to starter motor 'on' causing it to rotate. This initiates rotation of crankshaft which in turn starts engine
Complex-Derivational	Do you know why it is the case that we check the starter motor when an engine fails to start?	An engine requires rotation of the crankshaft, a spark & supply of fuel to start. The starter motor ,by a system of mechanical linkages rotates the crankshaft which provides the angular rotation required
Hypothetico-Deductive	SET PROBLEM SOLVING TASKS	

Figure 14.1
THE INFORMATION-GIVING STAGE OF THE EDUD TEACHING PROCESS

14.2.2 The EDUD criteria for determining the content of the intervention

In the EDUD model the content of the intervention is determined by the five categories of explanation defined in Chapter 8. Each category is seen as representative of an epistemic class considered necessary for the user's understanding of the domain as a whole. For each significant item of the target domain, explanation is provided for:

1. its definition
2. its function
3. its cause/effect relationships
4. its complex-derivational relationships
5. its hypothetico-deductive relationships

A detailed description of the knowledge decomposed according to these categories is given in Chapter 13.

The sequencing of explanation is based on psychological considerations which aim to ensure epistemic continuity. Thus explanations are hierarchically arranged and represented to the user in a way which emulates the temporal order of the information-processing operation which underlies the human learning process. Each node in the hierarchy represents a successive stage in the understanding process. Thus it is the ontology of understanding which guides the presentation of teaching sequences. By providing a 'genetic' organisation of explanation into a hierarchy, boundaries for the user's knowledge are provided and a focus for pedagogical activities is suggested. The hierarchy is, therefore, not merely a syntactic structure but represents logical links between the parts of an explanation and a framework which is an assertion about the fundamental organisation of the learning process. A full description of the structure of the EDUD hierarchy of understanding is given in Chapter 7.

14.2.3 The EDUD Teaching Strategy

Current teaching strategies incorporated into the design of ITS tend to select a single instructional strategy in accordance with the particular task which is to be taught. For example, the WHY system uses a Socratic dialogue management teaching method to tutor causes of rainfall, whilst SOPHIE (Bro82) uses a reactive environment with guided interactions to teach electronic trouble- shooting. An example of a system designed to reflect both domain-dependent and domain-independent principles is the Meno-Tutor which provides a general framework within which tutoring rules can be defined and tested. Its teaching strategy which is domain-independent is coupled with a domain-specific language generator in an attempt to formalise the discourse procedures developed earlier for GUIDON (Cla83).

In EDUD theory it is held that the act of learning about a single domain may incorporate a number of different processes and stages. To guide the learner through the different phases or stages of learning various, rather than a single instructional strategy should be integrated into system design. The selection of teaching strategy in the EDUD model is once again based on the view that a tutoring system should be capable of being used in three distinct phases. Each phase employs a different teaching strategy reflecting the goal sought in that phase.

Teaching Strategy for the Information-Giving Stage

The instructional aim in the first stage is for the user to learn the use and meaning of basic domain operators necessary for understanding a problem-space. The emphasis in the instruction is therefore on transmitting descriptive knowledge to enhance the understanding of procedures rather than on teaching process skills directly. It is important for example that information be presented in such a way as to ensure that the meaning as intended in a particular domain is preserved. The user therefore needs his attention to be directed to a stimulus in such a way as to suggest schemes for encoding it.

The EDUD program, therefore, operates, in this initial stage of functioning, in an instructional way with a directed learning strategy in which it takes the initiative (see Figure 14.1). Its instructional strategy is expository on the basis that students learn faster if they are told directly (Pir85).

The rationale for adopting this expository teaching strategy in the initial instruction as opposed to immediately placing the user in an exploratory type environment is that it is believed that by supplying him with an underlying structure before unleashing him into a browsing mode, the user will centre his attention on understanding the domain, becoming aware of its specific requirements and as a result be more conscious of the steps that bring about the solution. His search is, therefore, more highly selective in future problem-solving and analysis of a problem is by means of inference.

Teaching Strategy for the Problem-Solving Stage

The pedagogic strategy employed in the second stage is constructivist, that is, the user is encouraged to construct new knowledge from his existing knowledge. By this stage all the domain-relevant declarative knowledge necessary to solve a specific problem is known and the user now has to use this knowledge procedurally. It is believed that as a result of the instruction received in the first stage, the user will come to see basic relationships before making decisions and thus employ an analytic form of reasoning, seeing the reason for his decision as against attempting a series of trial and error decisions to see what result will be effected. In this way the risk of floundering and resorting to an unguided system of trial and error is avoided. The goal in this phase is for the user to gain insight into what makes problems solvable. In the implementation of the EDUD model this is encouraged by setting problems on which he may practice his problem solving skills in a directed way. Suitable practice of tasks is expected to enable the learner to acquire heuristics that reduce or eliminate the search he must carry out in solving problems. The belief being that performance improves with practice (Ros87). (It is acknowledged that in this stage it would be beneficial to provide a supportive environment in which the user may learn by doing or through simulation, however, because of the technical difficulties of so doing, this type of environment has not been implemented in the existing MEDUD program).

Teaching Strategy for use with the Articulate Expert System.

In the third stage the goal of the pedagogic strategy is to provide the user with a model that will enable him to move from domain skills to metacognitive strategies (Col88).

This is achieved by providing a model of heuristic problem- solving and decision making skills in the form of an articulate expert system. The user now has the appropriate knowledge structure with which to explore the expert knowledge base in a meaningful way. As he gains experience in the domain his capability for arranging his own learning processes improves and he engages in more self-instruction. In this stage he may therefore explore the topics of interest to him thus giving him control over the interaction which would be denied in a purely system-directed system. Questions such as 'why' drive explanations from within the expert system and can be used to demonstrate to the user, expert problem-solving behaviour.

14.2.4 EDUD control of the extent of the intervention

In the EDUD model explanation is given in assimilable increments (chunks) to provide for learning requirements. The size of each chunk attempts to reflect the limits of static memory structures, by including only that information which it is necessary for the user to assimilate at the relevant level in the EDUD hierarchy. Thus, if the user is functioning at the causal level, only information pertaining to the cause/effect relationship of the object under consideration will be offered to him.

However, the emphasis in the design of EDUD is not so much on the size of the chunk at intervention but rather on the conceptual context in which this chunk may be assimilated. Hence memory structures are not viewed as just static entities but as a dynamic reflection of the user's understanding of the domain. Therefore, the view of the size of the chunk is complimented with the notion of conceptual coherence with respect to this understanding.

Provision is also made within the EDUD model for a set of hints to assist the user on the basis that the more a hint restricts the number of directions a problem-solver might follow,(including the correct direction), the greater are the chances that the hint will assist him in solving the problem.

In the MEDUD program these hints are offered when a user has previously been exposed to the pertinent information but provides the incorrect answer to a related question. The assumption being that the user having made a mistake, is provided with the means to recover from the mistake and at the same time learn from having made the mistake in the first place. If the correct answer were to be provided immediately, there is a danger that the user would not develop the necessary skills for examining his own behaviour and look for the causes in his mistakes.

14.3 CONCLUSION

The goal of the Tutoring Module in the EDUD model is to teach subject material by leading the user to a better understanding of it through explanation. In order to achieve this objective the questions of when to intervene,what to discuss, which particular instructional strategy to use and how much to say in the tutorial interaction are addressed. In addition, consideration has been given to pedagogic and psychological issues and the way in which explanation effects such issues.

The Tutoring Module in the EDUD model is designed to support three distinct stages in system functioning.The initial stage explains the content of the domain so as to ensure that the user has the necessary conceptual framework for subsequent problem-solving. The second stage is geared to assist the user in converting and reorganising his knowledge in the context of procedures. Finally, explanation takes the form of a model which the user may observe to gain experience of heuristics and decision making skills.

The EDUD principles provide a framework for 'what' should be taught in each of the three stages by referencing the categories of explanation described in Chapter 8. The question of 'how' the explanation process should proceed is dealt with by emulating psychological processes and sequencing the explanations in a way which mimics the pragmatic rules of inferences as described in Chapter 10. It is this sequencing of explanation which ensures that epistemic faithfulness is achieved.

Chapter 15

AN EXAMPLE OF THE EDUD APPROACH IN ACTION

15.1 INTRODUCTION

The principles for ITS design expounded in this thesis have been implemented in a small Intelligent Tutoring System designed to teach the mechanics of a motor car's fuel system. The program called MEDUD has been written in Prolog and runs on an IBM AT system, using GEM graphics. (A listing of the program is given in Appendix A).

15.2 THE ARCHITECTURE OF MEDUD

The architecture of the MEDUD system reflects the separation of the tutoring system into the following distinct modules: (see Figure 15.1)

1. The Knowledge Base which contains the domain-specific articulate knowledge to be taught by the system.

2. The Tutoring Module, which contains the requisite information for determining the most appropriate teaching strategy. The tutor uses information both from the user and the User Model to control the tutorial interaction. It can present information to the user and request information from the user via the Interface. On the basis of the information which it receives, via the User Model, in response to questions/problems presented, it determines, in accordance with EDUD principles, which explanations to provide.

3. The User Model containing a trace of the user's progress through the EDUD levels of understanding in the form of dynamic records of the user's perceived level of cognitive functioning.

4. The User Interface, which provides text and graphics capabilities.

5. The Articulate Expert System, which has access to the

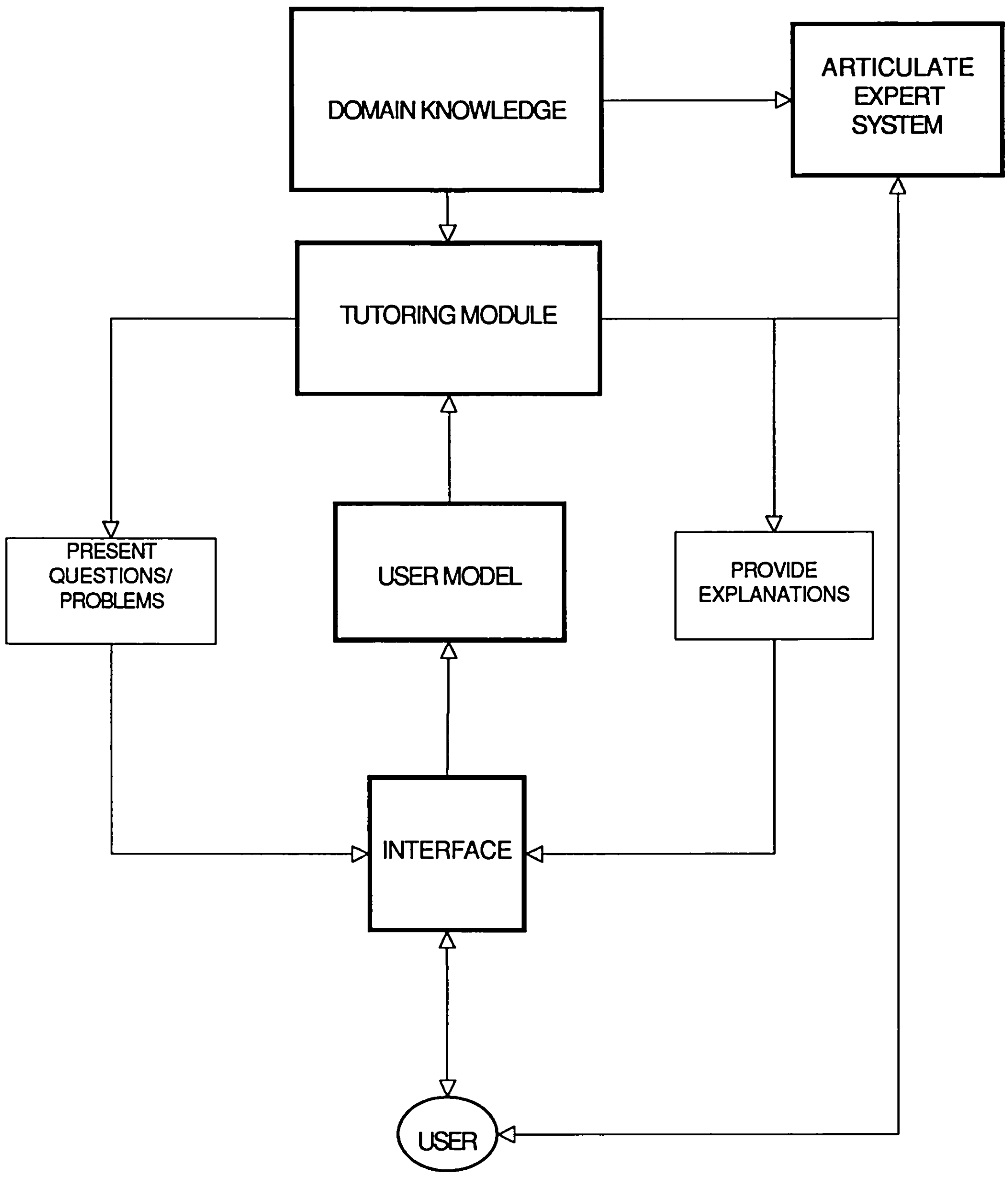

FIGURE 15.1
THE ARCHITECTURE OF MEDUD

information contained in the Knowledge Base, but which, in theory, should incorporate compiled, expert knowledge. This module is accessed either via the Tutoring Module or directly by the user.

15.3 REPRESENTATION OF DOMAIN KNOWLEDGE ABOUT A CAR'S FUEL SYSTEM IN MEDUD

The method for the decomposition of the body of target-knowledge about a car's fuel system used in MEDUD accords with the principles for knowledge-decomposition set out in Chapter 13 of this thesis.

The knowledge has been represented using a frame-like technique giving a complete hierarchical, semantic network of the body of knowledge in question. The network contains those domain concepts and knowledge 'chunks' that are considered to be most important, according to EDUD principles to facilitate understanding of the target-knowledge. The arrangement of the domain-knowledge into this hierarchical structure permits 'partial inheritance' of information between frames which enables limited inferences to be drawn. For example, it is possible to infer that the throttle-valve is part of the engine system, on the basis that the throttle-valve is part of the fuel system which in turn is part of the engine system.

The general structure of the semantic network for the knowledge in question is shown in Figure 13.2.

Each frame in MEDUD has the name of one of the parts that make up the fuel system and contains all the information it is considered necessary for the user to understand about that part. Thus, there are separate frames for the carburettor, fuel-pump, fuel-filter etc.

Within each frame there is a number of slots. Each slot contains information about the frame-object at a predefined level of abstraction as defined by the categories of explanation described in Chapter 7. Thus, each frame-object is defined in terms of its identity, function, cause/effect relationships, complex- derivational relationships and hypothetical states.

15.4 THE MEDUD USER MODEL

The method for representing the user's level of understanding at any particular moment in the MEDUD system is also based on a frame-like knowledge representation method.

Frames have been selected for representing the user in the MEDUD system as they allow the program to reflect the principles outlined in the EDUD theory. Each frame in the User Model reflects a cognitive structure pertaining to a specific concept in the domain. These frames are named in terms of their objective, that is, to assess understanding of a particular concept at various levels of the EDUD hierarchy (see Figure 15.2).

Frame Name	Slot	Value
Understanding-X	Figurative-Knowing	Yes/No
	Function	Yes/No
	Cause/Effect	Yes/No
	Complex-derivational	Yes/No
	Hypothetico-deductive	Yes/No

Figure 15.2
User Model Frame with Value Attachment to Slot

Each frame in the User Model contains a number of slots. Each of these slots represents one of the stages of understanding at which the user will be assessed according to the levels in the EDUD hierarchy of understanding. Thus there are separate slots to assess figurative knowing, function, cause/effect, complex-derivational and hypothetico-deductive levels of understanding. Attached to each of these slots is a value slot, which is dynamically updated to provide a record of the user's current level of understanding of that particular slot. Slot values are instantiated to Yes/No according to whether the system perceives the user to have attained that level of understanding or not (see Figure 15.3).

In order to obtain a value for the slot, the system must assess the user's level of understanding at a particular level in the EDUD hierarchy. This is achieved by attaching procedures to the relevant slots for which values need to be obtained. Thus, slots within frames either contain a Yes/No value or a procedure.

Frame Name	Slot	Procedure
Testing-X	Figurative-knowing	Consult Tutoring Module

Figure 15.3
User Model Frame with Procedural Attachment to Slot

15.5 THE MEDUD TUTORING MODULE

In the MEDUD program the Tutoring Module selects appropriate questions to ask the user, the answers to which will provide the instantiations for the yes/no values for the slots in the User Model frames. In addition the Tutoring Module drives suitable explanations which will serve to increase the user's understanding of the domain and determine which of the three teaching stages described in Chapter 14, that is, information- giving, problem-solving or observational should be supported.

In the information-giving stage, the basic procedure for determining the values of the slots in the User Model, as well as which explanation is appropriate based on the information received from the User Model, consists of an ordered set of questions and answers. The ordering of the questions within MEDUD reflect the way in which the pragmatic rules of inference are applied by the individual in the information-seeking process. They are therefore applied in a cyclical fashion at each level in the knowledge-decomposition hierarchy (see Figure 14.1).

In the problem-solving stage the basic procedure for determining the values of the slots in the User Model, as well as which explanation is appropriate based on the information received from the User Model, consists of an ordered set of production rules. These rules supply the dynamics by which the representation of the user's 'knowledge-level' is controlled. The rules in question have a general format, the domain-independence of which may be seen by examining the program code for MEDUD. However, for illustrative purposes an example is given of the rules applied to assess the User's understanding at the functional level as follows:

Rule A1:
Assess user's understanding of function.
If function not understood and user requested
information at this level;
And it is first attempt at answering the question
Then give a hint and re-ask the question.
If function not understood and user requested
information at this level;
And not first attempt then explain answer at functional
level, give information at complex-derivational level
And assess understanding at complex-derivational level.
If function understood and previous question complex-derivational
Then give information on complex-derivational and re-ask
complex-derivational level,
If function understood and previous question not complex-derivational
Then ask question at complex-derivational level.
If function not understood and user not requested
information at this level;
And did request information at definitional level then give information
function, complex-derivational,
If function not understood and user not requested
information at this level;
And not requested information at definitional level
Then assess understanding at definitional level.

A schematic representation of the functioning of the Tutoring Module in the problem-solving stage is depicted in the flow-chart in Figure 15.4.

In the observational stage of system functioning, the user is expected to explore the environment by himself. No representation of his actions is therefore recorded during this stage.

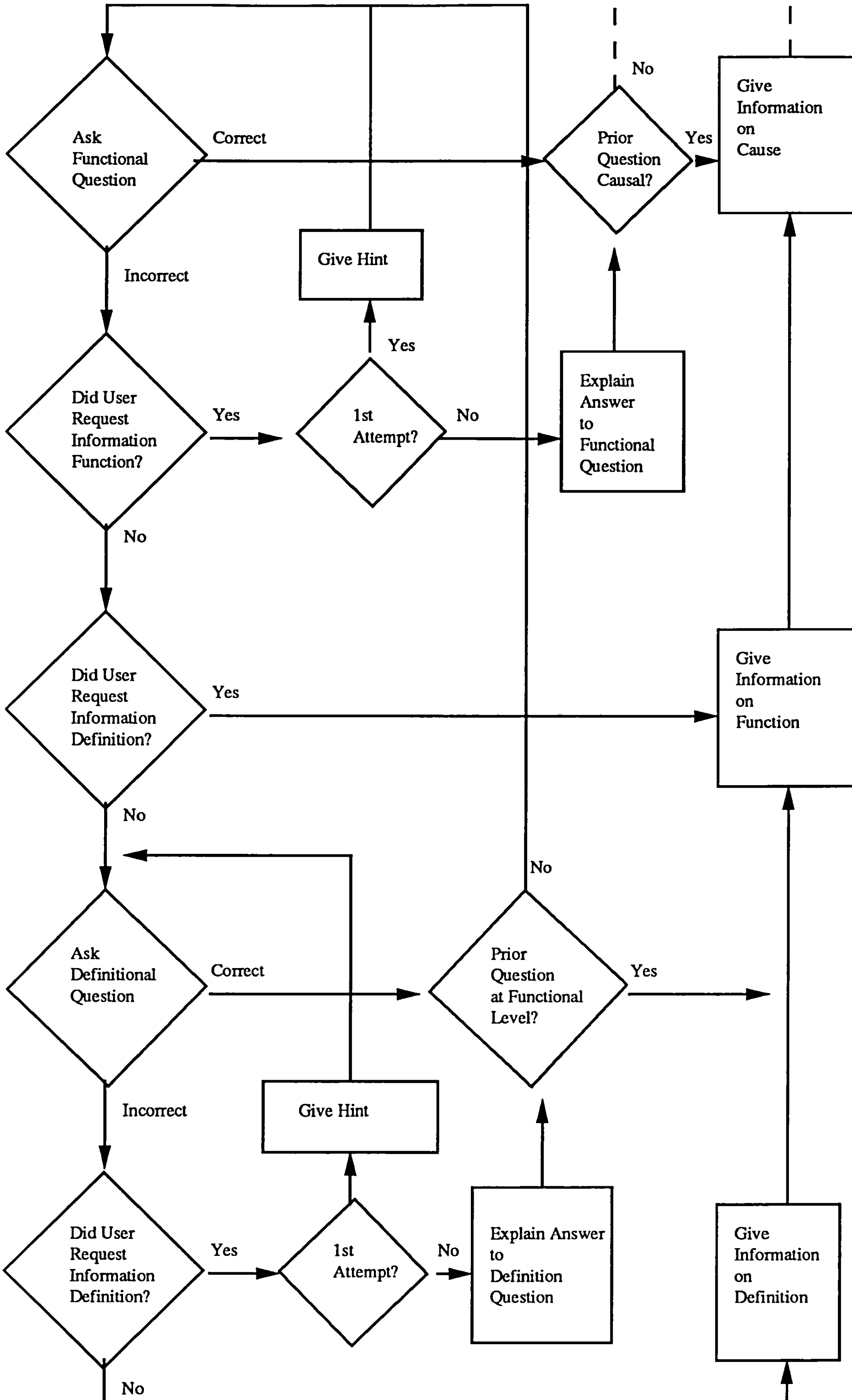

Figure 15.4
Flowchart of Problem-Stage of System Functioning in MEDUD

15.6 HOW MEDUD WORKS

THREE STAGES OF SYSTEM FUNCTIONING

MEDUD system functioning is divided into three separate phases, in accordance with the principles outlined in Chapter 14.

1. Initial Stage - Information-Giving Stage

In the initial stage of functioning, the Information-Giving Stage, the system's primary objective is to ensure that all the relevant domain concepts which may later be required for problem- solving are available to the user.

Accordingly, system-interaction commences with the system taking the initiative. The Tutoring Module, which at this stage adopts an expository teaching strategy, asks (via the interface) whether the user understands the task domain. This initial question gives the user the option of deciding either to be taught about the domain, or whether he feels he is sufficiently familiar with this information to be able to progress to a more advanced level of interaction. The general description of this process is given below:

Would you like to:
1. be taught about the fuel system, or,
2. do you feel you understand enough about it to answer a few questions?

The reason for providing the user with an option as to whether he wishes to be taught by the system or whether he would like to problem-solve in this initial stage, is to prevent any boredom and frustration which may arise if it is the case that the information is already known (or felt to be known), by the user. However, in providing the user with this option an important assumption is made, that is, that the user knows what he says he knows. In other words, if the user states that he understands a particular concept or relationship ,this is accepted as fact at this stage. This assumption, is not, however, as broad as it initially appears. Any misconceptions on the part of the user about the state of his knowledge will become apparent at a later stage of system functioning, where he will be required to demonstrate his ability to solve problems appertaining to this concept. Any inabilities to solve problems are highlighted then and, if required, corrective teaching done at that stage.

If the user indicates that he is familiar with the domain the User Model is updated to reflect that the user feels confident of his knowledge of this topic and system functioning is returned to the Tutoring Module, which initiates the second stage of system functioning, that is the problem-solving stage.

If the user indicates that he requires some tuition on the basic concepts in the domain he is given a brief introduction to the topic, see below:

Fine to start we will look at the basic facts about the fuel system

Introduction to the fuel system

In most cars the fuel system is petrol/air based
A petrol/air mixture is *EXPLOSIVE* when ignited

The explosions resulting from the ignited petrol/air mixture generate the energy which propels the car

The question is......what parts of the fuel system enable it to provide this EXPLOSIVE mixture?

Do you know what they are?

After the introduction to the topic the user is questioned as to whether he understands the concept at each level of abstraction as set out in the EDUD hierarchy of understanding. His replies at each of these levels are assessed (with reference to the EDUD hierarchy of understanding) and the User Model is updated accordingly. In this way a profile of the user's current level of understanding is constructed and made available for reference in the problem-solving stage.

The ordering of these questions and any subsequent explanations, follow the ordering specified in the EDUD hierarchy of understanding. Thus, if the domain concept to be considered is the fuel pump, the first question put to the user is whether he understands what the fuel pump is (Stage of Figurative Knowing) (see Figure 14.1).

At those levels at which the user requests additional information, the Tutoring Module will offer an appropriate explanation. The content of this explanation is defined by the appropriate category of explanation defined by the EDUD principles. Selective perception of the important features of the explanation is encouraged by the use of capital letters and other visual cues. Where possible the user is given the opportunity of seeing a graphics display of the information, rather than a textual one. See below and Figure 15.5.

O.K. Well the most important parts of the fuel system to know about are:

the storage tank
the fuel pump
the carburettor
the flexible hose
the petrol filter

Would you like to see a picture of the fuel system?

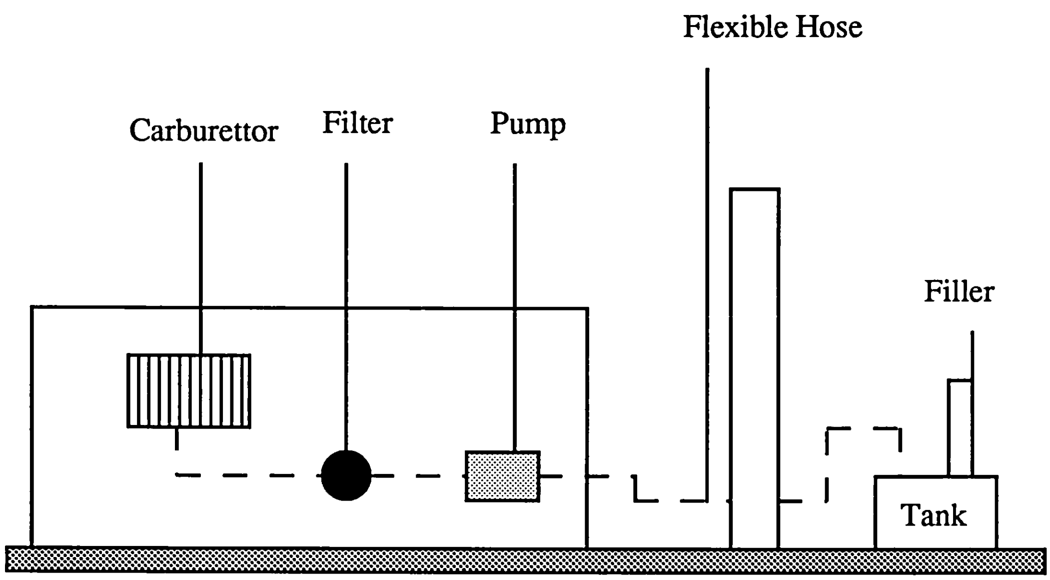

Figure 15.5
Graphical Representation of a car fuel system

When the user has either been exposed to or has indicated that he is familiar with all the content of the domain-knowledge at each level in the hierarchy of understanding, he is given the option of revising the concepts he has been taught or to continue on to the second phase of system functioning, the Problem-Solving Stage and have his understanding of the concepts tested by being given problems to solve.

2. The Problem-Solving Stage of System Functioning

In this stage, the objective of MEDUD is to increase the user's comprehension of and problem-solving capabilities in the target domain. Accordingly, the user is set a number of problems to solve in order to give him practice in compiling and strengthening his knowledge. The tutoring module takes the initiative and presents the user with a problem to solve.

<u>Selection of Initial Problem in Problem-Solving Stage</u>

The choice of the first problem to be given to the user to solve, is made with reference to the User Model. If the User Model indicates that the user intimated that he was familiar with all the concepts in the target-domain, then a problem is set to test understanding at an arbitrary level in the EDUD hierarchy of understanding, for example, at the Complex- Derivational level. (See question below).

There is fuel in the storage tank,
the flexible hose is intact
there is fuel at the inlet to the carburettor

........................but the engine system is spluttering

which part would you inspect next to try and diagnose the fault?

A correct answer will result in the user being returned to the Tutoring

Module, where he will be passed to the next stage in the Hierarchy of Understanding and finally to the next stage of system functioning.

An incorrect answer will result in the user being asked a question at a lower level in the hierarchy, for example, at the functional level. (See question below).

The fuel system fulfils 3 main functions with regard to the engine, which of the following describes these functions the best:
1. the storage, delivery and preparation of fuel for the engine or,
2. the preparation, ignition and distribution of fuel to the engine or
3. the filtering, ignition and delivery of fuel to the engine ?

If, however, the User Model indicates that the user is functioning at a level of understanding which is lower than the complex-derivational in the EDUD hierarchy of understanding then a question will be asked at the level at which he is perceived to be functioning. An example of a question at the identity level is given below:

Are all of the following parts of the fuel system?
1.the storage tank
2.the filter
3.the carburettor
4.the starter motor
5.the flexible hose

A correct answer to a question, will result in his being asked a question at the next highest level in the hierarchy and an incorrect answer will result in his being asked a question at a lower level.

Selection of Subsequent Problems in the Problem-Solving Stage

System Response to Correct Answers .
If the user can successfully solve a problem at any particular level of functioning,during the Problem-Solving Stage, he is then given a problem to solve at a higher level of the understanding hierarchy. This process of asking questions at higher and higher levels of the EDUD hierarchy of understanding is continued, until all questions at all levels in the hierarchy have been correctly answered. The user is then passed to the Observation Stage of system functioning where he may interact and observe an expert system in action.

System Response to Incorrect Answers.
When an incorrect answer is given to a question, the User Model must be consulted to determine what action is appropriate. If the User Model reports that the user has asked for information at that level before, and that this is his first attempt at answering the question, he is given a hint, on the basis that he may have forgotten the information and is re-asked the question. If, however, he has asked for information at this level and has tried to answer the question before unsuccessfully, he is given an explanation at this level and at all subsequent levels in the hierarchy, on the basis that until this level has been understood he will not be able to understand higher levels in the hierarchy.

If, however, the User Model reports that the user has not asked for information at that level, a question is asked at the next lowest level in the hierarchy, in an attempt to pinpoint the level at which he is functioning. This process is continued until it is felt that an accurate assessment of the user's level of functioning has been achieved.

Third Stage of System Functioning - The Observation Stage
In this stage the objective of the system is to increase the user's understanding of the domain through his observation of an expert

system which functions to solve a problem with the help of the user. The system therefore describes a symptom and hypothesises as to what the fault might be which is causing the symptom. If the user does not understand why a particular question is being asked by the system, he may ask WHY? this particular information is required (See below for description of a problem given)

A PROBLEM

A motor vehicle is exhibiting the symptom of spluttering. The aim of this exercise is to try to track down the fault .

Please answer the following questions

If you are not sure of why a question is being asked ask 'WHY?'

An important feature of the MEDUD system is the method of explanation which it uses to reply to WHY queries posed by the user during the Observation Stage of system functioning. Responses given to these questions do not merely consist of syntactic traces of the rules fired by the system, but by referencing the EDUD hierarchy of understanding and linking the explanation to the hierarchy, semantic explanations are offered (See below).

Is the storage tank in possession of sufficient fuel?

WHY?

Because the engine system requires the sufficient and correct mixture of fuel to prevent spluttering and this is provided by the fuel system.
However, in order for the fuel system to fulfil this function both it and its constituent parts must be functioning - the storage tank is a constituent part of the fuel system and is checked at this point because unless it is in possession of sufficient fuel then none of the subsequent functions of the fuel system can be achieved

---

15.7 INCORPORATION OF GENERIC PRINCIPLES IN MEDUD PROGRAM

Although the MEDUD program described here teaches about a motor car fuel system, the code of the program forms a shell so that it may be used with other knowledge-bases.

Thus, all objects (X) in the target-knowledge are defined in the knowledge-base at an abstract level as follows:

X is 'VALUE' ...... (to define the concept),
X 'DOES'....... (to describe behaviour at a functional level),
X function is 'ACHIEVED BY', 'PROCESS' (to describe behaviour at the causal level),
X performs this function because it is a 'REQUIREMENT' of Y (to describe behaviour at the complex-derivational level).
'IF'.........'THEN' (to describe hypothetico-deductive behaviour).

The rules incorporated in the knowledge base for describing the functionality of the concept are also defined at an abstract,domain-independent level as follows:

(concept) VALUE1 (is) : X and
VALUE2 (X consists of) : Z and
(X) DOES : Y and
(Y) ACHIEVED BY : process 1
(because) REQUIREMENT of W is Y and
IF: requirement W not met
by Y
THEN Symptom 1 results
and conclude X (or Z) malfunctioning

Rules for constraints on functionality are then defined as follows:

(constraint)Symptom 1
result of REQUIREMENT W not met :
because (not) Y ACHIEVED BY:
PROCESS 1 and
DOES Y PROCESS 1 and
VALUE Y CONSISTS OF : X Z and
VALUE IS: X and/or Z (malfunction)

Applying this information, for example, to the fuel-pump the variables become instantiated as follows:

(concept) (VALUE) IS: fuel pump
(VALUE) CONSISTS OF: diaphragm and one way valves
DOES: deliver (fuel)
ACHIEVED BY: PROCESS 1
because: REQUIREMENT of engine system is
delivery fuel
IF: fuel not delivered to engine system and
Symptom 1
THEN: conclude (possible malfunctioning)
fuel pump, diaphragm or one
way valve

Process 1: movement of diaphragm and one-way valves
Symptom 1: spluttering and choking

Defining information in this way has the advantage that it makes the inference procedures operational for multiple knowledge and sub-domains. For example, it is possible to remove the domain-knowledge about the fuel-pump and replace it with information about another part, for example, the carburettor.

In addition, to defining the knowledge-base at the abstract level, the User Model is also a self-contained generic mechanism, which may be incorporated in systems used for teaching in a number of knowledge areas.

15.8 APPRAISAL OF THE MEDUD SYSTEM

In the evaluation of the MEDUD system, a set of subject- independent questions derived from a set of axioms suggested as an appraisal method for ICAI systems by Ford are addressed (For88). The appraisal questions are divided into four broad categories, namely, subject knowledge, student knowledge, student control and mode of communication.

15.8.1. Appraisal of MEDUD Subject Knowledge

Q. Can the system answer arbitrary questions from the user about the subject?

A. The system's ability to answer student questions about the subject is limited. It does, however, allow the user to ask 'WHY?' a question is being asked in the Observational Stage of system functioning. These explanations are more than a trace of the rules fired by the system in reaching conclusions but are based on semantic considerations.

Q. Can the system give an explanation of a problem solution (including one of a problem posed by the user)?

A. The answer to the first part of this question is affirmative, the system can give an explanation to a problem solution as described in the answer to the previous question, however, it cannot respond to arbitrary problems posed by the user during an interactive session.

Q. Can the system give alternative explanations, using perhaps analogy?

A. No, at present alternative explanations cannot be provided by the system. However, this facility could be implemented easily at a later stage. As the explanation categories are already clearly defined, this would simply be a matter of providing a further example(s) within each category.

Q. Can it answer hypothetical questions, that is, questions not about the present situation but about some imagined situation relating to it?

A. Not at present, but again because the knowledge is clearly defined, this facility could be easily implemented by extending the rules in the knowledge base.

15.8.2 Appraisal of MEDUD User Knowledge

Q. Could the system give a report on the student's level of understanding?

A. Yes, at any point in the interaction, the system is capable of giving a report on the student's current level of understanding, together with 'predictions' of those relations which still need to be tutored and those which must have already been mastered in order to achieve the present level of understanding. This is achieved through the hierarchical arrangement of the MEDUD User Model whereby the student is assessed at each level of understanding for a particular target-domain.

Q. Are the system's explanations tailored to the user?
A. The system is indeed able to tailor its explanations to the user. Once an assessment has been made of the level at which the user is functioning, the appropriate explanation category, identified through its association with a particular level of understanding is evoked. Thus explanations are pitched at the correct level for a particular user and only those concepts which the system has some evidence that he understands are used.

Q. Does the system provide informative feedback?
A. The system provides informative feedback by referring to the records contained in the User Model and reacting to responses from the user accordingly. Thus, should the user make a mistake in the problem-solving stage, the User Model indicates where in the hierarchy the student shows a weakness, or has failed to request information and offers feedback accordingly.

Q. Are the problems presented by the system adapted to the user's needs?
A. Yes, MEDUD sets problems on the basis of the state of the User Model (which indicates the strengths and weaknesses of a user in terms of the concepts he knows and does not know, and hence suggests tasks to probe particular weaknesses).

15.8.3 Appraisal of MEDUD Student Control

Q. Does the system actively engage the user?
A. The system has been tested by a limited number of users which makes this question difficult to answer. However, it appears that by adapting the system's functioning to the user's ability it does engage the user.
Q. Can the user initiate some new area of investigation?
A. Indirectly, he can achieve some control in the sense that he can select

which of the three stages of system functioning he wishes to interact with.

Q. Does the system monitor such proposed changes, and comment upon them if they seem to be unwise?

A. Yes, although it does not comment on unwise selections at the time that the choice is made by the user. Thus, if the user does not ask for information in the Information-Giving Stage, the system monitors such a choice, and adapts its presentation accordingly. However, it does not overtly comment on this.

Q. Does the system intervene if the user appears to be having difficulty?

A. Again the system does not actively intervene but changes its level of functioning by selecting a different set of questions/explanations/ problems in a way which is transparent to the user.

15.8.4. Appraisal of MEDUD Mode of Communication

Q. Can the user express his inputs to the system in whatever way is most natural?

A. No, the EDUD program restricts the user to single word inputs.

Q. Does the system help if the user's input is not understandable by the system?

A. Yes, error messages are provided when the input is not intelligible to the system and the user is invited to try again.

Q. Are the system's outputs natural?

A. No the outputs provided by the MEDUD program appear contrived but are easily understandable. The fact that the output is produced from pre-stored information and this is evident to the user.

15.9 CONCLUSION

This chapter has described how the MEDUD program, a small tutoring system based on the principles of the EDUD model, has been implemented. It has also discussed the strengths and weaknesses of the system in the light of a set of criteria suggested by Ford for evaluation of tutoring systems.

Chapter 16

CONCLUSION

16.1 COMMENTS

During the course of this research, the conclusion has been drawn that whilst there are many theoretical and practical barriers to accomplishing the ideals of ITS, there is good reason to be optimistic for the future of this field.

Most ITSs built so far have been laboratory experiments primarily intended to demonstrate the feasibility of a particular design approach (Wen87). To move these systems into the real world generates a number of pragmatic constraints. These include the lack of available tools for ITS construction and the plethora of personnel, such as, teachers, computer scientists, students and domain experts required to develop an operational ITS. The EDUD model has attempted to contribute towards the solution of these practical problems by providing a tool in the form of a shell, which may facilitate construction of ITSs.

A major theoretical constraint on providing systems in the real world results from the lack of agreement on how to integrate the various components of the tutoring system in a single delivery system (Joh88). It is towards this objective that the EDUD model has attempted to make a contribution by providing a technique for integrating the tutoring system modules both structurally and operationally.

Furthermore, there is still the unresolved question of how best to evaluate ITSs. Few theoretical guidelines exist on what effective evaluation for these systems should consist of. In addition, the evaluation process is costly and time consuming (Bur88).

Many of the encouraging signs for the future of ITS are the same as those for CAI. Studies of the CAI literature (Kea83,Kem81,Orl83), have shown that the use of CAI leads to reduced training time and instructional costs. There is also a strong indication that in many cases students prefer taking a course by means of CAI rather than through conventional methods. These results should be interepreted with caution, however, when viewing them form the perspective of ITS, as experience with CAI has also shown that educators will only accept instructional technology which can withstand the passage of time and improve their teaching. These are pertinent criteria for ITSs and ones that would be difficult to fulfil at this stage (Shl88).

International programmes concerned with computer-based education such as DELTA(Whi87), organised by the Council of Ministers of the European Communities, continue the interest in the application of new computer-based technologies to learning and provide evidence of continued willingness to invest in new teaching technologies.

16.2 RESEARCH RESULTS

In this research a number of issues associated with ITS design were examined. The results provide both theoretical and practical contributions to the field and in particular to the design of the User Model, the Knowledge Base and the Tutoring Module. A summary of the results is given below.

(1) A set of principles derived from established disciplines, in particular, artificial intelligence, pedagogy and psychology relevant to the design of a shell for Intelligent Tutoring Systems has been identified. The principal motivation has been to develop a unifying framework for the design and operations of the User Model, the Knowledge Base and the Tutoring Module. The key notions of the framework developed using the identified principles are explanation and understanding. The theory underlying the framework proposes that learning can be seen as a progression from less to more advanced states of understanding.

(2) According to the proposed framework, the design and operations of the User Model, Knowledge Base and Tutoring Module can be unified via a hierarchy of explanations and a hierarchy of levels of understanding whereby one maps directly onto the other. These hierarchies enable a systematic decomposition of a body of knowledge and its communication to

the user via a linked set of explanation seeking questions.

(3) The hierarchy of understanding is composed of the following levels:

* the stage of figurative knowing,
* the stage of functional understanding,
* the stage of cause/effect understanding,
* the stage of complex-derivational understanding,
* the stage of hypothetico-deductive reasoning.

(4) A set of explanation seeking questions has been identified to enable a progression through the hierarchy of understanding. The set of questions associated with the hierarchy of understanding reflects the pragmatic rules of inference.

(5) The specified questions can also be used as general rules for knowledge decomposition enabling a body of knowledge to be structured and represented in a Knowledge Base. This knowledge decomposition method has been shown to be applicable across a number of knowledge domains.

(6) The hierarchy of understanding provides the basis for controlling teaching interactions whereby the student is presented with information appropriate to the currently valid level of understanding. Information stored in the User Model provides the criteria for judging the student's current level of understanding.

(7) The User Model based on the selected principles contributes to making the tutoring system more responsive to the needs of different individual users. It also permits system behaviour to reflect the transition of a user from novice, to competent, to a relatively-expert level of understanding.

(8) The framework has been implemented in a small tutoring system to demonstrate how the proposed design would work in practice.

(9) The principal advantages of the proposed shell for ITS include:

* ability to clearly separate domain-dependent and domain-independent aspects of knowledge communication,
* a structural and operational unification of the three crucial components of an ITS, i.e. User Model, Knowledge Base and Tutoring Module, and
* applicability across a number of knowledge domains.

16.3 LESSONS LEARNED

A number of lessons have been learned during this research work, some of which are specific to this particular research and others which apply more widely to the field as a whole. Listed below are some lessons that may help others avoid similar pitfalls or define better the nature of their study.

1. Knowledge elicitation and representation are time consuming activities and a tendency may exist to underestimate the difficulties involved in these processes in building an ITS. When the knowledge that has already been codified and packaged into the knowledge-base of a system is looked at, it may be difficult to recognise how much effort went into structuring the knowledge in that form (Par87). Although this pitfall is not peculiar to ITS design, because the issues which ITS design focuses on tend to be specific to the field, such as teaching strategy and representation of the user, the problems of eliciting and representing the target knowledge must not be underestimated.

The quality of the knowledge in the MEDUD program, and particularly in the articulate expert system are impoverished and could be improved.

2. It is difficult to evaluate the effectiveness of a tutoring program when there are few guidelines on what effective evaluation for ITSs should consist of. The field of ITS is still immature with the result that there is no standard set of evaluation methods appropriate to address such a question. Because building ITSs is still somewhat of an art and there are few ITSs that could be classified as 'finished', designers of these systems are more concerned with usefully guiding the development of their system than with determining whether they are effective educational end products. Even though educational evaluation is an established field,

methodologies have not been developed for evaluating educational systems that attempt to teach students to understand rather than simply to get the correct answer (Fry88). A difference in emphasis in the design of ITS and CAI is on the representation of cognitive processes occurring in students. Thus, the evaluators of CAI were able to make the pragmatic assumption that correct answers to questions were a good reflection of the student's internal processes, a luxury which is not afforded to ITS where the goal of evaluation is to discover how well the system teaches students knowledge and skills that support the processes of solving problems.

The MEDUD program was evaluated using criteria for evaluation of non 'real world' systems set out by Ford (For88). In addition, a small number of users were asked to use and comment on the program. However, the results of the evaluation procedure remain somewhat theoretical in the absence of any well-defined evaluation methodology.

3. The user modelling problem is a complicated one (Van88). Most design problems in computer science can be specified by describing the desired output of the program. Unfortunately this is not the case for designing the User Model for an ITS. User's abilities to acquire various types of knowledge may be attributed to many types of variables, such as their abilities, motivation, prior knowledge, learning styles, and many more. The task of constructing a User Model which includes all these aspects of behaviour and knowledge is clearly not a simple one.

The set of characteristics on which the user is assessed by the EDUD User Model is too narrow. Other characteristics of the user, other than his ability to traverse the EDUD understanding hierarchy, should be considered. Representation of characteristics such as his learning style and motivation would provide additional information for improving diagnostis and for remedial teaching purposes.

4. The representation of knowledge of teaching is a complex task. Because teaching a topic may be more difficult than 'knowing' the same topic, a teaching system is more complex than an expert system.

Whilst the EDUD model has identified three different teaching strategies to be used in the course of system functioning, the diversity of teaching strategies represented is still too limited. The Tutoring Module in EDUD suggests which levels of understanding in the EDUD hierarchy require tuition or further tuition. However, it does not provide remedial teaching for the student's difficulties at any significant depth.

5. The technology base of software is changing so rapidly, that intense effort must be expended to keep up with these changes as the ITS is being developed (Pso88).

A major problem experienced with the implementation of the MEDUD program resulted from the fact that at the outset of the project, the author chose micro-Prolog as the development language. It soon became apparent that the capabilities of this language were insufficient for developing anything other than the most trivial system. It was, therefore, decided to move to LPA Professional Prolog. This move from one software to another proved to be time consuming because programs had to be rewritten and debugged again. Even with the new version of Prolog it was found that its major limitation in so far as the needs of this project were concerned was that it used GSX graphics which displays graphics independently of the tutoring-text. This is a severe limitation on a system which purports to teach a subject such as car mechanics, where it would be a great advantage if the user could see a graphical representation of, and interact with, the various parts of the system. The tools that are commercially available for implementing powerful graphical interfaces, such as SunTools and the Macintosh Toolbox are very limited, commonly supporting basic window operations and simple graphics functions. There is a substantial gap between the capabilities of this software and those required to implement a tutorial program with the capabilities of systems such as STEAMER (Mil88).

16.4 SUGGESTIONS FOR FURTHER WORK

There are many directions for further research.

1. On the theoretical side the most important question to ask is whether the design of the Interface could be operationally and structurally linked to the

User Model, the Knowledge Base and the Tutoring Module via the principles stated in the EDUD model. It would be necessary to examine recent advances in the field of Human-Computer Interaction with a view to identifying established principles of how the notions of explanation and understanding have been approached in this discipline. The design of the Interface would then have to be linked to the hierarchies of understanding and explanation defined in the EDUD model. The objective would be to specify what kind of interfaces are best suited to which explanations. However, implementation of such an Interface would be impossible without competent graphical capabilities in the programming environment.

On the more practical side, the results presented in this thesis would benefit from further investigation of the following topics.

2. The MEDUD program generates instruction guided by explicit psychological and instructional theories. As such it provides a testbed for the EDUD theory based on these principles and would therefore be an attractive experimentation tool to test the levels of the EDUD hierarchy of understanding and the pragmatic rules of inference.

3. Because of the difficulties of knowledge representation mentioned earlier in the comments, it would be beneficial to apply the EDUD model to an already existing tutoring system. In this way the validity of the model itself could be better tested than as it stands at the present time, where the knowledge base is insufficient to provide any significant test of the model's validity.

4. It would be an advantage if the MEDUD program could undergo rigorous testing to establish where its strengths and weaknesses lie, both from the cognitive perspective, to evaluate the changes in users' knowledge and problem-solving skills and in order determine whether it is an effective educational end-product. The difficulties of carrying out such an evaluation have already been discussed.

5. The theory should be applied to other tasks, such as business problems. In this way the system could be extended to take advantage of its definition at the abstract level.

References

(Ach83) Achinstein, P., The Nature of Explanation. Oxford University Press, Inc., (1983).

(Ald79) Alderman,D.L., Evaluation of the TICCIT CAI System in the Community College. SIGCUE Bulletin, vol.13, no 3, pp5-17 (1979).

(And83) Anderson, J,R., The Architecture of Cognition. Cambridge, MA: Harvard University Press (1983).

(And85) Anderson, J.R., Boyle,C.F.,Farrell,R. & Reiser, B.J., Cognitive principles in the design of computer tutors. Proceedings of the Sixth Cognitive Science Society Conference, Boulder, Colorado, pp2-8 (1985)

(Bar76) Barr, A., Beard, M. & Atkinson, R.C., The computer as a tutorial laboratory: The Stanford BIP project. In International Journal of Man-Machine Studies, 8, pp567-596 (1976).

(Blo56) Bloom,B.S.,et al, Taxonomy of Educational Objectives; Handbook 1 - Cognitive Domain. David McKay, New York (1956).

(Bre88) Breuker,J., Coaching in Help Systems in Artificial Intelligence and Human Learning. In Self, J. (Ed), Intelligent Computer-Aided Instruction, Chapman & Hall Computing (1988).

(Bro82) Brown,J.S.,Burton,R.R. & De Kleer, J.,Pedagogical, natural language knowledge engineering techniques. In Sleeman, D.H. and Brown, J.S. (Eds.) Intelligent Tutoring Systems, New York: Academic Press (1982).

(Bro78) Brown, J.S.,& Burton, R.R. Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science,2,pp155-192 (1978).

(Bru66) Bruner,J.S., Toward a theory of instruction. Harvard University Press, Cambridge, Mass. (1966).

(Bro80) Brown,J.S., & Van Lehn,K., Repair Theory: A generative theory of bugs in procedural skills. Cognitive Science, 4, pp379-426 (1980).

(Bun77) Bunch, M.E.,Historical Orientation in Behaviour & Fact. In Marx, M.H. & Goodson, F.E. (Eds.) Theories in Contemporary Psychology, Second Edition, Macmillan Publishing Co., Inc. New York (1977)

(Bur82) Burton, R.R. & Brown, J.S. An investigation of computer coaching for informal learning activities. In Sleeman, D.H. & Brown, J.S. (Eds.) Intelligent tutoring systems. New York: Academic Press (1982)

(Bur88) Burton, R.R., The Environment Module of Intelligent Tutoring Systems. In Polson, M.C. & Richardson, J.J. (Eds.) Foundations of Intelligent Tutoring Systems, Lawrence Erlbaum Associates Publishers, Hillsdale, New Jersey (1988)

(Car70) Carbonell, J.R., AI in CAI: An artificial intelligence approach to computer assisted instruciton. IEEE Transactions on Man-Machine Systems, 11, pp190-202 (1970).

(Cha85) Charniak, E. & McDermott, D., Introduction to Artificial Intelligence. Addison-Wesley Publishing Company (1985).

(Cla79) Clancey, W.J., Tutoring rules for guiding a case method dialogue. In International Journal of Man Machine Studies, 11, pp25-49 (1979)

(Cla81) Clancey, W.J. & Letsinger, R., NEOMYCIN: Reconfiguring a rule- based expert system for application to teaching. Proceedings of the Seventh International Joint Conference on Artificial Intelligence, Vancouver, pp829-835 (1981).

(Cla83) Clancey, W.J., Overview of GUIDON. In Barr, A. & Feigenbaum, E. (Eds) The Handbook of Artificial Intelligence, Volume 2, Kaufmann, Los Altos, CA, pp267-278,(1983).

(Cla87) Clancey, W.J., Knowledge Based Tutoring: The GUIDON Program. The MIT Press, Cambridge, Massachusetts, (1987).

(Cla88) Clancey, W.J., The Role of Qualitative Models in Instruction. In Self, J.(Ed.) Artificial Intelligence and Human Learning, Intelligent Computer-Aided Instruction, Chapman & Hall Computing (1988).

(Col85) Collins,A., Warnock, E.,& Passafiume, J., Analysis and synthesis of tutorial dialogues. In Bower,G. (Ed.) The Psychology of Learning and Motivation (Vol 9), New York: Academic Press, (1985)

(Col88) Collins, A., The Computer as a Tool for Learning through Reflection. In Mandl,H. & Lesgold,A.(Eds.) Learning Issues for Intelligent Tutoring Systems, Preface, Springer-Verlag New York Inc.(1988).

(Ded86) Dede,C., A Review and Synthesis of Recent Research in Intelligent Computer-assisted Instruction. In International Journal of Man Machine Studies, 24, pp329-353 (1986)

(Dee76) Deese. J. Behaviour & Fact. In Marx, M.H. & Goodson, F.E. (Eds.) Theories in Contemporary Psychology, second edition, Macmillan Publishing Co., Inc. New York.

(duB87) du Boulay, B. & Sothcott, C., Computers Teaching Programming : An Introductory Survey of the Field. In Lawler,R.W. & Yazdani, M. (Eds.) Artificial Intelligence & Education,Volume One, Learning Environments and Tutoring Systems, Chapter 16, Ablex Publishing Corporation (1987)

(duB81) du Boulay, B., O'SHea,T. & Monk,J., The black box inside the glass box: Presenting computing concepts to novices. In International Journal of Man-Machine Studies, 14, pp237-249, (1981)

(duB88) du Boulay, B. & Sloman A., Bread Today, Jam Tomorrow: The Impact of AI on Education. In Proceedings of the Fifth International Conference on Technology and Education, Volume 1, Page 83, (1988).

(Fla63) Flavell, J.H., The Developmental Psychology of Jean Piaget.

Litton Educational Publishing, Inc., (1963).

(For88) Ford, L., The appraisal of an ICAI system. In Self, J. (Ed.) Artificial Intelligence and Human Learning, Intelligent Computer-Aided Instruction, Chapman & Hall Ltd (1988).

(Fry88) Frye,D., Littman,D.C., Soloway, E., The Next Wave of Problems in ITS: Confronting the "User Issues" of Interface Design and System Evaluation. In Psotka, J., Dan Massey L. & Mutter S.A., (Eds.) Intelligent Tutoring Systems, Lessons Learned, Lawrence Erlbaum Associates Publishers (1988)

(Gag62) Gagne, R.M., The acquisition of knowledge. In Psychological Review, 69, 355-365 (1962).

(Gar78) Gardner, H., Development Psychology - An Introduction. Little, Brown & Company (Canada) Ltd. (1978).

(Gil88) Gilmore,D. & Self,J., The application of machine learning to intelligent tutoring systems. In Self, J. (Ed.) Artificial Intelligence and Human Learning, Intelligent Computer-Aided Instruction, Chapman & Hall Ltd (1988).

(Gol77) Goldstein,I.P. & Carr,B. The computer as coach: An athletic paradigm for intellectual education. In Proceedings of the Annual Meeting of the Association for Computing Machinery, Seattle, Washington (1977).

(Ham78) Hamlyn, D.W., Experience and the Growth of Understanding.Routledge & Kegan Paul Ltd., (1978).

(Har73) Hartley, J.R.& Sleeman, D.H., Towards More Intelligent Teaching Systems. In International Journal of Man Machine Studies, 5, 215- 236 (1973)

(Har88) Hartley, J.R. & Smith, M. Question answering and explanation giving in on-line help systems. In Self,J. (Ed.) Artificial Intelligence and Human Learning, Intelligent Computer Aided Instruction, Chapman & Hall Computing, (1988)

(Hay74) Hayes, J.R. & Simon, H.A., Understanding written problem instructions. In Gregg, L.W. (Ed.) Knowledge and Cognition, Hillsdale, N.J.: Lawrence Erlbaum Associates, (1974)

(Hay83) Hayes-Roth, F., Waterman, D.A. & Lenat, D.B. Building Expert Systems. Addison-Wesley Publishing Company, Inc., (1983)

(Hay85) Hayes, J.R., Three Problems in Teaching General Skills. In Segal,J., Shipman,S. & Glaser,R. (Eds.) Thinking and Learning, Lawrence Erlbaum, Hillsdale, NJ (1985).

(Hil77) Hillix, W.A., Theories of Learning. In Marx, M.H. & Bunch, M.E. (Eds.) Fundamentals and Applications of Learning, Macmillan Publishing Co., Inc., New York,(1977)

(Hol84) Hollan, J.D., Hutchings, E.L., & Weitzman, L., STEAMER: an interactive inspectable simulation-based training system. In AI Magazine, 5,20, 15-27 (1984)

(Jac86) Jackson, P. , Explaining Expert Systems Behaviour. In Alvey IKBS Expert Systems Theme Workshop on Explanation, Report and Proceedings, pp83-95 March (1986).

(Jac76) Jack, R.D. & Engel, C.E., An Analytical Approach to Postgraduate Medical Education. In British Journal of Anaesthesia, 48,495,pp495-498 (1976)

(Joh84) Johnson, W.B. & Soloway, E., PROUST: Knowledge-based program understanding. In Proceedings of the seventh international conference on software engineering,369-380 (1984).

(Joh88) Johnson, W.B., Developing Expert System Knowledge Bases in Technical Training Environments. In Psotka, J., Dan Massey L. & Mutter S.A., (Eds.) Intelligent Tutoring Systems, Lessons Learned, Lawrence Erlbaum Associates Publishers (1988)

(Kea87) Kearsley,G.K. (Ed), Artificial Intelligence and Instruction, Applications and Methods. Addison-Wesley Publishing Company (1987)

(Kea83) Kearsley, G., Hunter, B. & Seidel, R.J., Two Decades of CBI research : What we have learned. Professional Paper 3-83, Alexandria, VA: Human Resources Research Organisation (1983)

(Kem81) Kemner-Richardson, S., Lamos, J.P. & West, A.S., The CAI decision handbook. Lowry AFB, CO: US Air Force Human Resources Laboratory (1981).

(Kim82) Kimball, R., A Self-Improving Tutor. In Sleeman,D.H. & Brown J.S. (Eds.) Intelligent Tutoring Systems, Academic Press. (1982)

(Lan83) Lantz, B.S.,Bregar, W.S. & Farley, A.M. An intelligent CAI system for teaching equation solving. In Journal of Computer-Based Instruction, 10, pp 35-42 (1983).

(Law87) Lawler,R.W., & Yazdani, M., Artificial Intelligence & Education,Volume One, Learning Environments and Tutoring Systems. Ablex Publishing Corporation (1987)

(Leh77) Lehnert,W.G.,The process of question answering. Lawrence Erlbaum Associates (1977)

(Les88) Lesgold, A., Curriculum for Intelligent Tutoring Systems. In Mandl,H. & Lesgold, A.(Eds.) Learning Issues for Intelligent Tutoring Systems, Chapter 6, Springer-Verlag (1988).

(Lin77) Lindsay,D.H. & Norman, D.A., Human Information Processing, An Introduction to Psychology. Academic Press, Inc. (1977)

(Man88) Mandle, H. & Lesgold, A.(Eds.). Learning Issues for Intelligent Tutoring Systems, Preface, Spinger-Verlag, New York, Inc. (1988).

(Mil82) Miller,M.L., A structural planning and debugging environment for elementary programming. In Sleeman,D.H. & Brown J.S. (Eds.) Intelligent Tutoring Systems, Academic Press. (1982)

(Mil88) Miller,J.R., The Role of Human-Computer Interaction in Intelligent Tutoring Systems. In Polson, M.C. & Richardson, J.J. (Eds.) Foundations of Intelligent Tutoring Systems, Lawrence Erlbaum Associates Publishers, Hillsdale, New Jersey (1988)

(Nec87) Neches,R., Langley,P., & Klahr, D., Learning Development and Production Systems. In Klahr,D.,Langley,P., & Neches, R.(Eds.) Production System Models of Learning and Development, Chapter 1, The MIT Press (1987)

(New72) Newell, A., & Simon, H.A, Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall (1972)

(New80) Newell,A., Reasoning, problem-solving and decision processes: The problem space as a fundamental category. In Nickerson, R .(Ed.), Attention & Performance VIII, Erlbaum, Hillsdale, NJ (1980).

(Nis87) Nisbett,R.E.,Fong,G.T.,Lehman,D.R. & Cheng,P.W., Teaching Reasoning. In Science, Vol.238 (1987).

(Ohl88) Ohlsson,S. & Langley,P., Psychological Evaluation of Path Hypotheses in Cognitive Diagnosis. In Mandl,H. & Lesgold,A. (Eds.) Learning Issues for Intelligent Tutoring Systems, Springer-Verlag New York Inc.(1988).

(Orl83) Orlansky, J., Effectiveness of CBI: A different finding. Electronic Learning, 6, pp 59-60 (1983)

(O'Sh79) O'Shea, T. A self-improving quadratic tutor. In International Journal of Man-Machine Studies, 11,97-124 (1979).

(Pap80) Papert, S., Mindstorms - Children, Computers, and Powerful Ideas.New York: Basic Books (1980)

(Par87) Parsaye, K. & Chignell, M., Expert Systems for Experts. John Wiley & Sons Inc (1987).

(Par89) Park, O., Perez, R.S. & Seidel, J., Applications and Methods. In Kearsley, G. (Ed.) Artificial Intelligence and Instruction. Addison-Wesley Publishing Company. (1989)

(Pay88) Payne, S.J., Methods and Mental Models in Theories of Cognitive Skill. In Self, J. (Ed.) Artificial Intelligence and Human Learning, Intelligent Computer-Aided Instruction, Chapman & Hall Computing (1988).

(Pir85) Pirolli, P.L. & Anderson, J.R., The role of learning from examples in the acquisition of recursive programming skill. Canadian Journal of Psychology, 39, pp240-272 (1985)

(Pso88) Psotka, J., Dan Massey L. & Mutter S.A., Intelligent Tutoring Systems, Lessons Learned. Lawrence Erlbaum Associates Publishers (1988)

(Pyl84) Pylyshyn,Z.W., Computation and Cognition:Towards a Foundation for Cognitive Science. Cambridge, Mass: MIT Press, (1984)

(Ric67) Rickman, H.P., Understanding and the Human Studies. Heinemann Educational Books Limited, London (1967).

(Ric83) Rich, E.A., Users are individuals: individualizing user models. In International Journal of Man-Machine Studies, 18,pp199-214 (1983)

(Rid88) Ridgway, J., Of course ICAI is impossible..worse though, it might be seditious. In Self, J., (Ed.) Artificial Intelligence and Human Learning, Intelligent Computer-Aided Instruction, Chapman & Hall Computing (1988).

(Ros87) Rosenbloom, P. & Newell, A., Learning by Chunking: A Production System Model of Practice. In Klahr, D., Langley, P & Neches R., (Eds.)

Production System Models of Learning and Development, The MIT Press (1987)

(Shl88) Schlechter, T.M., An Examination of the Research Evidence for Computer-Based Instruction. In Hartson, H. & Hix,R., (Eds.) Advances in Human-Computer Interaction, Volume 2, Ablex Publishing Corporation (1988)

(Sem86) Sembugamoorthy V. & Chandrasekaran, B., Functional Representation of Devices and Compilation of Diagnostic Problem- Solving Systems. In Kolodner, J.L. & Riesbeck C.K., (Eds.) Experience, Memory and Reasoning, Lawrence Erlbaum Associates, Inc. , (1986)

(Sho76) Shortcliffe, E.H., Computer-based Medical Consultations: MYCIN. American Elsevier (1976)

(Siu89) Siuru,W.D., Challenger: A domain-independent intelligent tutoring system. In IEEE page 77 Summer (1989)

(Sle82) Sleeman, D.H., Assessing aspects of competence in basic algebra. In Sleeman,D.H. & Brown J.S. (Eds.) Intelligent Tutoring Systems, Academic Press. (1982)

(Sle87) Sleeman,D.H. & Brown J.S. (Eds.) Intelligent Tutoring Systems. Academic Press (1987)

(Sle87b) Sleeman,D.H., PIXIE: a shell for developing intelligent tutoring systems. In Lawler, R., & Yazdani, M. (Eds.) Artificial Intelligence & Education,Volume One, Learning Environments and Tutoring Systems, Ablex Publishing Corporation (1987)

(Sle85) Sleeman, D.H., A user modelling front end subsystem. In International Journal of Man Machine Studies, 23, 71-88 (1985)

(Ste82) Stevens, A.L., Collins, A. & Goldin, S., Misconceptions in student's understanding. In Sleeman, D. & Brown, J.S. (Eds), Intelligent tutoring systems. New York: Academic Press (1982)

(Sup67) Suppes,P., Jerman, M. & Brian, D. Computer-assisted instruction at Stanford: 1966-68: Data, models and evaluation of arithmetic program. New York: Academic Press (1967)

(Sup81) Suppes, P., University-level computer-assisted instruction at Stanford; 1968-1980. Stanford, CA: Institute for Mathematical Studies in the Social Sciences, Stanford University (1981).

(Swa81) Swartout,W.R., Explaining and Justifying Expert Consulting Programs. In Proceedings of the International Joint Conference on Artificial Intelligence, 7, (1981)

(Van80) Van Fraasen, B.C., The Scientific Image. Oxford University Press, New York, (1980).

(Van83) Van Lehn,K. Competitive Argumentation in Computational Theories of Cognition. In Report Dn94903Y Xerox Palo Alto Research Center C. (1983)

(Van88) Van Lehn, K., Student Modeling. In Polson, M.C. & Richardson, J.J. (Eds.) Foundations of Intelligent Tutoring Systems, Lawrence Erlbaum Associates Publishers, Hillsdale, New Jersey (1988)

(Van81) Van Melle, W., Scott, A.,C., Bennett,J.,S. & Peairs, M.,A., The EMYCIN Manual. In Technical Report, Heuristic Programming Project, Stanford University (1981).

(Wen87) Wenger, E., Artificial Intelligence and Tutoring Systems. Morgan Kaufman Publishers, Inc., (1987).

(Wes77) Westcourt, K.T., Beard, M. & Gould, L., Knowledge-based CAI: CINS for Individualised curriculum sequencing. Technical Report 290. Stanford, CA: Institute for Mathematical Studies in the Social Sciences, Stanford Univeristy (1977).

(Whi87) Whiting, J., & Bell D.A. (Eds.) Tutoring and Monitoring Facilities for European Open Learning. Elsevier Science Publishers B.V., The Netherlands (1987)

(Wil87) Wilson, M., A Personal View of a Topical Meeting on Knowledge Representation. In AISB Quarterly Newsletter, Autumn, 63, Pg18 (1987).

(Woo61) Woodruff, A.D., The Learning Process. In Harris, T.L., & Schwahn, W.E., (Eds.) Selected Readings on The Learning Process,Oxford University Press, Inc.,(1961)

(Woo84) Woolf, B. & McDonald, D.D., Context dependent transitions in tutoring discourse. In Proceedings of National Conference on Artificial Intelligence, Kaufmann, Los Altos (1984).

(Yaz84) Yazdani, M., New Horizons in Educational Computing. Ellis Horwood/John Wiley (1984).

(Yaz87) Yazdani, M., Intelligent Tutoring Systems: An Overview. In Lawler,R.W. & Yazdani, M. (Eds.) Artificial Intelligence & Education,Volume One, Learning Environments and Tutoring Systems, Ablex Publishing Corporation (1987)

APPENDIX A

LISTING OF THE MEDUD PROGRAM

/* Before engaging the 3 stages of system functioning described in the EDUD model, the system and subject area are introduced to the user */

/* This clause calls and initialises the program */

```
medud :-  cuwind(&:),
kill(attempt),
close(tutorial),
close(escape),
kill(user_model),
assert(user_model(none,as,yet)),
assert(attempt(not,yet)),
crwind(tutorial,0,0,18,75),
escape_msg,
introduction,
user_choose_sub(_A),
```

/* The MEDUD program is introduced */

```
introduction :-  introduce_the_prog.
```

/* The user is invited to choose the particular topic (subject) they would like to study first */

```
user_choose_sub(_A) :-
question_sub(_B),
nl,  nl,
get_ans_sub(_A).

get_ans_sub(Val) :-
'R'(Ans),
valdef_sub(Ans,Val).

get_ans_sub(Val) :-
infomiss,
nl,  nl,
get_ans_sub(Val).
```

/* The user is invited to choose which particular section of the topic (subject) they would like to study */

```
user_choose_part(_A,_B) :-
question_part(_A),
nl,  nl,
get_ans_part(_A,_B).

get_ans_part(_A,B) :-
'R'(Ans),
put(12),
```

```
valdef_part(Ans,_A),
put(12).
```

/* The user is given the option of being taught about the topic he has chosen or being given a number of problems to solve */

```
user_select_mode(_A,_B) :-
question_mode(_A,_B),
nl,  nl,
get_ans_mode(_A,_B).

get_ans_mode(Subject,_B) :-
'R'(Ans),
(Ans = 1 ,
put(12) ,
substitution([Subject],[New]) ,
'PP'('Fine',to,start,we,will,look,at,the,basic,facts,about,the) , write(New) ,
nl ,
call(tutoring_module1(Subject)) ;
Ans = 2 ,
nl , nl , nl ,
write('Fine then try and answer the following questions') ,
nl , nl ,
call(tutoring_module2(Subject)) ,
nl , nl).

get_ans_mode(Subject,_B) :-
infomiss,
nl,  nl,
get_ans_mode(Subject,_B).
```

/* STAGE 1 OF SYSTEM FUNCTIONING -
THE INFORMATION-GIVING STAGE */

/* THE TUTORING MODULE FOR STAGE 1 */

/* If the user requests to be given information, a general introduction to the topic is given. He is then asked, at each of the levels in the EDUD hierarchy, whether he wishes information at that level and if so, such information, is given. The User Model is advised of his reply */

```
tutoring_module1(Subject) :-
write('              Introduction to the '),   substitution([Subject],[New]),
write(New),
nl,  nl,
introduction(Subject),
nl,  nl,
user_choose_info(definition,Subject),
user_choose_info(func,Subject),
user_choose_info(cause,Subject),
user_choose_info(cd,Subject).

user_choose_info(definition,Subject) :-
question_def(Subject),
nl,
```

```
get_ans_info(definition,Subject).

get_ans_info(definition,_A) :-
'R'(_C),
(_C = yes ,
nl , nl ;
_C = no ,
put(12) ,
assert(user_model(request_information,definition,_A)) ,
nl ,
substitution([_A],[_L]) ,
w rite('  O.K.  Well the most important parts of the') ,
write(' ') ,
write(_L) ,
nl ,
write(' to know about are') ,
nl ,
give_info(def,_A)).

get_ans_info(definition,_A) :-
infomiss,
nl,   nl,
get_ans_info(definition,_A).

user_choose_info(func,_A) :-
question_func(_A),
nl,
get_ans_info_func(_A).

get_ans_info_func(_A) :-
'R'(_C),
(_C = yes ;
_C = no ,
put(12) ,
assert(user_model(request_information,func,_A)) , give_info(func,_A)).

get_ans_info_func(_A) :-
infomiss,
 nl,   nl,
user_choose_info(func,_A).

user_choose_info(cause,_A) :-
question_cause(_A),
nl,
get_ans_info_cause(_A).

get_ans_info_cause(_A) :-
'R'(_C),
(_C = yes ; _C = no ,
put(12) ,
assert(user_model(request_information,cause,_A)) , give_info(process,_A) ,
put(12)).

get_ans_info_cause(_A) :-
infomiss,
```

```
nl,  nl,
get_ans_info_cause(_A).

user_choose_info(cd,Subject) :-
question_cd(Subject),
nl,
get_ans_info(cd,Subject).

 get_ans_info(cd,Subject) :-
'R'(Ans),
(Ans = yes ,
assert(user_model(request_information,cd,Subject)) ,
put(12) ,
nl , nl ,
'PP'('I',think,that,we,should,now,test,your, understanding) ,
nl , nl ,
tutoring_module2(Subject) ;
Ans = no ,
put(12) ,
assert(user_model(request_information,cd,Subject)) ,
give_info(cd,Subject) ,
revision_question(Subject)).

/* The user is given the choice of whether he would like to revise the material which has
been presented to him or whether he would like to progress to the problem-solving stage of
system functioning */


revision_question(Subject) :-
(nl , tab(5) ,
write('Now that you have been given all the information about the ') , nl ,
tab(25) ,
substitution([Subject],[System1]) ,
write(System1) ,
nl , nl ,
write('Would you like:') ,
nl ,
write('1: to go through this information again or ') ,
nl ,
write( '2: would you like to answer a few questions?') ,
nl , nl ,
write('Your choice: 1 or 2') ,
nl ,
'R'(Ans) , Ans = 1 ,
put(12) ,
call( tutoring_module1(Subject)) ;
Ans = 2 ,
nl , nl , nl ,
write('Fine, then try and answer the following questions') ,
nl , nl ,
tutoring_module2(Subject)).
```

```
/* STAGE 2 THE PROBLEM SOLVING STAGE */

/* THE TUTORING MODULE FOR STAGE 2 */

/* The Tutoring Module for Stage 2 is initialised */

tutoring_module2(Subject) :-
kill(attempt),
assert(attempt(bla,bla)),
kill(have_asked_question),
tutor(Subject).

/* The User Model is consulted and a problem-set to test the User's problem-solving ability
at the desired level */

tutor(Subject) :-
(user_model_record(X,(Y , Subject)) ;
user_model_record(X,(Subject , Y))),
get_ans(Y,Subject).

/*  The subject matter is tutored according to the rules set out in the EDUD Tutoring
Module*/

get_ans(Level,Subject) :-
(question(Level,Subject) , assert(have_asked_question(Level,Subject)) ,
nl ,
'R'(_C) ,
valdef(Level,Subject,_C) ,
check_level(Leve l,Subject,Nextlevel) ,
put(12) ,
nl , nl , nl ,
(\+ have_asked_question(Nextlevel,Subject) ,
put(12) ,
write('That is correct') ,
nl  , nl ,
write('Now try and answer the following question') ,
nl , nl ,
get_ans(Nextlevel,Subject) ;
Nextlevel = cd ,
tutoring_module 3(Subject) ;
put(12) ,
write('That is correct, so now -') ,
nl ,
give_info(Nextlevel,Subject) ,
write('Now try and answer the question..') ,
get_ans(Nextlevel,Subject)) ;
user_model_record(1,(Level , Subject)) ,
\+ attempt(Level,yes) ,
give_hint(Level,Subject) ,
assert(attempt(Level,yes)) ,
 nl , nl ,
write('Now, try and answer the question') ,
nl , nl ,
get_ans(Level,Subject) ;
user_model_record(1,(Level , Subject)) ,
```

```
attempt(Level,yes) ,
put(12) ,
write('No, this is still incorrect, the answer is ') ,
nl , nl ,
msg(M) ,
explain_answer(Level,Subject) ,
msg(M) ,
check_level(Level,Subject,Nextlevel) , (have_asked_question(Nextlevel,Subject) ,
nl , write( 'Now...') ,
give_info(Nextlevel,Subject) ,
nl , write('Try and answer the following question') ,
nl ,
get_ans(Nextlevel,Subject) ;
\+ have_asked_question(Nextlevel,Subject) ,
nl , write('Now try and answer this question') ,
nl ,
get_ans(Nextlevel,Subject)) ;
user_ model_record(4,(Level , Subject)) ,
check_level(Level,Subject,Nextlevel) ,
write('No, remember the....') ,
give_info(Level,Subject) ,
nl , write('Now...') ,
give_info(Nextlevel,Subject) ,
nl , write('So, try and answer....') ,
nl ,
get_ans(Nextlevel,Subject) ;
user_model_record(3,(Level , Subject)) ,
droplevel(Level,Lowerlevel) ,
put(12) ,
write('No, I am afraid you are wrong -
try and answer this question') ,
get_ans(Lowerlevel,Subject)).


droplevel(Level,def) :-  Level = func.
droplevel(Level,func) :-  Level = process.
droplevel(Level,process) :-  Level = cd.
droplevel(Level,def2) :-  Level = def.



check_level(Level,Subject,process) :-  Level = func.
check_level(Level,Subject,func) :-  Level = def.
check_level(Level,Subject,cd) :-  Level = process.
check_level(Level,Subject,cd) :-  Level = cd.


/* STAGE 3 - THE OBSERVATIONAL STAGE */

/* THE TUTORING MODULE FOR STAGE 3 */

/* THE TUTORING MODULE FOR THIS STAGE CALLS THE ARTICULATE
EXPERT SYSTEM */
```

```
tutoring_module3(_A) :-
artic_exp_system(_A).

/* THE USER MODEL */


user_model_record(4,(def , Subject)) :-
\+ user_model(request_information,definition,Subject).
user_model_record(1,(func , Subject)) :-
user_model(request_information,func,Subject).
user_model_record(1,(process , Subject)) :-
user_model(request_information,cause,Subject).
user_model_record(1,(def , Subject)) :-
user_model(request_information,definition,Subject).
user_model_record(3,(func , Subject)) :-
\+ user_model(request_information,func,Subject),
\+ user_model(request_information,definition,Subject).
user_model_record(4,(func , Subject)) :-
\+ user_model(request_information,func,Subject),
user_model(request_information,definition,Subject).
user_model_record(4,(process , Subject)) :-
\+ user_model(request_information,cause,Subject),
user_model(request_information,func,Subject).
user_model_record(3,(process , Subject)) :-
\+ user_model(request_information,cause,Subject),
\+ user_model(request_information,func,Subject).
user_model_record(1,(cd , Subject)) :-
user_model(request_information,cd,Subject).
user_model_record(4,(cd , Subject)) :-
\+ user_model(request_information,cd,Subject),
user_model(request_information,cause,Subject).
user_model_record(3,(cd , Subject)) :-
\+ user_model(request_information,cd,Subject),
\+ user_model(request_information,func,Subject).


\* THE KNOWLEDGE BASE */


/* Describing an object at the Definitional Level*/


give_info(def,_A) :-
findall((_F , _G),
value1(_A,_F,_G),[_H]),
substitution([_A],[Z]),
nl,
substitution([_H],[Y]),
bagof(_I,value2(_A,_J,_I),_K),
nl,
dejar(_K),
nl,  nl,
msg3.
```

Describing an object at a Functional Level*/

```
give_info(func,_A) :-
bagof(_F,_G ^ _W ^
does(_A,_F,_G,_W),_H),
substitution([_A],[_Z]),
nl, tab(15),
write('The functions of the '), write(_Z),
write(' are the'),
nl, nl,
tab(15),
substitution([_H],[T]), write(T),
nl, nl,
tab(22),
'PP'(of,fuel,for,the,engine),
nl,
tab(15), write(*****************************************),
nl,
msg(Message).

/* Describing an object at the cause/effect level*/

give_info(process,_A) :-
findall((_H , _L , _I , _J),
achieved_by(_A,_H,_L,_I,_J),_K),
substitution([_A],[_Q]),
write('To understand how the '),
write(_Q), write(' '),
'PP'(functions,consider,its,constituent,parts,and,their,functions,and,link,each,part,with,a
,function,'.'),
nl,
'PP'('So',we,have),
nl, write('   THE PARTS'),
tab(25), write('THE FUNCTIONS'),
nl, nl,
findall((_G , Verb , _M),
does(_A(_G,Verb,_M)),_W),
dejar(_W), nl, nl,
msg(Message),
put(12), 'PP'('However',there,are,also,certain,
      constraints,on,the,functioning,of,the),
write(_Q),
write('.'),
write('In order to:'),
nl, nl,
dejar(_K),
nl, nl,
msg(Message),
nl,
write('The process by which the'),
write(' '),
write(_Q),
nl,
write(' achieves its function is as follows :'),
msg(Message),
```

```
nl,
process(_A(X)).


process(Subject(X)) :-
process(Subject(Number,[Actor,Action,Substance,from,From,to,To])),
substitution([Actor],[Actor1]),  substitution([Substance],[Substance1]),
substitution([From],[From1]),  substitution([To],[To1]),
substitution([Action],[Action1]),
write(Number),
write(' '),
write(Actor1),
write(' '),
write(Action1),
write(' '),
write(Substance1),
write(' '),
write('from '),
nl,
write(From1),
write(' to '),
write(To1),
 nl,  nl,
msg(Message),
Number = 'FIFTH',
nl,  nl.

/* Describing an object at the  complex-derivational level */

give_info(cd,Subject) :-
bagof(_F,_G ^ _W ^ does(Subject,_F,_G,_W),_H),
findall((Part , Condition),
achieved_by(Subject,Verb,Substance,Part,Condition),_K),
has_symptom(Subject,System,Symptom),  value1(Subject,Value,System),
substitution([Subject],[Subject1]),
substitution([Value],[Value1]),
substitution([System],[System1]),
write('Because the'),  write(' '),
write(Subject1),  write(' is  '),
write(Value1),  write(' '),
write(System1),  write(' and the '),
nl,  write(System1),  write(' requires the '),
substitution([_H],[Z]),  write(Z),
nl,  write(' of fuel and the  '),
write(Subject1),  write(' '),  'PP'(is,capable,of,meeting,this,requirement,'.'),
nl,
msg(Message),
write('  Failure of the'),
write(' '),  write(Subject1),  write(' '),
write('to meet this requirement '),
write('will possibly'),
nl,  tab(15),
write('result in '),
substitution([Symptom],[Symptom1]),
write(Symptom1),
```

```
write(' the '),  write(System1),
nl,  nl,
tab(20),
write(*************************************),  msg(Message),
put(12).


/* Clauses defining the procedural knowledge used in the Articulate Expert System */

has_value(Subject,Slot,V) :-
value1(Subject,Slot,V).
has_value(Subject,Slot,V) :-
value2(Subject,Slot,V).

function(_A,_B,_C) :-
value2(_A,_D,_E),
does1(_E,_F,_G),
on(_F,_B),
is_working(_A,_E).

requirement_met(E).
requirement_met(_A) :-
requirement(_A,_B,_C),
value1(_D,_E,_A),
function(_D,_B,_C).

is_working(_A,_C) :-
achieved_by(_A,_F,_B,_C,_D),
substitution([_C],[X]),
substitution([_D],[Z]),
nl,
write('Is '),  write(X),
write(' '),  write(Z),
write(?),
nl,  nl,
write('Your choice: yes/no or why'),
nl,
'R'(_E),
 (_E = no ,
nl ,
write('  Then this is the problem that is causing') ,
nl ,
write(' the spluttering, I suggest you have it checked' ) ,
nl ,
end_message2 ;
_E = why ,
tell_why_cd(_G,_C) ;
put(12) ,
_C = the_flexible_hose ,
end_message ;
_C = the_one_way_valves ,
end_message).
```

```
\*Introducing the Articulate Expert System */


artic_exp_system(_A) :-
write('You seem to be quite familiar with this subject'),
nl,
write('So take a look at the following:'),
nl,
nl,
nl,
write('                    A PROBLEM'),
nl,
nl,
tab(6),
write('The aim of this exercise is to try to track down a fault'),
nl,
tab(25),
write(' in the '),
substitution([_A],[X]),
write(X),
nl,
nl,
tab(8),
write('In an attempt to locate the problem, the system will ask you'),
nl,
tab(8),
write(' to check certain parts of the '),
write(X),
write(' which may'),
nl,
tab(22),
write(' be possibly causing the fault'),
nl,
tab(2),
write('If you are not sure of why the system is asking a particular question'),
nl,
write(' type WHY'),
nl,
tab(20),
write(**********************************),
nl,
msg(Message),
put(12),
set_problem(_A).


/* The Domain Specific Information used in this program */

/* Introducing the program to the user */

introduce_the_prog :-
nl,
write('        THIS IS A TUTORIAL ABOUT MOTOR CAR MECHANICS'),  nl,  nl,

write('    The engine system of a motor car is made up of a number'),
```

```
nl,
write('    of sub systems, each of which it is necessary to understand'),
nl,
write('if one is to understand the functioning of the engine system as a whole.'),
nl,  nl,  nl,
write(' Please choose which of the following sub-systems of the engine you would like to
look at first'),
nl,  nl,  nl,  nl.

/* The question for asking which subject the user wishes to study */


question_sub(_B) :-
write(' 1 :FUEL SYSTEM    2: MECHANICAL SYSTEM    3:ELECTRICAL SYSTEM?
  '),
nl,  nl,
write('Your choice: 1,2 or 3 ').

/* Acknowledging the  user's choice of subject to study and advising the user whether they
have made a valid choice of subject*/

valdef1(Val) :-
(valdef_sub(1,Val) ,
put(12) ,
write('You have chosen to study the fuel system') ,
nl ,
nl , ! ;
valdef_sub(X,Y) ,
write(' I am sorry but this system has not been ') ,
 nl ,
write(' implemented as yet, please choose option 1 - the fuel system  ') , nl ,
get_ans_sub(Val)).

/* The question for asking which particular section of the topic the user wishes to study */

question_part(_A) :-  nl,
write('Would you like to study the '),
substitution([_A],[New]),
write(New),
write(' as a whole'),
nl,
write('or study a particular part?'),
nl,
write(' (at the present time the only part available for study'),
nl,
write('is the fuel pump)'),  nl,  nl,  write('your choice:'),  nl,  write(' 1. study the '),
write(New),
write(' as a whole'),
nl,
write(' 2. study the fuel pump'),
nl.

/* The question for asking the User whether they would like to be taught about the system,
or attempt problem-solving */
```

```
question_mode(_A,_B) :-
write('Would you like to'),
nl,  nl,
write('(1) be taught about the '),
substitution([_A],[New]),
write(New),
write(' or:'),
nl,
write('(2) do you understand enough to answer a few questions?'),  nl,  nl,
write('Your choice: 1 or 2'),
nl.

/* Introducing the subject to the user */

introduction(fuel_system) :-
write('      In most cars the fuel system is petrol/air based.'),
nl,  nl,
write('      A petrol/air mixture is *EXPLOSIVE* when ignited  '),  nl,  nl,
write('      The explosions resulting from the ignited petrol/air'),  nl,
write('      mixture generate the energy which propels the car  '),  nl,  nl.

introduction(fuel_pump) :-
write(' The fuel pump in modern cars may be electrical or mechanical'),
nl,
write('the fuel pump which we are discussing in this tutorial is mechanical').

introduction(fuel_pump) :-
write(' The fuel pump in modern cars may be electrical or mechanical'),
nl,
write('the fuel pump which we are discussing in this tutorial is mechanical').

user_model(none,as,yet).

set_problem(_A) :-
nl,
symptom_of(_A,E,S),
write('A motor vehicle is exhibiting the symptom of '),  substitution([S],[N]),
write(N),
nl,
write('Please answer the following questions'),  symptom_of(_A,E,spluttering),

/* QUESTIONS PUT TO THE USER TO ASSESS HIS UNDERSTANDING AT EACH
LEVEL IN THE EDUD HIERARCHY OF UNDERSTANDING */


question_def(fuel_pump) :-
write('Do you know what parts make up the fuel pump?'),
nl,  nl,
write('Your answer: yes/no '),
nl.

question_def(fuel_system) :-
write(' The question is.....what parts of the fuel system'),
nl,
```

```
write(' enable it to provide this EXPLOSIVE mixture?'),
nl,  nl,
write('       Do you know what they are?'),
nl,  nl,
write('Your answer: yes/no '),
nl.

question_func(A) :-
tab(12),
write('Looking at these parts, can you think'),
nl,
tab(12),
write('what functions the '),
substitution([A],[New]),
write(New),
write(' may perform?'),
nl,  nl,
write('Your choice: yes/no').

question_cause(A) :-
nl,
tab(5),
write('Now that you know what the parts and the functions of the '),   nl,
substitution([A],[New]),
tab(8),
write(New),
write(' are, can you link these parts and functions'),
nl,  tab(8),
write(' to see HOW the '),   write(New),
write('  achieves its function ?'),
nl,  nl,  write('Your choice: yes/no').


question_cd(fuel_system) :-
tab(8),
write('Now that you know what the parts of the fuel system are'),   nl,
tab(8),
write(' what it does and how it does it - do you see why it is'),
nl,  tab(8),
write(' an essential part of the engine system?'),
nl,  nl,
write('Your choice: yes/no').

question_cd(fuel_pump) :-
tab(8),
write(' Now that you know what the parts of the fuel pump are'),
nl,
tab(8),
write(' what it does and how it does it - do you see why it is'),
nl,  tab(8),
write('an essential part of the fuel system?'),
nl,  nl,
write('Your choice: yes/no').
```

/* QUESTIONS PUT TO THE USER TO TEST HIS UNDERSTANDING AT EACH OF THE LEVELS IN THE EDUD HIERARCHY */

```
question(def,fuel_system) :-
nl,
write('Are all of the following parts of the fuel system ?'),
nl,
write('1. the storage tank'),
nl,
write('2. the filter'),
nl,
write('3. the carburettor'),
nl,
write('4. the starter motor'),
nl,
write('5. the flexible hose ?'),
nl,  nl,
write('Your choice: yes or no'),
nl,

question(def,fuel_pump) :-
nl,
write('Are all of the following parts of the fuel pump?'),
nl,
write('1. the diaphragm'),
nl,
write('2. the one way valves'),
nl,
write('3. the carburettor'),
nl,  nl,
write('Your choice: yes or no'),
nl,
routine1(def,fuel_pump).


routine1(def,Subject) :-
nl,
'R'(Ans),
valdef(def,Subject,Ans),
question(def2,Subject).


question(def2,fuel_system) :-
nl,
tab(20),
write('Which part is not part of the fuel system?'),
 nl,  nl,
write('Your choice: 1. the storage tank'),
nl,
tab(13),
write('2. the filter'),
nl,
tab(13),
write('3. the carburettor'),
```

```
 nl,
tab(13),
write('4. the starter motor'),
  nl,
  tab(13),
  write('5. the flexible hose'),
  nl.

question(def2,fuel_pump) :-
nl,
tab(20),
write('Which part is not part of the fuel pump'),
nl,  nl,
write('Your choice : 1 the diaphragm'),
nl,
tab(13),
write('2. the one way valves'),
nl,
tab(13),
write('3. the carburettor'),
nl.
routine1(def,fuel_system),
!.

routine2(def,Subject) :-
(nl , '
R'(Ans) ,
valdef(def,Subject,Ans) ,
routine3(def,Subject) ;
explain_answer(def,Subject) ,
nl).

routine3(def,Subject) :-
question(def2,Subject).

question(func,fuel_pump) :-
nl,
write('which of the following describes the fuel pump  function the best:'),
nl,
write('(a) the pumping and directing of fuel to the engine or'),
nl,
write('(b) the pumping of air to the engine'),
nl,
write('Your choice: a or b'),
nl.

question(func,fuel_system) :-
nl,  write('The fuel system fulfills 3 main functions with regard to the engine'),
nl,
write('which of the following describes these functions the best:'),  nl,
write('(a) the storage deliverance and preparation of fuel for the engine or'),
nl,
write('(b) the preparation ignition and distribution of fuel to the engine or'),
nl,
write('(c) the filtering, ignition and delivery of fuel for the engine?'),
```

```
nl,  nl,
write('Your choice: a,b or c'),
nl.

question(process,fuel_pump) :-
nl,
tab(20),
write('There is fuel in the storage tank,'),
nl,
tab(20),
write('the flexible hose is intact'),
nl,  nl,
tab(20),
write('......but the engine system is spluttering'),
nl,  nl,
write('which part would you inspect next to try and diagnose the fault?'),
nl,  nl,
write('a: the carburettor, b. the fuel pump, c. the manifold'),
nl,  nl,
write('Your choice: a,b, or c:').

question(process,fuel_system) :-
nl,
tab(20),
write('There is fuel in the storage tank,'),
nl,
tab(22),
write('the flexible hose is intact'),
nl,
tab(15),
write(' there is fuel at the inlet to the carburettor'),
nl,  nl,
tab(20),
write('.......... but the engine system is spluttering'),
nl,  nl,
write('which part would you inspect next to try and diagnose the fault ?'),
nl,  nl,
write('a. the carburettor, b. the fuel pump, c.the manifold'),
nl,  nl,
write('Your choice: a, b or c').

question(cd,fuel_system) :-
write(' WHY would you have checked the carburettor at this point?'),  nl,  nl,
write('1.  Because it is part of the fuel system which supplies the'),  nl,
write('    engine system with the correct mixture of fuel and air'),  nl,
write('    spluttering of the engine system would indicate'),
nl,
write('    incorrect mixture caused by carburettor malfunctioning'),  nl,
write('  or'),
nl,
write('2. Because all the other parts are not part of the fuel system'),
nl,  write('  or'),
 nl,
write('3. Because the carburettor is part of the mechanical system'),  nl,  nl,
write('Your choice:  1,2,3'),
```

```
nl.

question(cd,fuel_pump) :-
nl,
write('Why would you have tested the fuel pump at this point'),
nl,  nl,
write('1: because ........'),
nl,
write('   or'),
nl,
write('2: because.......'),
nl,  write('   or'),
nl,
write('3: because......'),
nl,  nl,
write('Your choice: 1,2,or 3'),
nl.

/* THE EXPLANATION FACILITIES */

/* These are the HINTS that are given for each of the various levels in the EDUD Hierarchy
of Understanding */

give_hint(func,fuel_system) :-
write('Remember that the '),
substitution([fuel_system],[New]),
write(New),   write(' consists of'),
nl,
give_info(def,fuel_system),
nl,  write('Think of what function each of these parts'),
nl,  write('performs and then consider what the overall function of the system might be').

give_hint(func,fuel_pump) :-
nl,
write('This is the wrong answer'),
nl,
write('This is because you have not remembered the parts of the fuel pump'),
nl,
write('and what they do'),
nl,
write('Are these parts only capable of pumping fuel?'),
nl,  write('Try and answer the question again'),   assert(attempt(func,yes)),
get_ans(func,fuel_pump),  nl.

give_hint(def,Subject) :-
assert(attempt(def,yes)),
put(12),
'PP'('You',have,given,the,wrong,answer),
tab(5),  write('Here is a  HINT'),
nl,  nl,
tab(15),
write('Think again about the parts of the '),  substitution([_A],[New]),
write(New),
write(' '),
write(' that you may have seen in the picture '),
```

```
msg3,
nl,  nl,
tab(20),
write('and try again '),
nl,
get_ans(def,Subject).

give_hint(process,fuel_pump) :-   'PP'('This',is,not,the,correct,answer,-,here,is,a,hint),
nl,  nl,
proc_expl_first(Subject).

give_hint(process,fuel_system) :-
nl,
write('This is not the correct answer - here is a hint'),
nl,  nl,
proc_expl_first(fuel_system),
msg(Message),
nl,  nl,
proc_expl_second(fuel_system),
msg(Message),
nl,  nl,
proc_expl_third(fuel_system),
msg(Message),
nl,  nl,
write('Do we have confirmation that the fuel has been successfully'),  nl,
write('processsed through the carburettor?'),
nl,
write('Try to answer the question again:- '),
nl,  nl,
assert(attempt(process,yes)),
get_ans(process,fuel_system).

give_hint(func,fuel_system) :-
'PP'('This',is,the,wrong,answer),
'PP'('This',is,because,you,have,not,remembered,the,
      parts,of,the,fuel,system),
'PP'(and,what,they,do),
'PP'('Are',any,of,them,capable,of,ignition,?),
'PP'('Try',and,answer,the,question,again),
nl.

give_hint(cd,Subject) :-
nl,
write('Think about the importance of this part to the wider environment....'),
msg(Message),
nl.

/* Explaining the answers to questions at each of the levels in the EDUD hierarchy */


explain_answer(def,fuel_pump) :-
nl,
write('You are wrong.....the carburettor is not part of the fuel pump'),  nl,
write('It is important to remember this otherwise you'),
nl,  write('may have difficulty in answering questions later on'),  nl.
```

```
explain_answer2(def,fuel_system) :-
substitution([_A],[X]),
write('You are wrong....the starter motor is not part of the fuel system'),
nl,
'PP'('It',is,important,to,remember,this,otherwise,you),
'PP'(may,have,difficulty,in,answering,the,next,question),
nl.

explain_answer2(def,fuel_pump) :-
substitution([_A],[X]),
write('You are wrong......the carburettor is not part of the fuel pump'),   nl,
write('It is important to remember this otherwise you'),
nl,
write('may have difficulty in answering the next question'),
nl.


explain_answer(func,fuel_system) :-
'PP'('PREPARATION',',','STORAGE',and,'DELIVERY',of,fuel),
'PP'(not,to,be,confused,with,the,function,of,igniting,the,fuel),
nl,
write('Remember the parts of the system are'),
nl,   'PP'(the,fuel,tank,whose,function,is,storage,of,the,fuel),
nl,   'PP'(the,mechanical,pump,whose,function,is,delivery,of,the,fuel),   nl,
'PP'(and,the,carburettor,whose,function,is,preparation,of,the,fuel),   nl,   nl.

explain_answer(func,fuel_pump) :-
nl,
write('The answer is the pumping and directing of fuel'),
nl,
write('Remember the parts of the fuel pump are the diaphragm and the'),
nl,
write('one way valves'),
nl,
write('the diaphragm pushes the fuel and the one way valves direct the fuel'),
nl,   nl,
write('now take a look at how the fuel pump achieves its function'),   nl.

explain_answer(process,fuel_pump) :-
write('Here would follow an explanation of how the fuel pump'),
nl,
write(works).

explain_answer(process,_A) :-
exp_proc2(_A,process),
nl.

exp_proc2(_A,process) :-
_A = fuel_system,
nl,
tab(5),
nl,
write('arrived at by looking at the process by which the fuel system'),
```

```
nl,
write('achieves its function'),
nl,  process(_A('FIRST',[_C,_D,_E,from,_F,to,_G])),  substitution([_C],[C1]),
substitution([_E],[E1]),  substitution([_F],[F1]),  substitution([_G],[G1]),
write('First '),
write(C1),
write(' '),
write(_D),
write(' '),
write(E1),
write(' '),
nl,
write('from '),
write(F1),
write(' to'),
write(' '),
write(G1),
nl,
proc_expl_first(_A),
nl,  nl,
process(_A('SECOND',[_C1,_D1,_E1,from,_F1,to,_G1])),  write('Second '),
forall((on(X,[_C1,_D1,_E1,from,_F1,to,_G1]) , substitution([X],[Y])),(write(Y) ,
tab(1))),
nl,  proc_expl_second(_A),
nl,  nl,
process(_A('THIRD',[_C2,_D2,_E2,from,_F2,to,_G2])),
write('Third '),  forall((on(X,[_C2,_D2,_E2,from,_F2,to,_G2]) ,
substitution([X],[Y])),(write(Y) ,
tab(1))),
nl,
msg(Message),
proc_expl_third(_A),
nl,  nl,
process(_A('FOURTH',[_C3,_D3,_E3,from,_F3,to,_G3])),  write('Fourth '),
forall((on(X,[_C3,_D3,_E3,from,_F3,to,_G3]) , substitution([X],[Y])),(write(Y) ,
tab(1))),
nl,  nl,
proc_expl_fourth(_A),
nl,
process(_A('FIFTH',[_C4,_D4,_E4,from,_F4,to,_G4])),
write('Fifth '),
forall((on(X,[_C4,_D4,_E4,from,_F4,to,_G4]) , substitution([X],[Y])),
(write(Y) , tab(1))),
nl,
msg(Message),
nl,
tab(10),  'PP'('We',do,not,have,confirmation,that,the,fuel,has,passed),
tab(8),
write('successfully through the carburettor'),
write(' so this is the next'),
nl,  tab(30),
write('part to check').


proc_expl_first(fuel_system) :-
```

```
tab(15),
write('WE KNOW THAT THERE IS FUEL IN THE FUEL TANK....SO'),
nl,
tab(15),
write('the first stage of the process has been achieved').

proc_expl_first(fuel_system) :-
msg(Message),
nl,  nl,
proc_expl_second(Subject),
msg(Message),
nl,  nl,
proc_expl_third(Subject),
msg(Message),
nl,  nl,  'PP'('Do',we,have,confirmation,that,the,fuel,has,been,successfully),
'PP'(processed,through,the,carburettor,?,'Try',to,answer,the,question),
'PP'(again,-),
nl,  nl.

proc_expl_first(fuel_pump) :-
nl,
write('Here would follow a hint and the question be re-asked'),  nl,  nl,
write('Try to answer the question again').

exp_proc2(fuel_pump,process) :-
nl,  nl,
write('The way in which the fuel pump achieves its function is as follows'),
nl,
process(fuel_pump(X)).

proc_expl_second(fuel_system) :-
tab(15),
write('THE FUEL HAS REACHED THE INLET PIPE TO THE CARBURETTOR
..SO'),
nl,
tab(15),
write('the flexible hose must have carried the fuel successfully'),  nl,
tab(15),
write('and must therefore be intact').

proc_expl_third(fuel_system) :-
tab(15),
write(' THE FUEL HAS ARRIVED SUCCESSFULLY AT THE'),
nl,
tab(15),
write(' INLET PIPE.....SO'),
nl,  tab(15),
write(' it must have been filtered successfully through the filter').

proc_expl_fourth(_A) :-
tab(15),
write('THE FUEL HAS ARRIVED SUCCESSFULLY AT THE'),
nl,
tab(15),
write('INLET TO THE CARBURETTOR....SO'),
```

```
 nl,  tab(15),
write(' the mechanical pump must have pumped it there successfully'),
nl.

proc_expl(fifth,['We do not have confirmation that this ~M          stage has','~M~J
been successfully completed','
~M       ~M~J
The fuel has not been passed on from ~M
the carburettor','~M~J
to the inlet manifold we ~M
therefore cannot conclude','~M~J ~M
that the carburettor has successfully ~M
completed its part of the process']).

explain_answer(cd,Subject) :-
nl,  write('1'),
nl,
give_info(cd,Subject).

/* Explaining WHY a particular question has been asked */

tell_why_cd(_A,_B) :-
value2(_C,_D,_B),
value1(_C,_E,_A),
has_symptom(_A,_F),
requirement(_A,_G,_H),
achieved_by(_C,_I,_J,_B,_K),  process(_C(_L,[_B,_M,_N,from,_O,to,_P])),
substitution([_A],[X]),
nl,  tab(5),
write('Because the '),  write(X),
write(' requires the '),
substitution([_G],[R]),
write(R),
substitution([_H],[S]),
write(' '),  write(S),
write(' to prevent '),  write(_F),
write(' and this is'),  write(' provided by the '),  substitution([_C],[Q]),
write(Q),
nl,
tell_why_cont(_C,_B).

tell_why_cont(_A,_B) :-
value2(_A,_E,_B),
process(_A(_F,[_B,_G,_H,from,_I,to,_J])),
achieved_by(_A,_K,_L,_B,_M),
substitution([_A],[N]),
substitution([_B],[O]),
substitution([_M],[P]),
nl,  tab(8),
write('However,in order for the'),
write(' '),  write(N),
write(' '),  write('to fulfill this function'),
nl,  tab(10),
write(' both it and '),  'PP'(its,constituent,parts,must,be,functioning,-),
nl,  tab(25),  write(O),
```

```
nl,  write(' is a constituent part of '),
write(N),  write(' and is checked at this point '),
nl,  tab(2),
write('because unless it '),
write(P),
nl,  tab(25),
write(' then none of'),
nl,  tab(3),
write(' the subsequent functions of the '),
write(N),
write(' can be achieved'),
nl,
is_working(_A,_B).



routine1(def,Subject) :-
explain_answer(def1,Subject),
question(def2,Subject).

/* THE DATA BASE */


value1(fuel_system,part_of_the,engine_system).
value2(fuel_system,has_a_part,the_storage_tank).
value2(fuel_system,has_a_part,the_fuel_pump).
value2(fuel_system,has_a_part,the_carburettor).
value2(fuel_system,has_a_part,the_flexible_hose).
value2(fuel_system,has_a_part,the_filter). value2(fuel_pump,has_a_part,diaphragm).
value2(fuel_pump,has_a_part,the_one_way_valves).
value1(fuel_pump,part_of_the,fuel_system).

does(the_storage_tank,stores,the_fuel). does(the_fuel_pump,delivers,the_fuel).
does(the_carburettor,prepares,the_fuel). does(the_flexible_hose,delivers,the_fuel).
does(the_filter,delivers,the_fuel).
does(fuel_system,storage,the_fuel_to_be_used_by,the_engine).
does(fuel_system,preparation,the_fuel_for,the_engine).
does(fuel_system,delivery,the_fuel_to,the_engine).
does(fuel_system,[storage,pumping,preparation,deliver],[fuel]).
does([the_carburettor,mix,the_air]). does([the_fuel_pump,pumping,the_fuel]).
does([the_storage_tank,stores,the_fuel]).
does(fuel_pump,pumping,the_fuel_to_be_used_by,the_engine).
does(fuel_pump,delivery,the_fuel_to_be_used_by,the_engine).
does(fuel_system(the_storage_tank,stores,the_fuel)).
does(fuel_system(the_fuel_pump,delivers,the_fuel)).
does(fuel_system(the_carburettor,prepares,the_fuel)).
does(fuel_system(the_flexible_hose,delivers,the_fuel)).
does(fuel_system(the_filter,delivers,the_fuel)).
does(fuel_pump(the_diaphragm,pushes,the_fuel)).
does(fuel_pump(the_one_way_valves,direct,the_fuel)).

does1(the_carburettor,preparation,the_fuel). does1(the_fuel_pump,delivery,the_fuel).
does1(the_flexible_hose,delivery,the_fuel). does1(the_storage_tank,storage,the_fuel).
does1(the_one_way_valves,direct,the_fuel). does1(diaphragm,pushes,the_fuel).
```

```
process(fuel_system('FIRST',[the_storage_tank,supplies,sufficient_fuel,from,external_w
orld,to,the_flexible_hose])).
process(fuel_system('SECOND',[the_flexible_hose,carries,fuel,from,the_storage_tank,to,
the_filter])).
process(fuel_system('THIRD',[the_filter,filters,fuel,from,the_flexible_hose,to,the_mecha
nical_pump])).
process(fuel_system('FOURTH',[the_fuel_pump,pumps,fuel,from,the_flexible_hose,to,t
he_carburettor])).
process(fuel_system('FIFTH',[the_carburettor,mixes,fuel_with_air_and_passes_it,from,t
he_carburettor,to,the_inlet_manifold])).
process(fuel_pump('FIRST',[diaphragm,increases,the_pressure_in_pump_chamber,from,
low,to,high])).
process(fuel_pump('SECOND',[the_diaphragm,draws_in,the_fuel_via_one_way_valve,fr
om,fuel_tank,to,the_pump_chamber])).
process(fuel_pump('THIRD',[the_pressure,is_decreased,in_the_pump_chamber,from,hig
h,to,low_by_distortion_of_the_diaphragm])).
process(fuel_pump('FOURTH',[the_pressure,is_increased,in_the_pump_chamber,from,l
ow,to,high])).
process(fuel_pump('FIFTH',[the_one_way_valves,direct,the_fuel,from,fuel_tank,to,the_
carburettor])).

achieved_by(fuel_system,'STORE',the_fuel,the_storage_tank,has_in_store_sufficient_fue
l).
achieved_by(fuel_system,'DELIVER',the_fuel,the_fuel_pump,is_working).
achieved_by(fuel_system,'DELIVER',the_fuel,the_filter,is_clear).
achieved_by(fuel_system,'PREPARE',the_fuel,the_carburettor,mixes_fuel_and_air_to_co
rrect_proportions).
achieved_by(fuel_system,'DELIVER',the_fuel,the_flexible_hose,is_in_tact).
achieved_by(fuel_pump,'PUMP',the_fuel,fuel_pump,must_be_working).
achieved_by(fuel_pump,'PUMP',the_fuel,diaphragm,must_be_working).
achieved_by(fuel_pump,'DIRECT',the_fuel,the_one_way_valves,must_be_working).

valdef_def(Subject,yes).
valdef_def(Subject,no).
valdef_cd(yes).
valdef_cd(no).
valdef(cd,fuel_system,1).
valdef(cd,fuel_pump,1).
valdef(function,fuel_system,a).
valdef(process,fuel_system,a).
valdef(process,5,yes).
valdef(func,fuel_system,a).
valdef(def,fuel_system,no).
valdef(def,fuel_system,4).
valdef(process,fuel_pump,b).
valdef(def,fuel_pump,no).
valdef(def,fuel_pump,3).

symptom_of(fuel_system,engine_system,spluttering).
symptom_of(fuel_pump,fuel_system,spluttering).
symptom_of(fuel_system,engine_system,spluttering).
symptom_of(fuel_pump,fuel_system,spluttering).

requirement(engine_system,[storage,preparation,delivery],the_fuel).
requirement(fuel_system,[pushes,direct],the_fuel).
```

```
has_symptom(fuel_system,engine_system,spluttering_of).
has_symptom('mechanical-pump',jammed,'leaking-petrol').
has_symptom('petrol-filter',blocked,spluttering).
has_symptom('one-way-valve',blocked,'petrol-overflow').
has_symptom(diaphragm,jammed,'petrol-overflow').
has_symptom(fuel_pump,fuel_system,
        lack_or_overflow_of_petrol_in).
has_symptom(engine_system,spluttering). has_symptom(fuel_system,spluttering).

/* THE INTERFACE */

/* The clauses 'dejar' and 'substitution' make the program more readable for the user */

dejar(_A) :-
(_A = [the_storage_tank,the_fuel_pump,the_carburettor,the_flexible_hose,the_filter] ;
_A = [the_storage_tank,the_fuel_pump,the_filter,the_carburettor,the_flexible_hose]),
write('the storage tank'),
nl,
write('the fuel pump'),
nl,
write('the carburettor'),
nl,
write('the flexible hose'),
nl,
write('the petrol filter').

dejar(_A) :-
_A = [(the_storage_tank , stores , the_fuel),(the_fuel_pump , delivers ,
the_fuel),(the_carburettor , prepares , the_fuel),(the_fl exible_hose , delivers ,
the_fuel),(the_filter , delivers , the_fuel)],   write('The storage tank'),
tab(19),   write('STORES the fuel'),
nl,  write('the fuel pump'),
tab(22),   write('DELIVERS the fuel'),
nl,  write('the carburettor'),
tab(20),   write('PREPARES the fuel'),
nl,  write('the flexible hose'),
tab(18),   write('DELIVERS the fuel'),
nl,  write('the filter'),
tab(25),   write('PREPARES the fuel').

dejar(_A) :-
 _A = [('STORE' , the_fuel , the_storage_tank ,
has_in_store_sufficient_fuel),('DELIVER' , the_fuel ,
the_fuel_pump , is_working),( 'DELIVER' , the_fuel , the_filter , is_clear),('PREPARE' ,
the_fuel , the_carburettor , mixes_fuel_and_air_to_correct_proportions),( 'DELIVER' ,
the_fuel , the_flexible_hose , is_in_tact)],
write('STORE the fuel'),
tab(4),   write('the storage tank'),
tab(6),   write('must have in store sufficient fuel'),
nl,  write('DELIVER the fuel'),
tab(2),   write('the petrol filter'),
tab(5),   write('must be clear,'),
nl,  tab(18),   write('the fuel pump'),
tab(9),   write('must be working and'),
```

```
nl,  tab(18),  write('the flexible hose'),
tab(5),  write('must be intact'),
nl,  write('PREPARE the fuel'),
tab(2),  write('the carburettor'),
tab(7),  write('must mix the air and fuel'),
nl,  tab(40),  write('to the correct proportions').

dejar(A) :-
A = [diaphragm,the_one_way_valves],
write(' diaphragm and the one way valves').

dejar(A) :-  A = [(the_diaphragm , pushes , the_fuel),(the_one_way_valves , direct ,
the_fuel)],
write('the diaphragm'),
tab(22),  write('pushes the fuel'),
nl,  write('the one way valves'),
tab(17),  write('direct the fuel').

dejar(A) :-
A = [('PUMP' , the_fuel , fuel_pump , must_be_working),('PUMP' , the_fuel ,
diaphragm , must_be_working),('DIRECT' , the_fuel , th e_one_way_valves ,
must_be_working)],
write('PUSH the fuel'),  tab(6),
write('the diaphragm must be working'),
nl,  write('DIRECT the fuel'),
tab(4),  write('the one way valves must be working').

word(supplies).
word(carries).
word(fuel).
word(filters).
word(mixes).
word(pumps).
word(from).
word(to).
word(diaphragm).

substitution([spluttering_of],['spluttering of']).
substitution([lack_or_overflow_of_petrol_in],['lack or overflow of petrol in']).
substitution([fuel_system],['fuel system']).
substitution([[storage,preparation,delivery]],['STORAGE,  PREPARATION  and
DELIVERY']). substitution([the_storage_tank],['the storage tank']).
substitution([(part_of_the , engine_system)],['part of the engine system']).
substitution([(part_of_the , fuel_system)],['part of the fuel system']).
substitution([sufficient_fuel],['sufficient fuel']).
substitution([the_flexible_hose],['the flexible hose']).
substitution([the_carburettor],['the carburettor']).
substitution([external_world],['external world']).
substitution([the_filter],['the petrol filter']).
substitution([the_fuel_pump],['the fuel pump']).
substitution([fuel_with_air_and_passes_it],['fuel with air and passes it']).
substitution([the_mechanical_pump],['the mechanical pump']).
substitution([the_inlet_manifold],['the inlet manifold']).
substitution([part_of_the],['part of the']).
substitution([engine_system],['engine system']).
```

```
substitution([has_in_store_sufficient_fuel],['in possession of sufficient fuel']).
substitution([the_fuel],['the fuel']).
substitution([is_working],[working]).
substitution([is_clear],[clear]).
substitution([mixes_fuel_and_air_to_correct_proportions],['mixing the fuel and air to the
required proportions']).
substitution([is_in_tact],[intact]).
substitution([[sufficient,and,correct,mixture,of]],['sufficient and correct mixture of']).
substitution([spluttering],[spluttering]).
substitution([[pumping,delivery]],['pumping and delivery']).
substitution([the_diaphragm],['the diaphragm']).
substitution([the_one_way_valves],['the one way valves']).
substitution([fuel_tank],['the fuel tank']).
substitution([fuel_pump],['the fuel pump']).
substitution([fuel_pipe],['the fuel pipe']).
substitution([the_pressure_in_pump_chamber],['the pressure in pump chamber']).
substitution([the_pump_chamber],['the pump chamber']).
substitution([draws_in],['draws in']).
substitution([the_fuel_via_one_way_valve],['the fuel via one way valve']).
substitution([the_volume_of_fuel],['the volume of fuel']).
substitution([is_decreased],['is decreased']).
substitution([low_by_distortion_of_the_diaphragm],['low by distortion of the
diaphragm']).
substitution([is_increased],['is increased']).
substitution([increases],[increases]).
substitution([low],[low]).
substitution([high],[high]).
substitution([the_pressure],['the pressure']).
substitution([in_the_pump_chamber],['in the pump chamber']).
substitution([direct],[direct]).
substitution([X],[X]) :-   word(X).
substitution([[pushes,direct]],['the supply and directing']).
substitution([must_be_working],[working]).



/* Defining the graphical representation of the fuel_system */

show_part2 :-   fill([500,500,1000,500,1000,1000,500,1000,500,500]).

show_part :-   fill([1100,16600,5000,16600,5000,16800,1100,16800,1100,16600]).

graphs :-
gdev(11),   fill(2,1),   fill([100,4000,30000,4000,30000,4800,100,4800,100,4000]),
fill(0,1),   fill([1100,4800,12100,4800,12100,12800,1100,12800,1100,4800]),
fill([24100,4800,27100,4800,27100,7800,24100,7800,24100,4800]),
fill([26500,7800,27000,7800,27000,11000,26500,11000,26500,7800]),
fill(1,1),   fill([16100,4800,16500,4800,16500,18800,16100,18800,16100,4800]),
line(1,7,1),
line([13100,6000,23600,6000]),   line([13100,6000,13100,7250,10500,7250]),
line([23600,6000,23600,10000,25600,10000,25600,7800]),
line([4750,7250,6000,7250]),   line([6500,7250,9000,7250]),   fill(1,1),
fill([6000,6750,6500,6750,6500,7500,6000,7500,6000,6750]),
line([4750,7250,4750,10000]),
fill(2,1),   fill([9000,6500,10500,6500,10500,8000,9000,8000,9000,6500]),   fill(3,1),
```

```
fill([4000,10000,5500,10000,5500,11500,4000,11500,4000,10000]),
fill(3,1),
line([4750,12000,3000,16000]),
line([6250,8000,8000,14000]),
line([9750,8500,11750,15000]),
line([14000,7000,19500,11000]),  line([26750,11500,26000,12000]),
text([25000,6000],'TANK'),
text([1100,17000],'CARBURETTOR'),
text([11100,16000],'PUMP'),
text([17000,12000],'FLEXIBLE HOSE'),
text([25000,13000],'FILLER'),
text([7000,15000],'FILTER'),
fill(1,1),
show_part,
show_part2,
get0('TRM:',Key),
gdev(0).
attempt(not,yet).

/* Various messages displayed to the user during system functioning */

escape_msg :-
cuwind(tutorial),
crwind(escape,20,0,3,35),
write('If you wish to leave this tutorial'),
nl,
write(' press CTRL <-'),
cuwind(tutorial).


infomiss :-  write('I am afraid I do not understand your reply'),
nl,
write('please try and answer the question again').
valdef_part(Ans,_A) :-
(Ans = 1 ,
user_select_mode(_A,C) ;
Ans = 2 ,
user_select_mode(fuel_pump,C)).
msg(Message) :-
cuwind(tutorial),
crwind(continue,20,40,3,30),
nl,
write('Press any key to continue'),
get0('TRM:',Key),
cuwind(tutorial),
close(continue).

user_model(none,as,yet).

msg3 :-  cuwind(tutorial),
crwind(graphics,20,20,3,40),
nl,
write('Would you like to see a picture of this?'),
nl,
 'R'(_Ans),
```

```
(_Ans = yes ,
close(escape) ,
graphs ,
cuwind(tutorial) ,
close(graphics) ,
escape_msg ;
cuwind(tutorial) ,
close(graphics)).


end_message :-
nl,   tab(3),
write('I am afraid I do not know what the problem is with'),
nl,   tab(3),
write('you car - try taking it to a car mechanic!'),
nl,   nl,   tab(20),
write('THANK YOU FOR USING THE MEDUD PROGRAM '),
nl,   nl,
abort.

end_message2 :-
nl,
write('THIS IS THE END OF THE MEDUD SYSTEM'),
nl,
write('THANK YOU AND GOODBYE'),
nl,   nl,
abort.
```

APPENDIX B

LIST OF PUBLICATIONS:

1. Cox,B., Jenkins, J.O., & Pollitzer, E. Explanation-Driven Understanding-Directed Approach to Knowledge Transfer, presented and published in the proceedings of the Alvey IKBS Research Workshop on Tutoring Systems, November, 1987

2. Cox,B., Jenkins, J.O., & Pollitzer, E. Understanding and Concept Acquisition in Adaptive Intelligent Tutoring Systems in Proceedings of the Fifth International Conference on Technology and Education, Edinburgh, March, 1988

3. Cox,B., Jenkins, J.O., & Pollitzer, E. An Organisation of Domain Knowledge for Tutoring Systems, Proceedings of Third International Symposium on Knowledge Engineering, Madrid, Spain, 1988

4. Cox,B., Jenkins, J.O., & Pollitzer, E. An explanation-driven, Understanding-Directed User Model for Intelligent Tutoring Systems, presented and published in IEE Colloquium on Intelligent Tutorial Systems, Digest No: 1988/89, May, 1988

5. Cox,B., Jenkins, J.O., & Pollitzer, E. Explaining and Understanding Engineering Problems - An Intelligent Tutoring Approach, presented at Third international Conference on Applications of Artificial intelligence in Engineering, Stanford, California, 1988
(Also published in 6).

6. Cox,B., Jenkins, J.O., & Pollitzer, E. Explaining and Understanding Engineering Problems - An Intelligent Tutoring Approach, published in Artificial Intelligence in Engineering: Diagnosis and Learning, Ed. J.S. Gero, Elsevier with Computational Mechanics Publications, 1988

7. Cox, B. Reconciling Problem-Solving and Instruction in One Knowledge Decomposition Framework. To be presented at Expert

Systems89 and published in a book by Cambridge University Press.

SEMINARS GIVEN:

The Tutor in your Computer
presented at Imperial College of Science & Technology, 1988

The Tutor in your Computer
presented at the City University, 1989

APPENDIX C

GLOSSARY

Articulate Expert System. An expert system that contains human like representation of knowledge which is capable of explaining its reasoning.

Authoring System. A domain-independent component of an ITS that allows the developer to enter specific domain knowledge into the tutor's knowledge base.

Bugs. Student misconceptions in declarative or procedural knowledge.

Bug Catalog. A set of well-analyzed and carefully collected patterns of typical errors

Coach. A form of student modelling in which the ITS intervenes only when it is fairly sure the student is doing something wrong. The intervention is with graduated hints and examples.

Cognitive Diagnosis. Description of specific mental processes occurring in a particular individual with respect to a particular task.

Cognitive Model. A representation of human cognitive processes in a particular domain.

Cognitive Structure. An internal mental representation of an external fact or phenomenon.

Computer Based Instruction (CBI). The use of computers for instruction and training. Generally this refers to instruction in which no expert system or production rules are used to order the sequence of information presented. It often results in linear sequences, or chains of presented material.

Constructivism. A pedagogical philosophy that views learning as constructing knowledge, rather than absorbing it.

Declarative Knowledge. A form of knowledge representation distinct from Procedural knowledge (although this distinction is not always useful) in which the knowledge is portrayed as static and structural; for example, data structures, frames, productions and semantic nets.

Discourse Strategy. The method used to present instructional material during the course of interaction.

Dynamic Systems. Complex mechanisms that require swift and effective interaction, so that instruction and tutoring must be terse and to the point, and more lengthy instruction delayed to a later debriefing.

Epistemology. Theory of the method or grounds of knowledge.

Expert System. A computer program that uses a knowledge base and inference procedures to act as an expert in a specific domain. It is able to reach conclusions very similar to those reached by a human expert.

Heuristics. Rules of thumb that are practical and often work, but are not based on a principles, theoretical understanding and therefore are not guaranteed to work.

Instructional Strategy. A general approach toward teaching or training, including objectives, plans and teaching style.

Intelligent Tutoring System(ITS). A computer program that tries to individualise instruction by creating a computer-based learning environment that acts as a good teacher, correcting mistakes, offering advice, suggesting new topics and sharing curriculum control. It should have the ability to analyse student responses, develop a history of the learner's preferences and skills and tailor the material to suit the trainee. Some important subtopics for ITS are knowledge representation, simulation, natural language, expert systems and induction.

ITS Architecture. A systematic approach to structuring the many components that comprise an effective, working ITS. Usually these consist of a student model, an organised domain of knowledge, instructional principles and a tutorial interface.

Knowledge Base. Codified knowledge (usually represented on a computer) of a domain or subject matter.

Knowledge Decomposition. The division of a body of target knowledge into smaller information-containing structures.

Knowledge Representation. Computer-based techniques for storing and retrieving knowledge organised according to specific principles. Prominent techniques include frames, semantic networks and object oriented techniques.

Mental Model. A popular theoretical construct for a knowledge representation form that supposes that people simulate their environments with models of the world that they are able to run in their minds. These runnable mental models can be used to predict the outcomes of thought experiments using novel conditions. Mental models can also be used to trace the causal connections of events and devices in the world.

Microworlds. Computer-based learning environments in which trainees are free to explore and discover the limits of their own understanding. The computer provides little direction or guidance, but it does narrow and constrain the topics for search to those that are valid within the current world. The environments can also raise sharply focused contrasts between alternative hypotheses about the world to facilitate insight and discovery.

Misconception. An item of knowledge that the student has and the expert does not have. A type of student-expert difference. A bug.

Mixed Initiative Tutors. An ITS that accepts and responds in natural language to both solicited and unsolicited natural language input from the user.

Overlay Models. Student modelling technique in which trainee performance is measured against the standard of an expert's model.

Pedagogics. The science of teaching.

Pragmatic Rules of Inference. Abstract, domain-independent inferential rules of reasoning.

Procedural Knowledge. A form of knowledge representation distinct from Declarative knowledge (although the distinction is not always useful) in which the knowledge is portrayed as active and functional, for example, functions, objects, demons and algorithms.

Production Rule. A rule of the form condition(s) imply action(s) used in modelling cognitive behaviour. A set of production rules and an interpreter for processing them is terms a production system.

Qualitative Models. A computer-based simulation composed of ordinal or even nominal metrics, such as 'good' and 'better' rather than higher order mathematical models.

Repair mechanisms. A unit of behaviour found in a general purpose reasoning procedure.

Repair Theory. A generative theory of bugs, a method of deriving bug libraries directly from correct procedures, reducing the need to collect bugs through empirical observation.

Semantic Networks. A graph structure that links concepts with conventional links such as 'part-of', 'isa'. Often seen as a Declarative form of knowledge.

Shell. A framework which provides procedures for the insertion of application-specific information into a computer program.

Student Model. (See User Model).

Tutoring Module. The component of an ITS that selects and orders the material to be presented to the student.

User Model. The component of an ITS that is used to make inferences about a trainee's stage of knowledge. Various student modelling systems have been proposed: bug catalogs, overlay models, issue oriented models and psychometric systems.