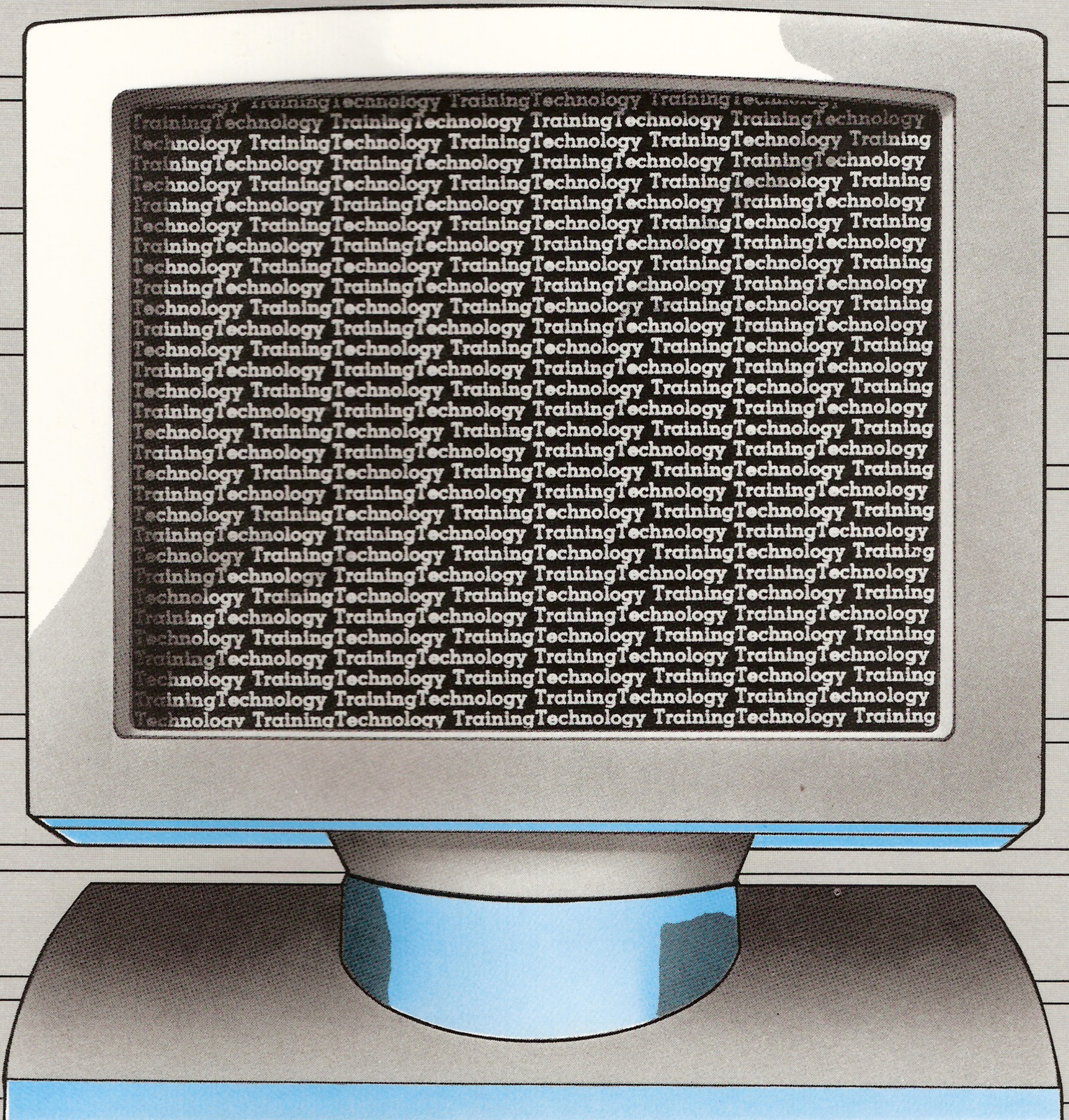
ARTIFICIAL INTELLIGENCE: APPLICATIONS TO TRAINING

A research study by the Open University

for the Training Technology Section of MSC

THE OPEN UNIVERSITY



CONTENTS

ACKNOWLEDGEMENTS

This report would have been impossible without the advice and comments of a number of my colleagues in the Open University, especially: Tony Hasemer of the Human Cognition Research Laboratory; Diana Laurillard, Tim O'Shea, Mark Elsom-Cook and Paul Lefrere of the Institute of Educational Technology; Phil Butcher of the Academic Computing Service; John Sparkes and John Monk of the Electronics Group; and Brian Bloomfield of the Systems Group.

Among those in industry who were kind enough to give of their time and experience, special thanks are due to: George Tsintas, Mike Dickens and John Garner at Lucas Open Learning; Derek Moore, Derek Eden and Peter le Blond at STC Idec Ltd.; Karl Wiig of Arthur D. Little Inc.; Kevin Doyle at Digital Equipment Corporation; Don Sherman and his colleagues at British Airways; Fiona Mirkowska of Frost and Sullivan; and Mike Turner of PA Computers and Telecommunications.

My thanks are also due to those who smoothed the administrative path to the execution of the contract, and in particular to Kitty Chisholm of the OU's Science and Technology Updating Programme, and to Keith Lindsey, Les Goodman and Neil Smith of the Manpower Services Commission.

Finally, it should be said that the Report owes a great deal to Ouida Rice who cheerfully and reliably tackled the mysteries of text-transfer, page make-up, graphics preparation and laser printing and in the process showed that when it comes to making computers behave according to their documentation, there is currently no artificial substitute for *real* intelligence.

John Naughton
Systems Group
The Open University
March 1986.

Executive Summary

Overview

This Report examines the implications of Artificial Intelligence (AI) research for industrial training over the course of the next five to ten years. To that end, it first reviews the nature, history, concerns, means and ends of AI research and identifies a number of central research areas which are potentially relevant to training. These areas are: **expert systems**, **knowledge representation and processing, natural language understanding** and **learning**. Each of these topics is then discussed in more detail before going on to consider the likely impact of AI-derived products on the training scene. The *general* impact of AI research on training is examined in terms of (i) the key factors which will affect the evolution of a market for such products, and (ii) the views of leading AI practitioners about the state of their art and of its likely development in the short- to medium term. Finally, the Report looks at Computer-Based Training (CBT) - the training technology most likely to be directly affected by AI research. It examines the limitations of conventional CBT systems and looks at the specific contributions which AI research could make towards overcoming these deficiencies.

Conclusions

(1) The main conclusion of the Report is that the AI revolution implied by 'Fifth Generation' research is still a long way off. Fundamental problems in such areas as : knowledge representation, elicitation, processing and communication; reasoning; machine learning; and language understanding present formidable obstacles to the development of *general*-purpose intelligent systems, and in the opinion of leading practitioners are unlikely to be overcome in the time-scale of this Report.

(2) A second conclusion is that significant advances *have* nevertheless been made in more specialised areas - for example, expert systems for particular applications, and natural-language front ends for database software. Almost all of these advances have come from the realisation that (i) *specialisation* rather than generalisation and (ii) *knowledge representation and processing* are the keys to more 'intelligent' software. The fruits of this research are now robust enough to be of direct use to conventional CBT, and it is argued that the Manpower Services Commission should henceforth support and encourage projects aimed at specific enhancements to CBT systems using AI techniques and approaches. Among the likely areas for such projects are:

* Enhancing the functional competence of CBT systems by imbedding expert systems within them.

* Improving the explanatory capabilities of CBT systems by incorporating more sophisticated methods of knowledge representation.

* Using AI approaches to construct more adaptive and 'intelligent' student models.

* Making the user interface of CBT systems more robust and less user-hostile by borrowing from AI experience with intelligent front-ends, language understanding, input checking, and so on.

* Adopting an AI-type approach to create sophisticated courseware development environments (hardware and software) for the authoring of CBT material.

* Supporting research and development into more powerful delivery vehicles (e.g. learners' workstations) for AI-enhanced CBT material.

(3) Thirdly, it is argued that, contrary to popular belief, there is much more to AI research than expert systems, and that these 'other areas' (knowledge representation, reasoning, planning, search, learning, language understanding, etc.) have at least as great a potential relevance for trainers as do expert systems. Conversely, merely imbedding expert systems in conventional CBT systems is not, in itself, sufficient to make significant enhancements to them.

(4) Fourthly, it is suggested that AI research may have useful insights and methods to offer conventional, non-computer-based, training. One way of looking at AI is to see it not so much as an academic discipline but as a distinctive *way of thinking* about computers, intelligence and cognition. Characteristics of the AI cast of mind are: a strong interest in the nature of knowledge, its structure, elicitation, representation and communication; a willingness to work with incomplete or conflicting information; an acceptance of uncertainty; and an interest in learning sharpened by experience of trying to make machines learn. Useful insights can follow from looking at training-related issues like skills auditing, course content and structure, tutoring styles, interpretation of student error and so on from an AI perspective. It is therefore recommended that the MSC should sponsor projects aimed at increasing trainers' appreciation of AI research.

INTRODUCTION

Background and Aims of the study, time horizons considered and structure of the Report

Background

The background to this study is the desire of the Manpower Services Commission (MSC) to commission an investigation of the implications of Artificial Intelligence (AI) research for industrial training. To that end, John Naughton, Senior Lecturer in Systems at the Open University, was contracted to conduct an urgent study over a six-month period ending March 31 1986, and to produce a report summarising his findings and making recommendations to the MSC about what initiatives, if any, it should next take in this area. This Report is the outcome of the project.

Aims

The main aim of the research was to provide a realistic assessment of the likely impact of AI on industrial training in the short- to medium-term future. A secondary aim was to identify areas in which the MSC might encourage AI-related projects and to make recommendations to the Commission about how it might generally support the diffusion of AI-related techniques, concepts and tools into the training field.

Understanding AI

Ever since the launch of the Japanese 'Fifth Generation' research programme, AI has been a fashionable topic in the West. It is difficult at the moment to open a newspaper or magazine aimed at the computing fraternity without encountering at least one item about AI and its related technologies. Much of this material is ill-informed and ludicrously abuses the term 'intelligence'. Some of it is fantastic in the literal sense, but mainly it is just the recycled hype of equipment and software vendors. Such coverage tends to be unduly sanguine about the achievements and prospects of AI research. And it tends to be unduly biased towards particular areas - notably so-called 'expert systems' - in which AI ideas are currently being commercially exploited, thereby giving the impression that the only things which matter in AI are those for which venture capital is currently forthcoming.

But there is more to AI than expert systems. In particular, there are grounds for saying that the most striking feature of AI is that it is more a state of mind - a particular approach to problem-solving and to using computers - than a rigorously-defined academic discipline. But however one defines it, the scope of AI research is vast, and the *potential* impact on society is correspondingly large. Likewise, the *potential* impact of AI on industrial training is enormous. But whether this potential will ever be realised in practice depends on a lot of things, and in particular on whether certain fundamental problems which currently dog researchers in the field are likely to be overcome within the relevant timescale. This implies that a realistic assessment of the likely impact of AI on training must start from a realistic appreciation of the state of the AI art at the time of writing. It also implies that a portion of this Report should be devoted to attempting a synoptic overview of the subject.

Time Horizons

The trouble with trying to predict the future, as Leonid Breznev used to say about free elections, is that one can never be sure of the outcome. In any area connected with computer technology, the problem is even more severe. Nevertheless, the time-horizon of this study is placed between five and ten years ahead.

This is not quite as foolhardy as it may sound, for the time lag between fundamental research in AI and its successful application in practical or commercial systems is at least as long (and in some cases longer) as that in other industries. Most of today's 'expert systems', for example, are based on fundamental research (in knowledge representation, logic programming, etc.) that was initiated and conducted in the early 1970s - i.e. 15 years ago. The increasing volume of funding and rising commercial pressures, together with advances in tools will, no doubt, reduce the time-lag between research and deployment. But it is nevertheless probably realistic to assume that the outline, if not the details, of the systems that will be widely available in the early 1990s can be predicted from what we know about the state of the AI art today.

The Impact(s) of AI

In essence, the conclusions we have reached are that AI will impact on industrial training in two ways - *indirectly* , through the impact of AI-related technologies on employment and the demand for skilled labour, and *directly* through the considerable impact that AI will have on computer-based training (CBT). Of these two types of effect, the first is difficult to quantify or even envisage with any precision: it depends too much on imponderables like the diffusion of new technology through the industrial system.

However, the second effect - the impact of AI on CBT - is something we can be more specific about. For it would seem that the whole of the CBT business is heading for a period of upheaval. The demands made on CBT are likely to increase very rapidly as organisations try to reap the benefits of standardised, high-throughput training. There will be increasing pressure to widen the scope and extend the range of CBT - to take it into areas where CBT has not hitherto been used.

These pressures, however, will serve initially to expose the current deficiencies of conventional CBT technology. Coincidentally, it also happens to be the case that some of the areas where conventional CBT stands most in need of enhancement - for example, better (more 'intelligent', friendly, robust) user-interfaces, knowledge representation, student modelling - are areas where AI research has made some progress and may have something worthwhile to contribute to training systems. A substantial proportion of the Report is therefore devoted to outlining what these contributions are, and how they might come about.

Structure of the Report

The overall structure of the Report follows from the above considerations. In Chapters 2,3 and 4 we examine AI as a research field. We look at its history, achievements, concerns, aspirations, failures and potential. Chapters 5 and 6 then build on this background to assess the impact which AI-derived technologies are likely to have on training generally and on computer-based training in particular. Finally, in Chapter 7 we present the conclusions reached as a result of the study, and an argued set of recommendations for consideration by the MSC.

Chapter Two

ARTIFICIAL INTELLIGENCE : AN OVERVIEW

The nature, concerns and history of Artificial Intelligence research.

What is Artificial Intelligence ?

A cynic might say that AI is whatever AI researchers choose to do, so wide is the range of activities which go on within the general umbrella of Artificial Intelligence.

Definitions

Here is how some experts in the field define their subject:

"Artificial intelligence is the science of making machines do things which would require intelligence if done by a human."

(Marvin Minsky, MIT)

"The goals of the field of artificial intelligence can be defined as attempting to make computers more useful and to understand the principles that make intelligence possible".

(Patrick Winston, MIT)

"The field of artificial intelligence has as its main tenet that there are indeed common processes that underlie thinking and perceiving, and furthermore that these processes can be understood and studied scientifically. In addition, it is completely unimportant to the theory of artificial intelligence *who* is doing the perceiving - man or computer. This is an implementation detail."

(Nils Nilsson, Stanford University)

"Artificial Intelligence research is that branch of computer science that investigates symbolic, non-algorithmic reasoning processes and the representation of symbolic knowledge for use in machine intelligence."

(Bruce G. Buchanan and Ed Feigenbaum, Stanford University)

"There are two quite different starting points to define AI - the dream and the technology. As a dream, there is a unified (if ill-defined) goal of duplicating human intelligence in its entirety. As a technology, there is a fairly coherent body of techniques... that distinguish the field from others in computer science."

(Terry Winograd)

"One of the world's deepest mysteries - the nature of mind - is at the centre of AI. It is our holy grail."

(Alan Newell)

A common theme running through these definitions is that AI is not just a discipline concerned with building particular types of machines but as an activity which aims to understand the nature of *human* intelligence through the construction of computer programs which exhibit intelligent behaviour. This means that a criticism which is often made of AI research - namely that it is 'just' another sort of computer programming - rather misses the point. Programs are the essential tool of AI workers, but they are means to an end rather than the end itself.

AI vs conventional programs

Moreover, there is an important distinction to be made between an 'intelligent' program and a conventional one. Both will generally be designed to solve some problem; but in the former, the program itself *generates* the *method* to be used in solving the problem calling on a range of *reasoning processes* that have been incorporated into it. In a conventional program, all the reasoning will have been done beforehand by the programmer; the program merely carries out the computations implied by this reasoning. Another way of approaching the distinction would be to say that conventional programming involves instructing the machine in every minute step to be performed whereas AI programs are designed so that during execution they are able to make many of these low-level decisions for themselves. The more 'intelligent' a program is, the more it is able to control its own behaviour in response to its inputs or to its environment. Conventional programs are designed to achieve one specific task; AI programs are or will increasingly be flexible and adaptive - i.e. more like human intelligence.

Because flexibility and adaptivity are desirable properties in many computing applications, what this implies is that we can expect a continuing tendency for conventional programming to import ideas originally developed within AI research. We are already seeing this, for example, in the migration of heuristic search techniques to microcomputer database packages like Ansa Corporation's *Paradox*, and pseudo 'natural-language' interfaces to other database products like Q&A. In this sense, the distinction between AI and non-AI programming is invariably one which shifts over time. As one software engineer put it: "AI is merely the computer science the rest of us cannot afford just yet".

The methodology of AI research

AI research generally proceeds along the following lines:

1. The researcher selects an activity which is generally acknowledged to require 'intelligent' behaviour. This could be anything from playing chess or other intellectual games to diagnosing faults in computer systems or managing patients in intensive care.

2. The researcher frames some hypotheses concerning the reasoning processes involved in the activity.

3. These reasoning processes are then incorporated in a computer program of the type discussed above.

4. The behaviour of the program is carefully observed and appropriate inferences made about the hypotheses framed in 2.

Combinatorial explosions

The problems dealt with by AI researchers tend to be dominated by what is called the "combinatorial explosion", which is really just a way of saying that they cannot be solved simply by going through an exhaustive review of all the opportunities and possibilities open to the problem-solver at any given time. This is because there are far too many such possibilities. If we take chess as an illustration, the average number of moves possible at any point in a chess game is 35. This means that a chess-playing program designed to look only three moves ahead would have to examine over 1.8 *billion* moves.

Knowledge prunes the search space

In technical terms, such a search space is so vast as to be 'unsearchable' by brute force methods. In general, the way AI programs deal with this is by using *knowledge* to reduce the number of options that must be examined. This knowledge generally takes two forms - *factual* and *heuristic.*

Thus in the chess-playing example, the program might use *factual* knowledge about the value of a piece, together with knowledge about the likely threat to it following each of the legal moves available, as a way of pruning the search space.

This factual knowledge could also be supplemented by *heuristics.* An example might be the rule "avoid moves which involve losing control of the centre of the board". Such rules may not be absolutely rigid of accurate – in fact they are usually provisional and tentative, which is why they are called 'heuristic', i.e. based on experience or involving trial and error. Searching generally involves using factual knowledge plus heuristics to reduce the computational problem inherent in deciding what to do next.

Searching is an essential part of much AI research, and takes many forms. One important type is *goal-directed search* where successive subgoals on the way to the main goal are computed as the search proceeds, using knowledge and heuristics to reduce as far as possible any chance of following a fruitless path. Generalising, one could say that the objective of any search program is to minimise the amount of fruitless searching it does on the way to its goal. But conversely, the amount of computation necessary to achieve that minimum may turn out to be more expensive than a bit of fruitless searching! What this illustrates is that there are few absolutes in AI programming: it is usually a matter of finding a trade-off between the costs of various approaches and their associated benefits.

Characteristics of AI programs

AI programs are characterised by some or all of the following features:

1. *Symbolic:* They deal mainly with non-numerical values. The value of a symbol is analogous to the meaning of an English word. Just as an English speaker manipulates words in order to generate meanings, so a computer program manipulates symbols in order to reach a conclusion of some sort. The difference is that in the process, the program may exhibit *behaviour* which is far more interesting (to the researcher, at any rate) than the conclusion itself. Symbolic computation involves the manipulation of symbols whose *values* are usually far more complex entities than mere numbers.

2. *Heuristic:* They attack problems for which no general solution algorithm is known. (An algorithm is a rule for solving a mathematical problem in a finite number of steps. Solving quadratic equations or differentiating expressions are examples of algorithmic processes.) Algorithms represent *deductive* processes along the lines of "All men are mortal; Socrates is a man; therefore Socrates is mortal". Because of the absolute statement begining "All ...", or its equivalent in programming terms, algorithms must produce the expected results under all circumstances. Heuristics, on the other hand, represent *inferential* processes such as "The sun has risen every day throughout recorded history, therefore it will rise again tomorrow". The latter is not necessarily a valid prediction.

Although AI programs use algorithms as much as possible, their creators are not deterred when an algorithmic solution does not exist. In such cases they tend to use 'heuristics' - i.e. informal methods, rules-of-thumb, etc. which are not guaranteed to succeed but which are thought to be useful on the basis of past experience. (Integration of mathematical expressions is a heuristic process.) Examples of real-world heuristic rules are:

"If you see an oily sheen on the water, the odds are that oil has been spilled."

or, at a more general level,

"If you have competing hypotheses, you can identify the best one by trying to rule them all out".

3. *Knowledge representation:* Because AI programs tend to use *knowledge* to reduce search tasks, they need some method of representing knowledge in a computable form. They cannot store it simply as strings of symbols - the usual form of storage in conventional programs - because such strings are not amenable to reasoning. Thus, storing the statement "John went to Milton Keynes" as a string will not enable a program to answer the question "Who went to Milton Keynes?". There are different ways of representing knowledge in AI, but all have the property that they make the stored knowledge available to inferential procedures. Among the requirements for a knowledge-representation method are that it should be able to express complex situations precisely and that it should enable the commonality of two apparently dissimilar things to be detected - e.g. say in being able to detect that 'porous limestone' and 'compacted limestone' are not two completely unconnected objects.

4. *Incomplete and/or conflicting data:* Unlike conventional programs, many AI programs can provide some solution(s) to a problem, even if some key data are missing, unavailable or contradictory.

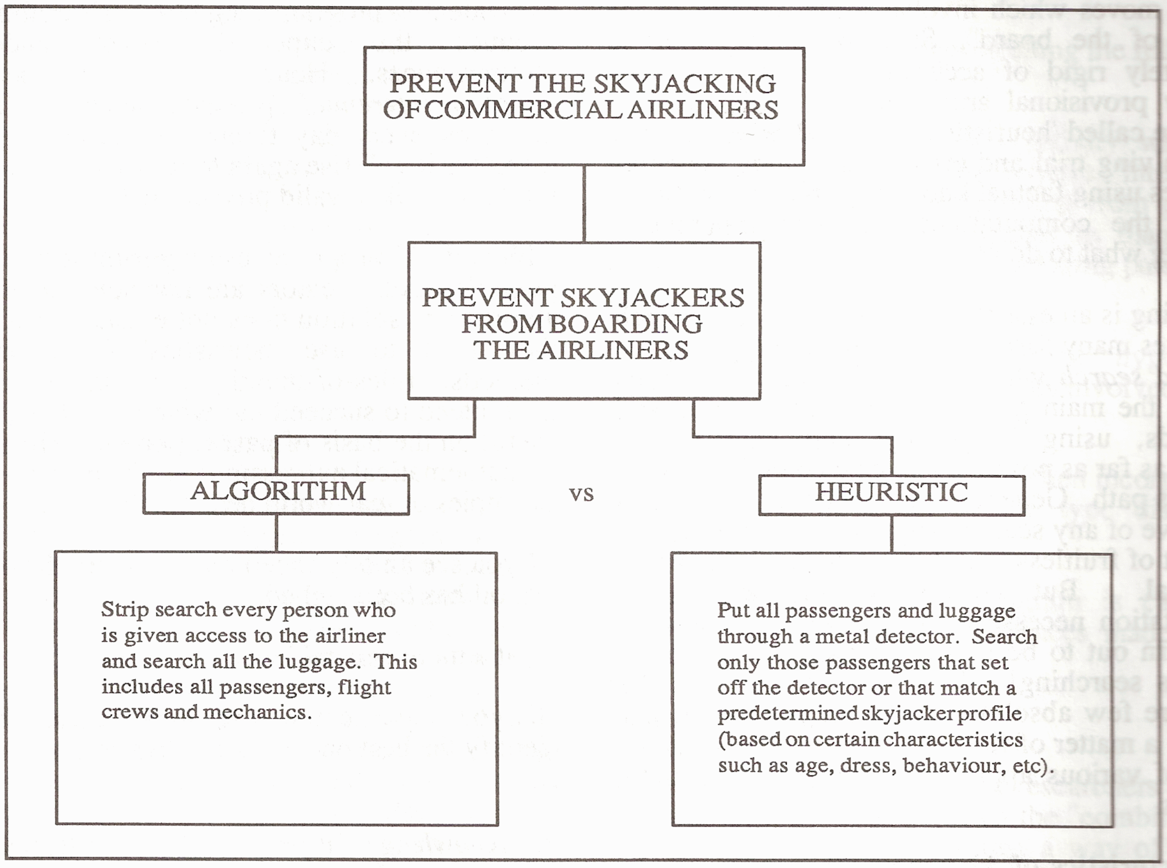


Figure 2.1 (from Waterman, page 17)

Illustration of difference between an algorithmic and heuristic approach to the problem of preventing skyjacking. The algorithm would certainly stop hijacking since it virtually guarantees that nobody could board a plane with a weapon. But it would be too time-consuming, expensive and unpopular to be of much practical value. The heuristic approach would stop most hijackings but could not guarantee that none would occur. The use of heuristic rules makes the search for solutions much easier and more practical.

5. *Learning:* Some AI programs have a (primitive) capacity to learn from experience through mechanisms like generalising, drawing analogies and selectively discarding information.

Concerns of AI research

We have seen earlier that AI is an activity which aims to understand the nature of intelligence through the construction of computer programs which exhibit intelligent behaviour. That implies that the goals or *ends* of AI research (understanding intelligence) should not be confused with the *means* towards those ends (computer programs). One way of surveying AI research is therefore to look at it in terms of the *ends* to which it is dedicated, the *means* which have been devised to achieve those ends, and the *products* which have emerged in the course of applying means to achieve ends.

Ends

Unpacking the general aim of understanding intelligence implies that major concerns of AI research are topics like:

* *reasoning* : how do we draw inferences, deduce conclusions from premises ? What rules do we use in doing so ?

* *language understanding:* how do we manage to utter and comprehend millions of unique sentences? How do we extract information from the incoherence and ambiguity of everyday speech? What's involved in building computer programs that can analyse natural language? Why do humans have no difficulty in distinguishing between the two sentences: *Time flies like an arrow* and *Fruit flies like a banana* while a machine cannot?

* *vision:* how do we make sense of what our eyes tell us ? Innumerable psychological experiments tell us that seeing is an active process, in which the brain continually interprets visual signals. We routinely and effortlessly manage to 'see' in 3-D and to identify moving objects, complex shapes etc. with great accuracy under a wide variety of lighting conditions. Why is it proving so difficult to build machines which can do the same ?

* *knowledge representation:* how do humans hold and access their knowledge? And what is 'knowledge' anyway ? What kinds of 'containers' are available for enabling programs to store and access knowledge (as distinct from data) ?

* *learning:* what is 'learning'? And how do humans learn? What's involved in building computer programs that can learn ?

* *planning and problem-solving:* A characteristic feature of purposeful human activity systems is *planning*. Much attention has been paid by AI researchers to devising programs that can plan, but these have mainly been confined to artificial micro-worlds' of regularly-shaped blocks. Nevertheless, it is clear that if we are ever to have, for example, sophisticated robots capable of dealing with a changing environment, then they must be able to plan their work to a considerable extent.

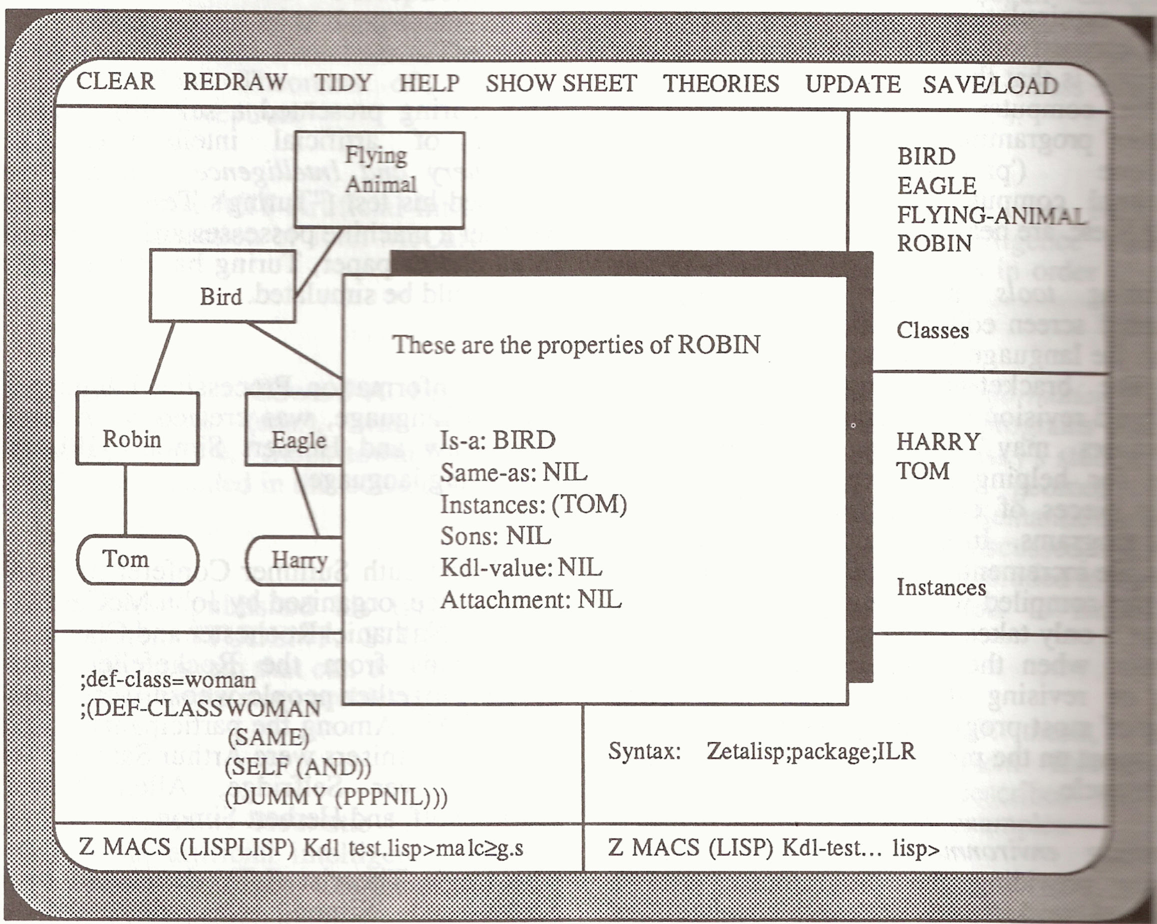


Figure 2.2

Illustration of a Symbolics screen

Means

Since AI research is almost invariably conducted by writing computer programs, it's not surprising that among the *means* developed to help achieve the above ends, programming languages, tools and what are called 'environments' (see below) tend to be pre-eminent.

As far as *languages* are concerned, the most commonly-used AI languages are LISP, PROLOG and POP-11. LISP is by far the oldest (it was invented by John MacCarthy and predates FORTRAN) and is still the dominant language of AI research, especially in the US. PROLOG is a European invention, and was adopted for the Japanese Fifth Generation project. POP-11 is a derivative of earlier POP languages originated and developed at the Universities of Edinburgh and Sussex.

A characteristic feature of all AI languages is that they are oriented towards symbolic rather than numerical manipulation. The name LISP, for example, comes from 'LISt Processing'. Another characteristic is that they are more geared towards telling the computer *what* has to be done ('declarative' programming) rather than *how* it is to be done ('procedural' programming). Conventional computer languages like Pascal, FORTRAN etc. are heavily procedural.

Programming *tools* include such things as sophisticated screen editors which 'understand' the syntax of the language being used and can handle chores like bracket-balancing, indenting and layout, rapid revision of program text, and so on. Such editors may include smart 'browsing' facilities for helping the programmer to find particular pieces of code in large and complex suites of programs. In the tools category also are facilities like incremental compilation, in which the code is part compiled while being edited, with the result that it only takes a few seconds to complete compilation when the programmer has finished creating or revising the program. The basic objective of most programming tools is to reduce the time spent on the program-compile-run-revise-recompile cycle

Programming *environments* take this a stage further by providing the programmer with a sophisticated working environment consisting of intelligent software (including the kinds of tools mentioned above) and powerful hardware (fast processors, large amounts of memory and backing store, a large, high-resolution screen and excellent graphics with windowing).

The history of AI

AI is a comparatively new subject which dates mainly from the 1950s. Some histories of the subject locate its origins in a celebrated conference held in the summer of 1956 at Dartmouth College, New Hampshire which included four people who subsequently became the dominant researchers in the field - Marvin Minsky, John McCarthy, Herbert Simon and Allen Newell.

The following chronology, adapted from Scown (1985), provides a trace of important events in the history of the subject to date. A popular history of the early years is given in McCorduck (1979).

1950
Alan Turing presented a scientific paper on the subject of artificial intelligence, *Computing Machinery and Intelligence*. In this paper, he proposed his test ("Turing's Test") for determining whether a machine possesses artificial intelligence. In an earlier paper, Turing had suggested that the brain could be simulated.

1955
IPL-II (Information Processing Language-II), the first AI language, was created by Allen Newell, J.C. Shaw and Herbert Simon. IPL is a list-processing language.

1956
The Dartmouth Summer Conference on Artificial Intelligence, organised by John McCarthy, Marvin Minsky, Nathaniel Rochester and Claude Shannon, with funds from the Rockefeller Foundation, brought together people whose work founded the field of AI. Among the participants in addition to the four organisers were Arthur Samuel, Trenchard More, Oliver Selfridge, Allen Newell, Ray Solomonoff, and Herbert Simon.

The Logic Theorist (LT), developed by Newell, Shaw and Simon, was discussed at this conference. The LT, considered the first AI program, used heuristic search to solve problems in Whitehead and Russell's *Principia Mathematica*.

Mid-late 1950s
John McCarthy, then at MIT, designed the LISP (List Processing) language.

1957
Newell, Shaw, and Simon began developing the General Problem Solver (GPS). This program applied codified problem-solving techniques, including means-ends analysis, to problems in a number of different task environments.

1959
Culminating years of experimentation, Arthur Samuel completed a checkers-playing computer program that performed as well as some of the highest-rated players of that time. His paper, entitled "Some Studies in Machine Learning Using the Game of Checkers" was published in the *IBM Journal of Research and Development* .

1959
Frank Rosenblatt's paper describing his pattern-recognition machine, the Perceptron, was published in *Proceedings of a Symposium on the Mechanization of Thought Processes*. The paper was entitled " Two Theorems of Statistical Separability in the Perceptron."

1960
Research began in the MIT Artificial Intelligence Project under the direction of John McCarthy and Marvin Minsky.

1963
Computers and Thought, Edward A. Feigenbaum and Julian Feldman, (eds.), was published. Marvin Minsky's article, "Steps toward Artificial Intelligence", was included in this collection.

1964
Daniel G. Bobrow published his Ph.D. thesis, based on his system STUDENT. STUDENT is a natural language program that can understand and solve high school algebra story problems.

1965
The Stanford University Heuristic Programming Project (HPP), an artificial intelligence research laboratory within Stanford's Computer Science Department, began research in expert systems. The HPP is now part of Stanford's Knowledge Systems Laboratory. Edward A. Feigenbaum is currently the principal investigator in the HPP.

1965
Work began on DENDRAL, the first expert system. Developed at Stanford University by a group including Joshua Lederberg, Edward A. Feigenbaum, Bruce G. Buchanan, Dennis Smith and Carl Djerassi, DENDRAL analyses information about chemical compounds to determine their structures.

1966
"ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine" was published in *Communications of the Association for Computing Machines*. Joseph Weizenbaum created ELIZA to illustrate that natural-language capabilities can make a computer seem deceptively intelligent. ELIZA was a psychology program that simulated the responses of a therapist in interactive dialogue with a "patient".

1966
Richard D. Greenblatt began developing a computer chess game capable of competing successfully in tournaments. This system was described in "The Greenblatt Chess Program", in *AFIPS Conference Proceedings*.

1966-1972
SHAKEY, a mobile robot, was built at SRI International. Shakey's "intelligence" allowed it to perceive and to plan actions in order to carry out tasks.

1968
Marvin Minsky's *Semantic Information Processing* was published. One of the programs described in the book, developed by Minsky's student Thomas G. Evans, could answer geometry analogy questions from an IQ test. "Semantic Memory" by M. Ross Quillian, which discussed his semantic network concept, was also included in the volume. Quillian used semantic nets to model human associative memory.

1970
Patrick H. Winston's Ph.D. thesis, *Learning Structural Descriptions from Examples*, was published. The thesis describes ARCHES, a program that learned from examples.

1970
MIT's Artificial Intelligence Project became MIT's Artificial Intelligence Laboratory under the direction of Marvin Minsky and Seymour Papert. The laboratory has been under the direction of

Mid-late 1950s
John McCarthy, then at MIT, designed the LISP (List Processing) language.

1957
Newell, Shaw, and Simon began developing the General Problem Solver (GPS). This program applied codified problem-solving techniques, including means-ends analysis, to problems in a number of different task environments.

1959
Culminating years of experimentation, Arthur Samuel completed a checkers-playing computer program that performed as well as some of the highest-rated players of that time. His paper, entitled "Some Studies in Machine Learning Using the Game of Checkers" was published in the *IBM Journal of Research and Development* .

1959
Frank Rosenblatt's paper describing his pattern-recognition machine, the Perceptron, was published in *Proceedings of a Symposium on the Mechanization of Thought Processes*. The paper was entitled " Two Theorems of Statistical Separability in the Perceptron."

1960
Research began in the MIT Artificial Intelligence Project under the direction of John McCarthy and Marvin Minsky.

1963
Computers and Thought, Edward A. Feigenbaum and Julian Feldman, (eds.), was published. Marvin Minsky's article, "Steps toward Artificial Intelligence", was included in this collection.

1964
Daniel G. Bobrow published his Ph.D. thesis, based on his system STUDENT. STUDENT is a natural language program that can understand and solve high school algebra story problems.

1965
The Stanford University Heuristic Programming Project (HPP), an artificial intelligence research laboratory within Stanford's Computer Science Department, began research in expert systems. The HPP is now part of Stanford's Knowledge Systems Laboratory. Edward A. Feigenbaum is currently the principal investigator in the HPP.

1965
Work began on DENDRAL, the first expert system. Developed at Stanford University by a group including Joshua Lederberg, Edward A. Feigenbaum, Bruce G. Buchanan, Dennis Smith and Carl Djerassi, DENDRAL analyses information about chemical compounds to determine their structures.

1966
"ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine" was published in *Communications of the Association for Computing Machines*. Joseph Weizenbaum created ELIZA to illustrate that natural-language capabilities can make a computer seem deceptively intelligent. ELIZA was a psychology program that simulated the responses of a therapist in interactive dialogue with a "patient".

1966
Richard D. Greenblatt began developing a computer chess game capable of competing successfully in tournaments,. This system was described in "The Greenblatt Chess Program", in *AFIPS Conference Proceedings.*

1966-1972
SHAKEY, a mobile robot, was built at SRI International. Shakey's "intelligence" allowed it to perceive and to plan actions in order to carry out tasks.

1968
Marvin Minsky's *Semantic Information Processing* was published. One of the programs described in the book, developed by Minsky's student Thomas G. Evans, could answer geometry analogy questions from an IQ test. "Semantic Memory" by M. Ross Quillian, which discussed his semantic network concept, was also included in the volume. Quillian used semantic nets to model human associative memory.

1970
Patrick H. Winston's Ph.D. thesis, *Learning Structural Descriptions from Examples,* was published. The thesis describes ARCHES, a program that learned from examples.

1970
MIT's Artificial Intelligence Project became MIT's Artificial Intelligence Laboratory under the direction of Marvin Minsky and Seymour Papert. The laboratory has been under the direction of

Patrick H. Winston since 1973. Earlier work on artificial intelligence at MIT led to the first basic tools for word processing and the concept of time-sharing computers. Current research at MIT includes computer vision, all areas of robotics, expert systems, learning and common-sense reasoning, natural language, and computer architectures.

1970
Jack D. Myers and Harry E. Pople began work at the University of Pittsburgh on INTERNIST, now call CADUCEUS, a system intended to aid physicians in the diagnosis of human diseases.

1970
Alain Colmerauer and his colleagues began developing the Prolog programming language. Prolog development has also been active in Edinburgh, London and Budapest.

1970
Terry Winograd, then at MIT, wrote for his Ph.D. thesis SHRDLU, a natural language-understanding program that could respond to questions and plan actions in a simplified "blocks world". The thesis was later published as *Understanding Natural Language*.

1971
Nils Nilsson and Richard Fikes completed work on STRIPS at SRI International. STRIPS made use of plans, sequences of operators, to achieve goals.

1971
MACSYMA was first used. MACSYMA was developed over more than a decade at MIT by William Martin and Joel Moses. MACSYMA'S design was based on prior work by Martin, Moses and Carl Engleman. MACSYMA performs differential and integral calculus and simplifies symbolic expressions. Both inputs and outputs are symbolic and the program is knowledge-based. This program is widely used by mathematicians, research physicists and engineers.

1971-1976
The United States Defense Advanced Research Projects Agency (DARPA) sponsored research into connected-speech understanding capability in the Speech Understanding Research (SUR) Program. Some of the resulting programs were SPEECHLIS and HWIM (Hear What I Mean), from Bolt Beranek and Newman Inc., and HEARSAY-I, HEARSAY-II, DRAGON and HARPY from Carnegie-Mellon University.

1972
William Woods et al., at Bolt Beranek and Newman, developed LUNAR, an information retrieval system that uses augmented transition networks (ATNs) as the representation form for its natural language system grammar. LUNAR was intended for use by geologists in the evaluation of materials obtained from the moon during the *Apollo-11* mission. Woods had developed the ATN concept earlier in a 1970 paper on the subject.

1973
SUMEX-AIM (Stanford University Medical Experimental Computer Project - Artificial Intelligence in Medicine) was formed as a community resource for the development of AI techniques with support from the National Institutes of Health. SUMEX-AIM has been the source of MOLGEN and other projects in medecine, biochemistry and psychology.

1973
Roger C. Schank's "Conceptual Dependency: A Theory of Natural Language Understanding" was published in *Cognitive Psychology*. Schank et al., at the Stanford University AI Laboratory, later used the conceptual dependency knowledge representation in MARGIE, a natural language understanding program that could make inferences and generate paraphrases.

1973
Computer Models of Thought and Language, Roger C. Schank and Kenneth M. Colby (eds.), was published.

1973
Pattern Classification and Scene Analysis, by R.O. Duda and P.E. Hart was published.

mid-1970s
The initial version of MYCIN, an expert system that makes recommendations for the treatment of meningitis and other bacterial infections in the blood, was developed within SUMEX-AIM by Edward H. Shortcliffe. MYCIN's medical knowledge is encoded as production rules.

1975
DARPA began the Image Understanding Program to sponsor research into machine vision, including developing a theory of vision and hardware for image processing. ACRONYM, a model-driven interpretation system, was developed under this program by R.A. Brooks.

1975
The Psychology of Computer Vision, (Patrick Winston, ed.) was published. Marvin Minsky's paper, "A Framework for Representing Knowledge," was included in this collection. Minsky's paper discussed frames as useful structures for organising knowledge in many systems including natural language and vision systems. The collection also included "Understanding Line Drawings of Scenes with Shadows", by David Waltz. This paper discussed a new way to use the edges of shadows to interpret visual images.

1975
Roger C. Schank and Robert Abelson et al., at Yale, published a paper describing SAM (Script Applier Mechanism), a natural language understanding program that added the use of scripts to conceptual dependency representations.

1975
Representation and Understanding, Daniel G. Bobrow and Allan Collins (eds.), was published. This volume included important papers on knowledge representation.

1976
Douglas B. Lenat wrote AM, a type of learning program that defines and evaluates mathematical concepts in set and number theory. This process has been described as "automated discovery".

1976
Randall Davis published his thesis for a Ph.D. at Stanford University on TEIRESIAS, a system that utilises metalevel knowledge to enter and update knowledge bases used in expert systems. The thesis, published as a Stanford AI Memo, was later published in *Knowledge-Based Systems in Artificial Intelligence,* Randall Davis and Douglas B. Lenat, joint authors.

1977
Programmers at Hungary's Institute for Computer Coordination (SZKI), in Budapest, completed the first of many practical expert system applications utilising the Prolog language.

1978
R.O. Duda et al., at Stanford Research Institute International, published a paper discussing PROSPECTOR, an expert system that assists in the analysis of information related to geological exploration.

1980
XCON, the first expert system successfully used on a daily basis in a commercial environment, went into operation at Digital Equipment Corporation. The prototype for XCON was developed under the direction of John McDermott of Carnegie-Mellon University.

1981
The first volume of the three-volume set, *The Handbook of Artificial Intelligence,* Avron-Barr (ed.) Vols I and II, Edward A. Feigenbaum (ed.) Vols. I-III, and Paul R. Cohen (ed.) Vol. III was published. The other two volumes of the handbook were published the following year.

1981
Japan announced its intention to organise a Fifth-Generation Computer Systems Project.

1982
Japan's Institute for New Generation Computing Technology (ICOT) was formally launched at its Tokyo headquarters.

1982
The Microelectronics and Computer Technology Corporation (MCC) was formed in the United States to respond to the Japanese fifth-generation program.

1982
The United Kingdom began the Alvey Program of Advanced Information Technology to perform fifth-generation computer research.

1983
The European Economic Community formed ESPRIT to compete in the race to develop a fifth-generation computer.

1983
MCC opened for business in Austin, Texas.

1983
The Turing Institute opened at the University of Strathclyde in Edinburgh, Scotland, offering training in subjects related to machine intelligence.

Lessons of history?

The great paradox of AI is the way in which the expectations of its founders have been more or less contradicted by experience. What we seem to have discovered is that things which humans find difficult (e.g. integral calculus, algebra, theorem-proving, chess) can often be accomplished by machines with surprising ease, while activities which humans do effortlessly - and which are therefore taken for granted (for example, walking, talking, seeing, riding a bicycle) - appear to be very difficult for machines. Of these, the most important from a training point of view is the ability to understand natural language.

The history of AI research to date can be crudely summarised in terms of three main phases (see Figure 2.3).

In the 1960s, AI researchers tried to simulate the processes of thinking by finding general methods of solving broad classes of problem and embodying these methods in general-purpose programs. But this strategy provided no breakthroughs. It turned out that developing general-purpose programs was too difficult and was felt by many of those involved to be ultimately fruitless, for the more classes of problem a particular program could handle, the worse it seemed to perform on any individual problem.

Consequently, researchers sought other ways of attacking the problem of how to make computer programs 'intelligent'. They retreated from the search for general methods of problem solving and moved into research on much more specialised issues like *representation* and *search* (i.e. how to control the search for a solution so that it doesn't take too long or require more computing power than is readily available). This strategy was more successful than the earlier quest for generality; in particular, it led to the development of really useful techniques for representation and search. But again it provided no breakthroughs.

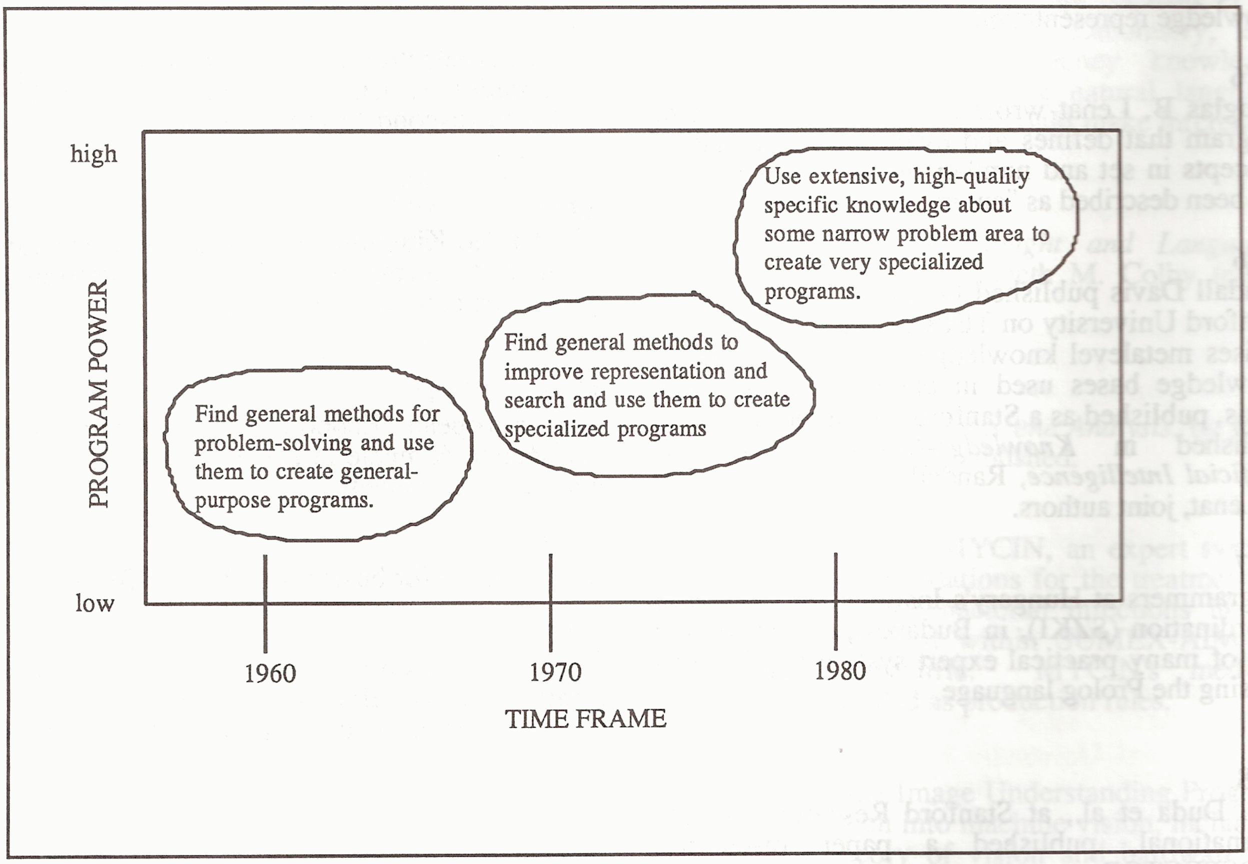


Figure 2.3 (Adapted from Waterman, 1986, page 4)

The shifting focus of AI research

"I think it might be more accurate to say that the failures of the early problem-solving machines highlighted how little was known about problem representation and search. Of the two, the latter has borne much fruit and is now so well known that it is thought of as virtually the province of Computer Science rather than of AI, whilst the former remains incompletely understood. The major question now, with the inexorable rise of expert systems, is how to represent knowledge. The brute force methods of data processing are inadequate to handle complex and often dynamically-changing 'chunks' of knowledge. The failure of computers to handle the combinatorial explosion is paralleled by their failure to make individual inferences from large knowledge bases at anything like the speed that we can."

Tony Hasemer

It wasn't until the middle- to late-1970s that AI people began to realise that *knowledge* rather than, say, inference might be the key. In other words, the way to make a computer program 'intelligent' might be to provide it with lots of high-quality specific knowledge about some problem area. This led to the proliferation of special-purpose programs embodying detailed knowledge of some rather specialised area. Such programs were called 'knowledge-based systems' (KBS) or 'intelligent' KBS (IKBS). Inevitably, because the knowledge some of them contained was very specialised and abstruse, they also came to be known as 'expert systems'.

'Expert Systems' have rapidly become the dominant vehicle for applied AI research, for several reasons:

(1) Because some of the early IKBS embodied very high-level knowledge (the program DENDRAL, for example, which deduces molecular structure from mass-spectroscopy and nuclear magnetic resonance data, was built using the knowledge of two Nobel laureates in chemistry), they were genuinely 'expert' in their fields of application. For many people, they thus constituted the first proof that AI research might have something useful to offer the outside world.

(2) Secondly, the knowledge-based approach was quickly recognised by industry as having considerable commercial potential, especially in those areas (like the financial and banking sectors) where professional *expertise* is effectively what adds value to the service provided, or where (as in the oil industry) scarce or expensive expertise is needed in geographically-dispersed locations.

(3) From the point of view of researchers, expert systems bring together many of the diverse strands in earlier AI research. They require, for example,

* methods for representing and accessing knowledge
* inference mechanisms,
* sophisticated user interfaces (ideally including natural language processing)
* sophisticated development environments.

Likewise, they pose, in a sharply practical context, intellectual problems that urgently require solutions (e.g. how to provide systems in which dialogue with the user is not entirely determined by the program).

(4) Also of relevance to the academic community is the fact that industrial interest in expert systems technology means relatively abundant funding for research.

The result of all this is that when people talk nowadays about 'applied AI' what they usually mean is expert systems research, development or application. This is understandable, but tends to undervalue the extent to which AI research is relevant to other areas like robotics, language, teaching and learning, planning and computing generally.

"There is no doubt, as far as I am concerned, that the development of expert systems is the major advance in the field during the last decade....The emergence of expert systems has transformed the enterprise of AI, not only because it has become the main driver of the current wave of commercialisation of AI, but because it has set the major scientific problems for AI for the next few years - namely, to assimilate expert systems into the general body of scientific knowledge in AI ..."

Alan Newell, 1984.

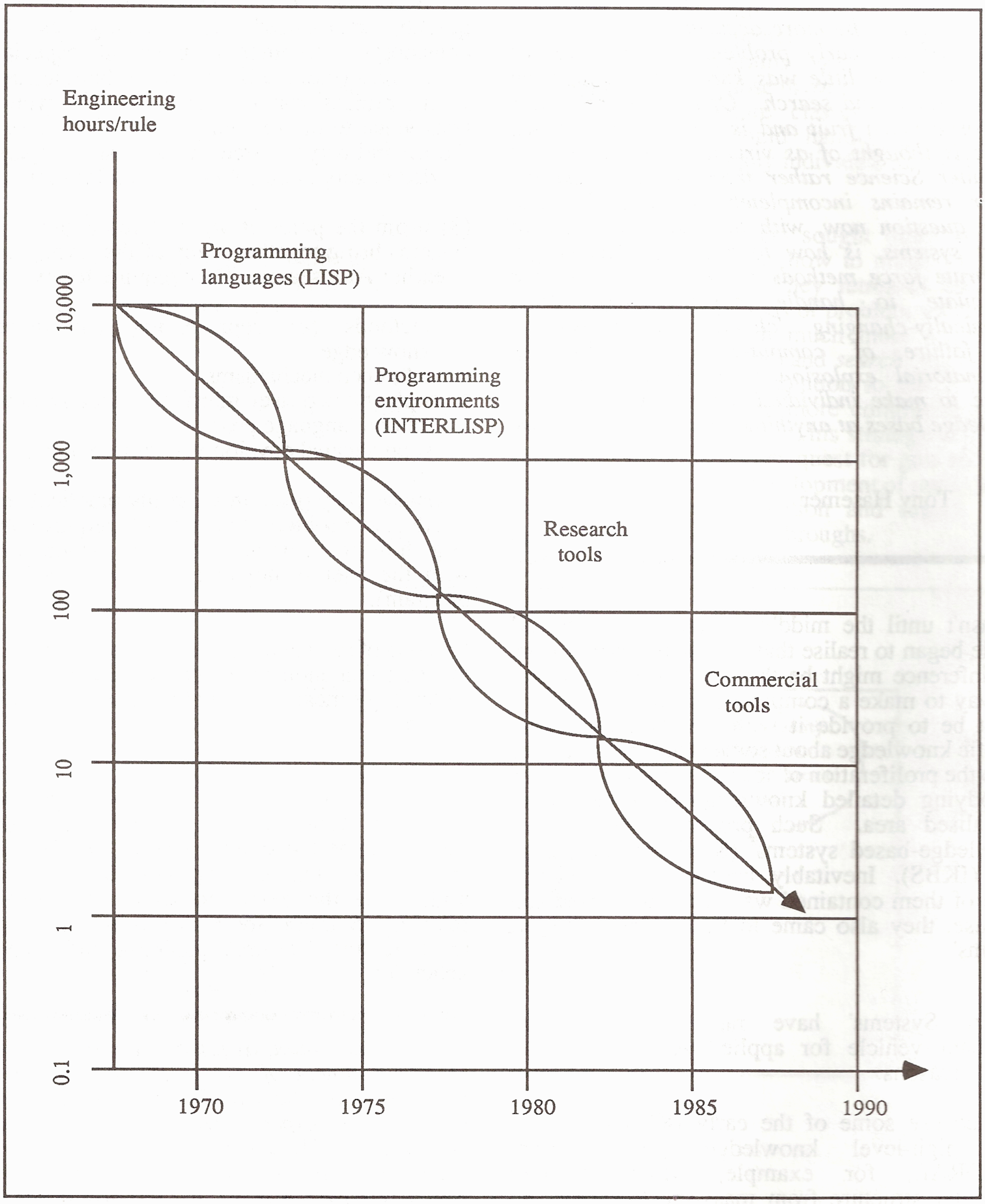


Figure 2.4 (from Harmon and King, page 9)

How languages, tools and environments combine to speed up the IKBS-building process.

Relevance to Training

Even from the above brief survey, it is clear that several areas of research in AI may be relevant to training. In particular:

Expert systems are potentially important because (i) they may affect the supply of, and demand for, certain specialist skills, and (ii) they offer a way of remedying one of the major deficiencies of conventional computer-based training (CBT) systems, namely the fact that they are generally not 'expert' in the subject(s) they purport to teach (see Chapter 6 below).

Natural language understanding is potentially important because of its potential effects on the demands for some skills (e.g. copy typing/secretarial) and also because of the way NL interfaces could improve CBT packages as well as conventional software.

Knowledge representation is important to conventional face-to-face trainers because training is ultimately an attempt to make explicit certain kinds of knowledge in order to transfer it to other people. Thus it may be that the insights of AI into knowledge representation might be useful to trainers.

AI-based research into human cognition may also be relevant to CBT because AI is concerned with constructing psychologically plausible models of human mental operations. If a CBT system is to have any 'understanding' of its interactions with its student/trainee, it needs to be able to create an internal dynamic model of the student and to embody a non-naive representation of what it is to 'teach' a subject effectively.

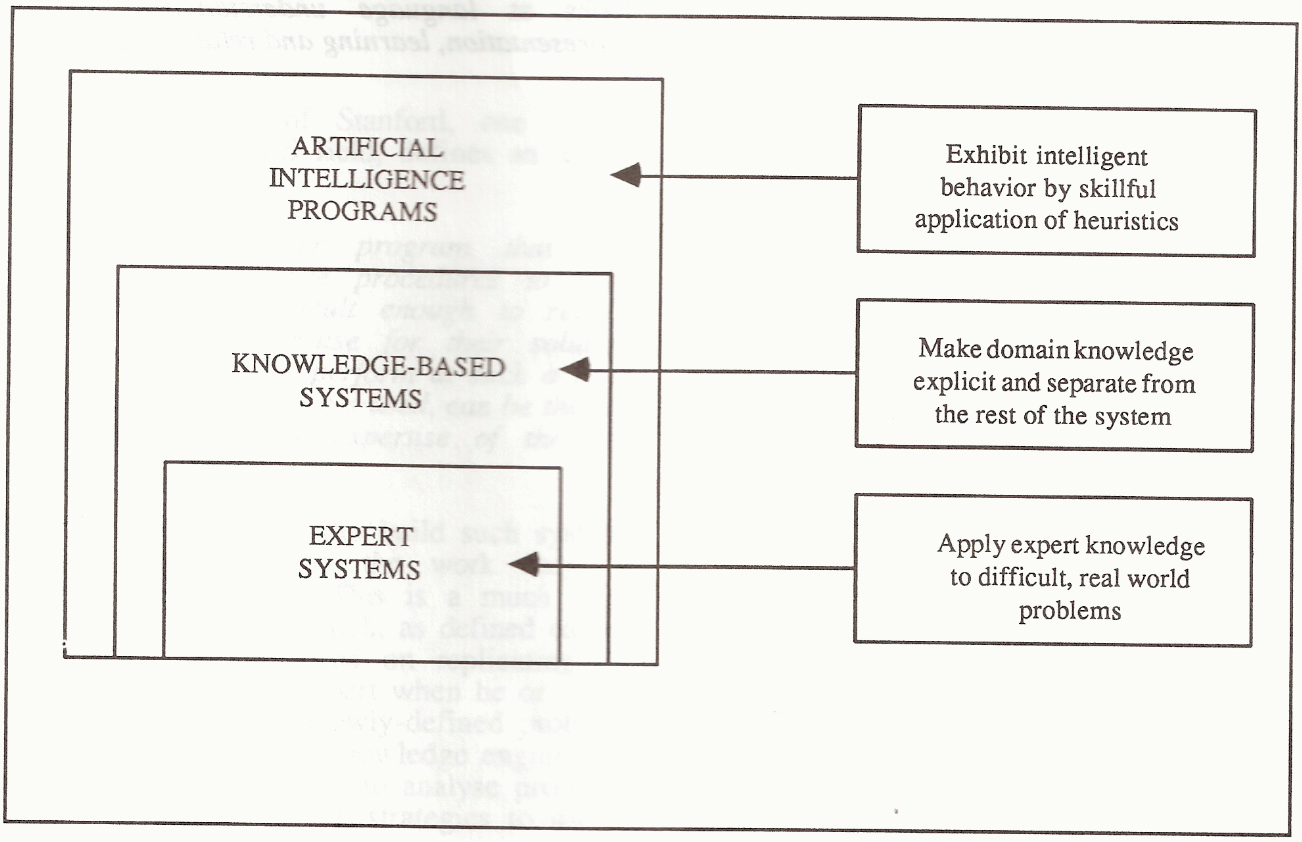


Figure 2.5 (from Waterman, page 18)

Expert systems in relation to other types of AI programs

For the same reasons, AI research into learning may be useful to teachers and trainers. The research has been conducted by embodying theories about learning in computer programs and studying their behaviour. In the process, significant insights into what is involved in learning have emerged. An example is provided by the work of Tim O'Shea, Richard Young and others who have used AI approaches in studying how children learn arithmetic. By building programs which can do arithmetic and then degrading the programs in systematic ways they were able to model the mistakes made by children and thus to pinpoint the crucial gaps in the students' knowledge or understanding.

Vision research is relevant - though in a less direct way - because sophisticated vision systems are a prerequisite for more sophisticated robotics, and deployment of such robotics systems will in turn impact on employment and skills.

Other potential contributions of AI include techniques for eliciting knowledge from an expert. Generally, this is done using tape-recorded protocols, but how should the researcher (the 'knowledge-engineer') set about extracting from the transcript the essence of expertise which s/he may not even understand ? Knowledge at this level is not comparable to a set of premises or statements; sub-sections of the knowledge are interconnected, related and contrasted in highly complex ways. AI is investigating these structures, and how to recognise and represent them. In doing so, it is implicitly tackling problems which also confront the trainer who is seeking to capture expert knowledge in a subject on which he himself is not an expert, for 'repackaging' for a particular audience of trainees.

Because of this wide range of potentially relevant research, it is important to assess the state of the art in the AI areas of most obvious relevance to training. Consequently, Chapter Three considers *expert systems* in more detail, and Chapter Four looks at *language understanding, knowledge representation, learning and related subjects.*

Chapter Three

EXPERT SYSTEMS

'Intelligent Knowledge-Based Systems' or 'Expert Systems' is currently the area of applied AI research which is receiving most attention and funding. Although such systems appear to have considerable promise, relatively few commercially-viable systems have emerged as yet, and the cost-effectiveness of the technology remains to be demonstrated over a range of application domains. Nevertheless, because expert systems represent a way of capturing high-level knowledge in software, they are of considerable interest to those concerned with improving conventional computer-based training systems.

Edward Feigenbaum of Stanford, one of the luminaries of the IKBS field, defines an 'expert system' as

an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. Knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of the expertise of the best practitioners of the field.

Feigenbaum calls those who build such systems *knowledge engineers* and the work they do *knowledge engineering*. This is a much more specific field than AI research, as defined earlier. Knowledge engineers focus on replicating the behaviour of a specific expert when he or she is engaged in solving a narrowly-defined problem. AI's main contribution to knowledge engineering lies in its insights into how to analyse problems and develop general search strategies to use in solving them.

Components of an expert system

Structurally, an expert system has three components (see Figure 3.1) - a knowledge base, an inference system (sometimes called an 'inference engine') and a user interface.

A key point to note is that the system contains a *knowledge* base, not a *data*base. The knowledge base of course includes a considerable amount of data and factual knowledge, but it also includes knowledge about problem analysis and solving in the form of heuristic rules (or other representations). Another way of putting it is that a database contains facts and/or statements taken to be true, whereas a knowledge base contains relationships between facts and/or statements.

Knowledge bases versus databases

To illustrate the difference between the two, consider a hospital doctor approaching a patient in bed. The database in this case is the clipboard file at the foot of the bed containing factual information about the patient's temperature, pulse, medication, etc. But the *knowledge* base is all the knowledge locked up in the doctor's head - factual knowledge picked up from textbooks, hints picked up from colleagues, rules of thumb derived from practical experience, trial-and-error, what he or she reads in journals, and so on.

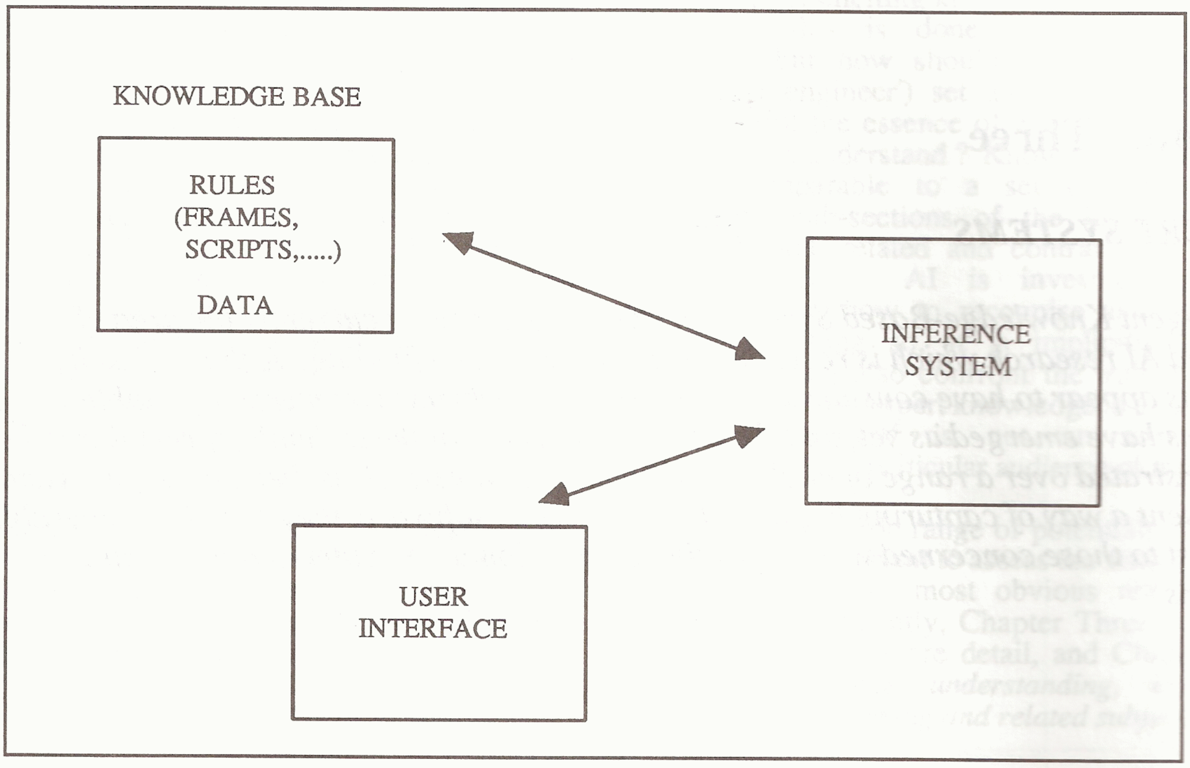


Figure 3.1

Precisely how expert knowledge should be 'represented' in a computer program is a lively research topic in AI, and various methods have been proposed - notably rules, semantic nets, object-attribute-value triplets, frames and scripts. In most practical systems to date, rule-based representations have dominated. In these, the knowledge is coded in the following form:

IF <something is the case>
AND <something else is the case>
AND <something else
AND <etc. >
THEN
<conclusion 1 follows>
AND <conclusion 2 follows>
AND <etc.>

Thus an expert system to identify the origins of tattoos might have a rule:

IF the tattoo is of a fish
AND the colour of the fish's scales is pink
THEN the origin of the tattoo is China

The *inference system* is a program which uses logical procedures to draw inferences from the knowledge encoded in the knowledge base. It is the inference system which decides, for example, the information to be sought from the user of the program, which rules should be 'fired' (activated) as a result of what the user types and when a conclusion to a particular line of inquiry has been reached.

Two kinds of logical procedures are employed by most inference systems. *Forward chaining* or *data-driven* reasoning tries to establish which conclusions follow from particular instances or data, while *backward chaining* or *goal-driven* reasoning tries to establish whether particular (conclusions) are valid by seeing whether their premises are true. Practical expert systems use a combination of both types of inference.

The third component of an expert system is the *user interface* which is essentially another program designed to make dialogue between the user and the inference system easier. In many cases, the ideal interface would be one in which the dialogue takes place in ordinary English (this would be a 'natural-language interface'), but existing systems fall well short of this ideal, and will continue to do so until AI researchers make more progress on the problem of machine understanding of unconstrained natural language. However, as we will see later, it is possible to construct useful NL interfaces for applications where the domain of discourse is tightly constrained.

Modular Structure

A point to note is the modular structure of an expert system. This is important because of the provisional and changing nature of knowledge. If the knowledge-base is too tightly interwoven with the code of the rest of the program, every addition or modification of the knowledge will require delving into the program itself. Hence the pattern has been for the three components to be kept separate with clean and well-defined dividing lines between them. This means that any one component can be changed relatively easily.

Expert system 'shells'

Since expert systems are relatively expensive and time-consuming to construct, there is a great incentive to find ways of making some of the code reusable. Some researchers, noting that in many cases two of the components - the user interface and the inference system - will be relatively unchanged from one application to another, came up with the idea of a 'shell' which contains these two subsystems and into which different knowledge-bases can be 'plugged in' (Figure 3.2).

Such shells are now commercially available and are often recommended as a quick way of developing expert systems applications. In some cases, they do seem to represent a cost-effective way of getting started, but they also have serious limitations since using a shell implies an assumption that its inference procedures are suitable for the particular problem. There is a trade-off, in other words, between the convenience of using a shell and the need to respect the unique requirements of the problem for which an expert system is required.

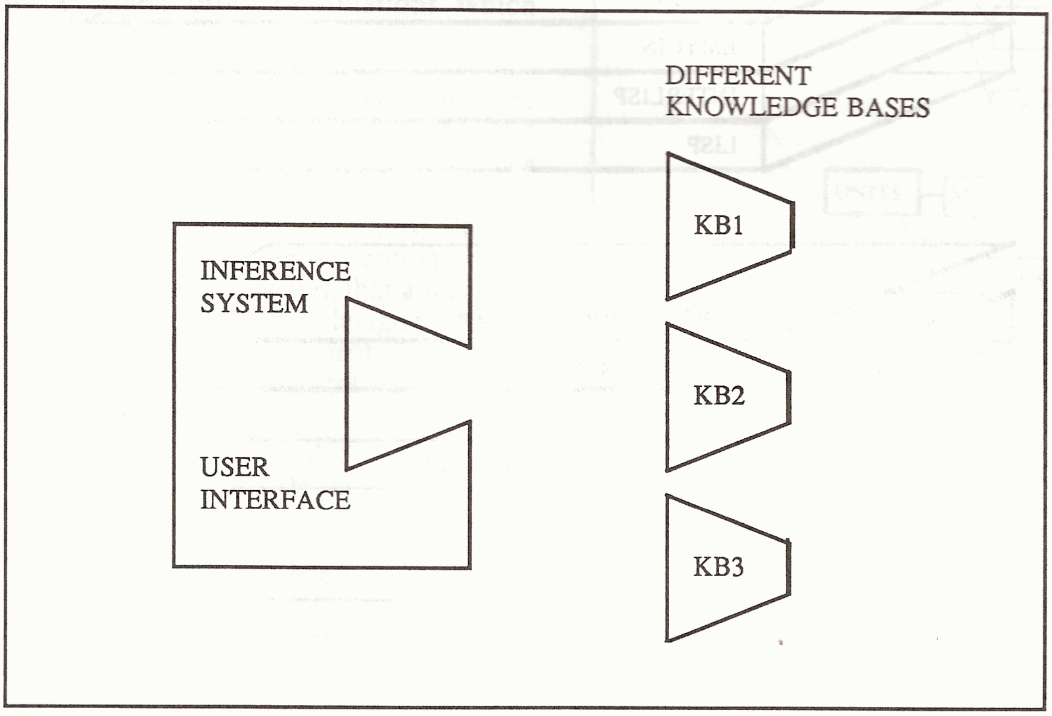


Figure 3.2

The shell concept

Expert Systems in use

In general, expert systems are *consultation* systems designed to make expertise more generally available than it would be if one had to require the presence of a human expert on site. Most systems work by requiring the user to volunteer information and by querying him or her at various points during the consultation.

Early systems (i.e. those constructed in the 1970s) were in very specialised areas like medical diagnosis, mathematics, mineral prospecting and research chemistry. Among these were:

MYCIN - a system for diagnosing blood infections

DENDRAL - a system for inferring molecular structure from experimental data

MACSYMA - a system to assist mathematicians, engineers and scientists in solving complex mathematical problems

HEARSAY - a system to demonstrate the possibility of a speech-understanding system

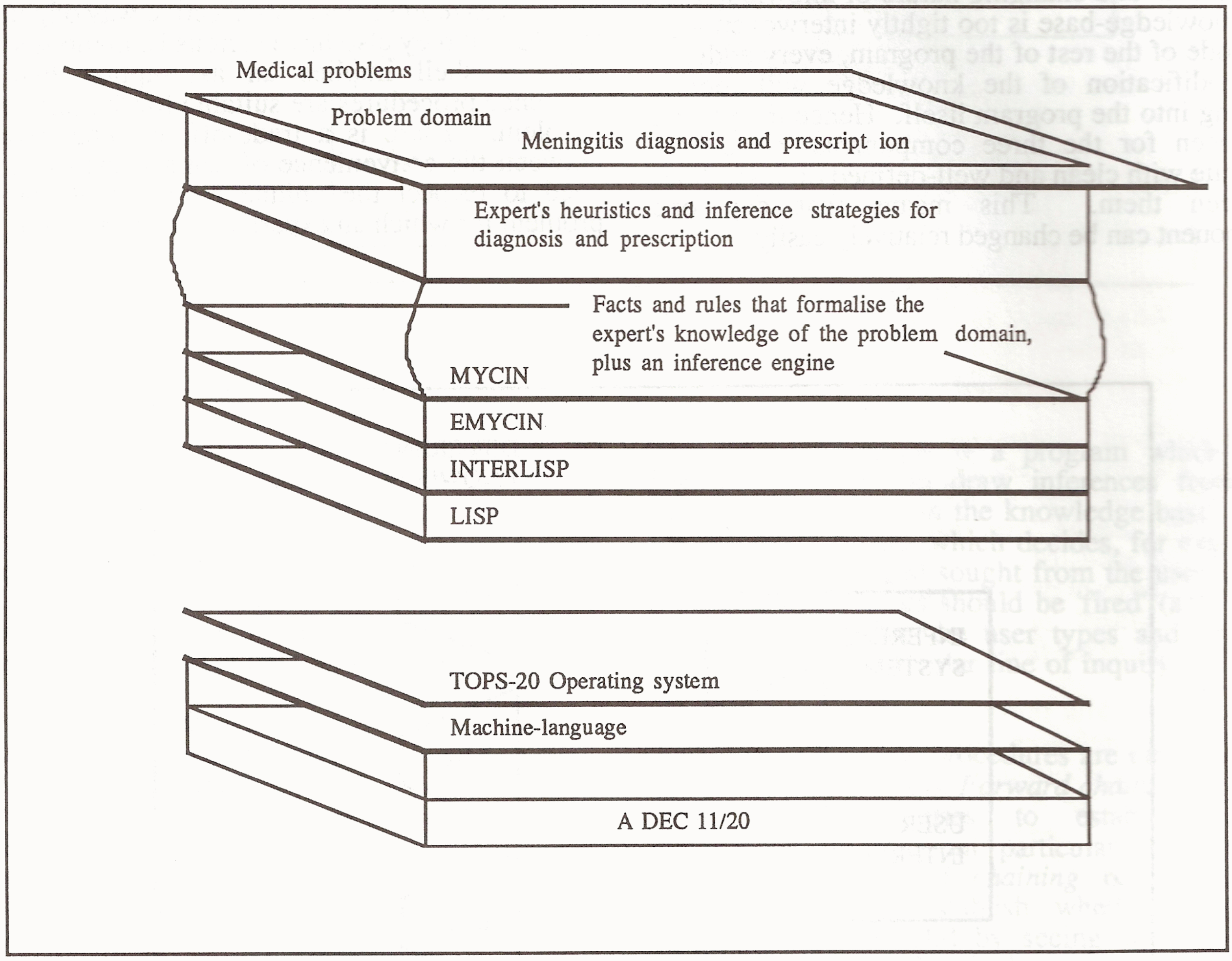


Figure 3.3 (Adapted from Harmon and King, page 90)

The levels of software between MYCIN's domain (blood infections) and the computer hardware on which the system runs.

More recent systems include:

XCON - (originally R1) - a commercial expert system which configures Digital Equipment Corporation's VAX computer systems. Its input is a customer's order, and its output is a set of diagrams displaying the spatial relationships among the components on an order. These diagrams are used by the engineers who install the system. XCON checks an order for validity and ensures that the components are compatible etc. The system is in daily use worldwide in Digital, and current levels of operations and installation would be impossible without it. The corporation devotes a significant amount of resource annualy to maintaining and updating the knowledge-base as the VAX range changes.

GENESIS - a system to help molecular geneticists design complex experiments to determine the nature of a DNA molecule.

DELTA/CATS-1 - a system developed by the General Electric company to help railway maintenance service diesel locomotives.

DRILLING ADVISOR - a system to assist oil drilling rig supervisors in resolving and subsequently avoiding problem situations which arise in drilling.

All of the above systems are sophisticated and complex pieces of software which required significant computational and personnel resources to construct. In terms of performance, many of them are quite 'expert' in their domains, though rigorous 'clinical trials' are the exception rather than the rule. It is also worth noting that a majority of the best-known systems are designed to be used, not by lay people, but by professionals in their domain, and in experimental rather than commercial environments. Thus MYCIN is designed to be used by hospital physicians rather than by, say, nurses. There is still very little experience of building systems for naive users.

In the last few years, there has been an explosion of commercial interest in the potential of expert systems and many companies have been experimenting by building systems for in-house use or sale to external customers. There has also been an upsurge in the number of system-building tools (shells, programming environments, specialist hardware, etc.) available in the market-place. (See Figure 3.4 for a representation of the relationship between tools and systems.)

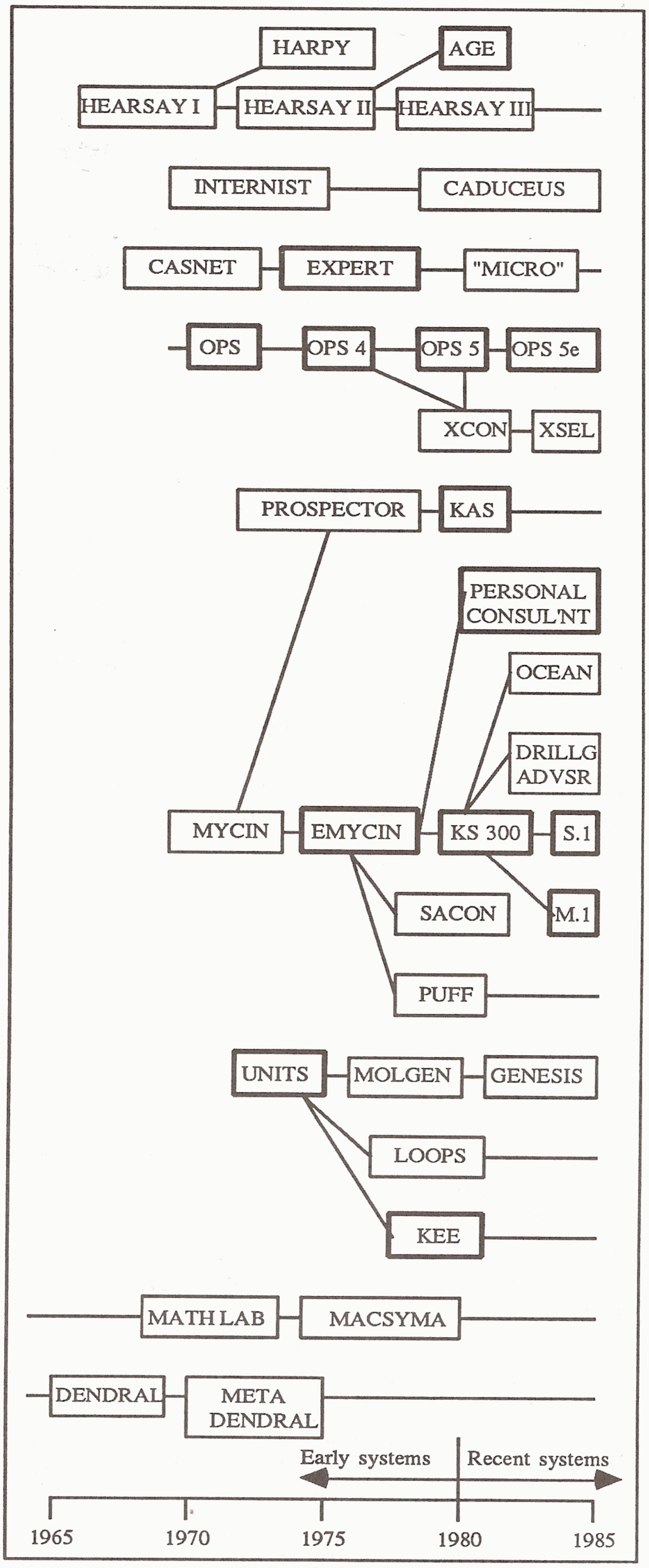


Figure 3.4 (Adapted from Harmon & King, p 91)

Development of several expert systems shown in relation to sytem-building tools available at any given time.

However, most of the systems which have resulted from the most recent surge of activity are of the proof-of-concept, 'toy', demonstrator or prototype kinds - i.e. with relatively small knowledge-bases (30 - 200 rules). Serious commercial systems (say with 500 - 5000 rules) are still relatively rare, and making the transition from demonstrator systems to ones that are commercially or operationally viable is not simply a matter of scaling up. A large rule-base, for instance, will have complex interactions between rules which are not detectable by observation: it may have, in other words, unexpected 'emergent' properties.

Because of the relative paucity of serious expert systems in use in real-life environments, we do not as yet have much objective data on how effective they are in practice - i.e. on the extent to which they actually contribute to solving or ameliorating the problems for which they were designed. This is also partly a reflection of the fact that one generally only hears about systems which the designers regard as being 'successful'. However, some research - e.g. Kidd(1985) pp. 9 - 20 - suggests that some end-users find the interfaces and dialogue styles of existing systems rather off-putting and irrelevant to their immediate concerns.

For an extensive catalogue of known systems, see Chapter 25 of Waterman (1985).

State of the Art

Expert systems are clearly where the action is in applied AI research at present. Some impressive and elaborate systems have been built and shown to work. It has been demonstrated that it is possible to capture, represent and access specialist knowledge of a high level in specific domains. The basic idea behind the technology - that of making scarce expertise more widely available on some basis - is sufficiently attractive to ensure continued funding and growth in the field. However, extensive diffusion of expert systems into the market-place requires that a number of serious problems be solved, of which the following are probably the most significant.

(1) The knowledge-engineering bottleneck. Currently it is generally necessary to elicit the specialist knowledge from one or more experts. This is generally done by some variant of in-depth interviewing, but this is known to be an exceedingly time-consuming, erratic and expensive business. There is currently a good deal of research into computational tools and investigative methodologies which might ease the bottleneck, but these have not yet made much impact on the problem. One approach which is frequently advocated by some is to use machine induction algorithms to infer rules from examples or case studies which have been previously analysed by experts. The assumption is that whereas experts may find it difficult to articulate their knowledge in the abstract (e.g. in an interview), they are quite good at discussing concrete examples. This approach is often a way of extracting rules quickly, but the reliability of the knowledge obtained is questionable.

(2) Availability of suitable delivery vehicles for expert systems. 'Delivery vehicles' means microcomputers sufficiently powerful to run consultation systems. Many of the most famous expert systems in the business run only on expensive and rare computer systems. The increasing availability of high-performance microcomputers (e.g. the IBM PC/AT, Apple Macintosh Plus) means that the right delivery vehicles for expert systems software are, or will soon be, available, so this is not likely to be a limitation for much longer.

(3) The acceptability of expert systems software to end users. This is important because the experience of some organisations suggests that it is possible to build expert systems which are cognitively sophisticated and effective at solving their target problems, but which the intended end-users still refuse to consult because they find the user interface clumsy or difficult, because the dialogue mode imposed by the program is uncongenial, or because they cannot understand the explanations provided by the system for its conclusions and questions. Alison Kidd's research (Kidd, 1975) has shown, for example, that the dialogue which takes place between human experts and their clients is much more sophisticated and evenly-balanced than that which takes place between an expert system program and its user. If expert systems are to find wide acceptability, then this problem must be solved.

However, solving it will require that some fundamental difficulties in AI research have to be

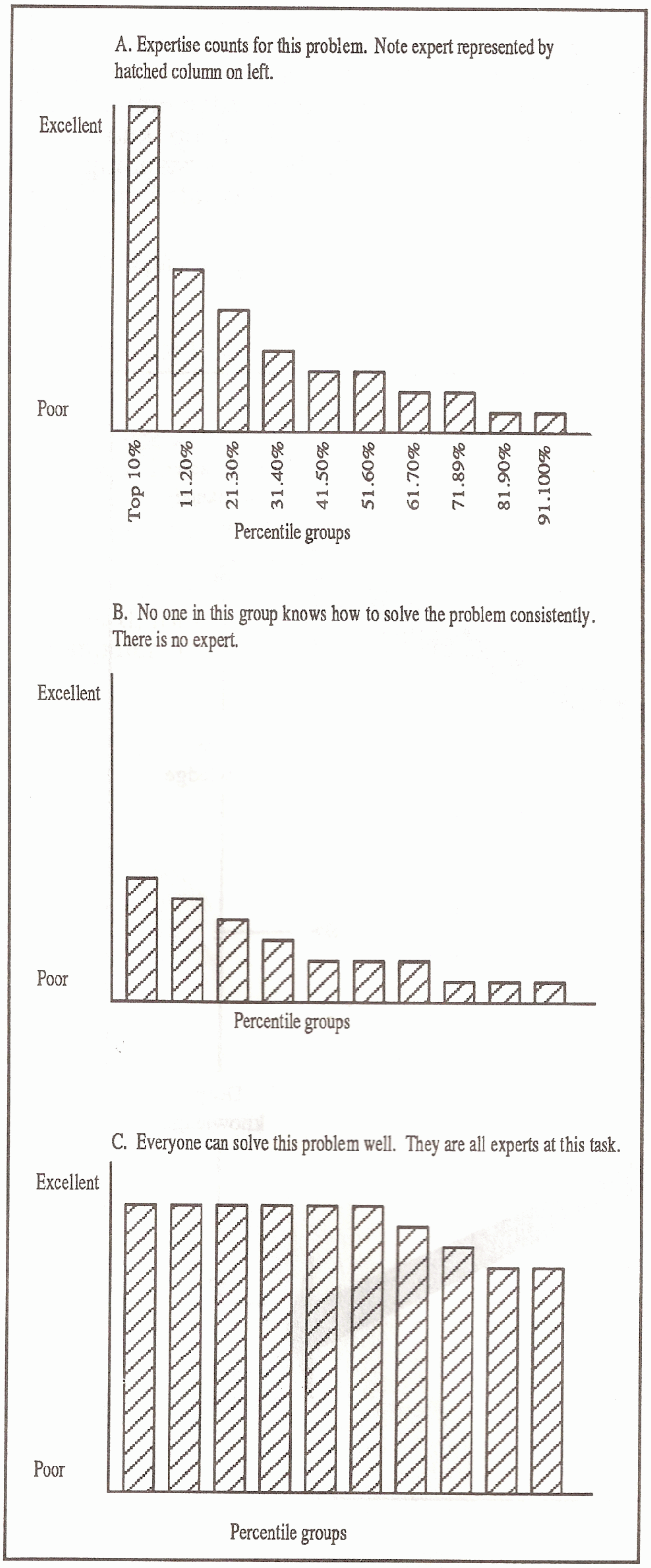


Figure 3.5 (Adapted from Harmon & King p 32)

Distribution of expertise in three different domains. Only the top domain is suitable for expert systems development.

solved first. Sophisticated dialogue implies, among other things:

* natural-language *understanding* (as compared with *processing*.

* the ability to (i) handle mixed-initiative dialogue, (ii) be sensitive to the immediate context of the consultation, (iii) 'learn' from the progress of the consultation, and (iv) provide explanations of reasoning that are meaningful to the client.

These problems are nowhere near being solved at present.

(4) Uncertainty about appropriate domains of application. It is not clear at present which domains (i.e. application areas) are suitable for expert systems deployment. Current heuristic rules suggest that problems which can normally be solved by talking to an expert for a relatively short time on the telephone are likely to be amenable. Other heuristics suggest categorising domains along the lines shown in Figure 3.5.

Allied to this is the limitations imposed by the implicit assumption of almost all expert systems that human experts are accurately described as *problem solvers*. A 'problem' is a perceived discrepancy between an actual and a desired state of affairs. To call something a 'problem' therefore implies that both the actual and desired states of affairs are known. 'Solving' the problem then involves finding an optimal way of bridging the gap between the two.

This kind of bridging activity may well be a reasonable description of some kinds of human expert, but it is an inadequate description of what many professionals do for a living. Some writers (e.g. Schon, 1984) have argued persuasively that professionals (architects, medical practitioners, lawyers, etc.) are not problem-solvers but *problem setters*. That is to say, they specialise in helping clients reformulate unstructured or confused *situations* into 'problems' which can be solved. What this involves is working with the client to clarify descriptions of actual and desired states. The work of the expert in such cases is not 'problem solving' *per se*, but a much more creative endeavour involving the *creation* of problems which are amenable to solutions within the client's resources and time-scale.

Builders of expert systems almost always work on the premise that expertise is about problem solving. This is fine in its way, but it means that a large swathe of professional expertise may, in fact, lie outside the reach of the current technology, simply because embodying what Schon calls *problem setting* in a program is impossible with current technology and knowledge.

(5) Depth of knowledge. A major concern is that the knowledge encoded in expert systems is 'surface' knowledge - e.g. knowledge embodied in heuristic rules - rather than the 'deep' knowledge which results from scientific understanding of a domain. The distinction is illustrated in Figure 3.6.

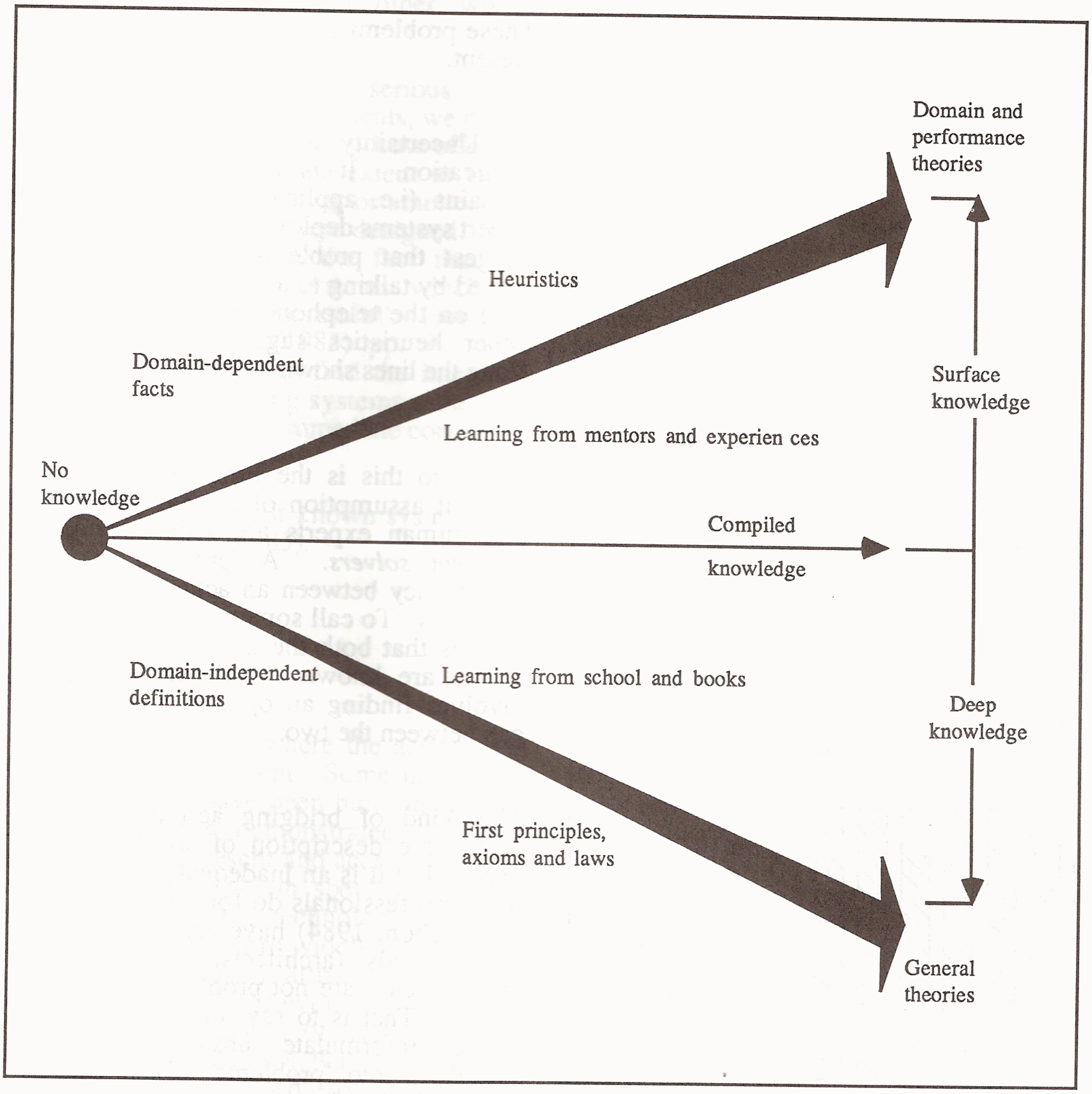


Figure 3.6 (Adapted from Harmon and King, page 33)

Surface versus deep knowledge

Thus, an electronic circuit trouble-shooting system might well have 'knowledge' in the form of rules like:

IF
capacitor C12 gives spiky output

THEN
replace capacitor C12

AND
test capacitor C30.

Such knowledge however represents only empirical knowledge about component C12. The program does not 'know' anything about capacitors and their electrical properties. In a straightforward trouble-shooting application, this might not be so important. But if it were proposed, for example, to imbed the system within a computer-based training system designed to teach trouble-shooting in electronic circuits, then such surface knowledge might well prove inadequate.

The incorporation of 'deep' rather than 'surface' knowledge in expert systems may have implications not only for tutoring systems. Surface knowledge inevitably means that the knowledge is very specific to a particular application. Yet the truth is that problems in different domains may have an identical causal structure. Consider, for example, a Hi-Fi amplifier which will not work because its fuse has blown, and a car which will not start because its battery has run down. Though both problems are radically different, they share a common cause, namely that neither is receiving the power it requires to function. The ability to spot the essential commonality of both situations is something which humans have, or can be trained to acquire; embodying such abilities in machines is difficult, but essential if expert systems are to become more 'generalisable' accross domains.

Chapter Four

LANGUAGE , KNOWLEDGE AND LEARNING

AI research into some other areas - notably natural language (NL) understanding, knowledge elicitation and representation and learning has considerable relevance for training - even of the conventional face-to-face sort.

Natural language understanding

'Natural language' (NL) is a catch-all term covering a wide range of research into the communication between people and machines. It includes language input, output and understanding. Of these, the last presents by far the most intractable problems at the moment. Yet it is also the one which offers the most potential benefits in terms of applications. For example:

Conversational interfaces to other types of computer software and hardware - databases, word processors, operating systems, computer-based training systems, expert systems, robots

Machine translation systems for translating from one vernacular language to another.

Document-understanding systems that enable a computer to read and understand enough of the material in written documents to summarise the contents, redirect points of importance to other members of an organisation and organise and store the information contained in the documents in intelligent ways.

Approaches to Natural Language understanding

An important distinction should be made between NL *understanding* and NL *processing*. Historically, the latter came first, and is still sometimes confused with understanding.

Processing Systems

Processing systems can accept NL input and produce NL output in an apparently intelligent manner, but in fact are simply operating some kind of simple-minded conversion process and are invariably thrown when the input moves away from the range of topics/words/ideas envisaged by the programmer. The best example of such a system is Joseph Weizenbaum's ELIZA, a program which apparently responds to user input in the manner of a sympathetic psychotherapist. A typical dialogue with ELIZA goes something like this (user input in italics, ELIZA's response in capitals):

Men are all alike.
IN WHAT WAY ?
They're always bugging us in one way or another.
CAN YOU THINK OF A SPECIFIC EXAMPLE?

Well, my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE?

He says I'm depressed much of the time.
I'M SORRY TO HEAR THAT YOU ARE DEPRESSED.

And so on.

Superficially, this is impressive at first, but it is really just processing based on a very simple idea. It has a pre-stored set of string patterns which it looks for in the input sentence, together with a set of stock responses for each pattern. Thus, whenever the term 'mother' or 'father' appears in the input, the system will respond with the question:

TELL ME ABOUT YOUR PARENTS

But ELIZA will gave exactly the same response to the input sentence:

Prince Charles' mother is going to France tomorrow.

NL Understanding

NL understanding is a much more ambitious activity than processing. A NL-understanding system must be able to do much more sophisticated things. For example:

Respond to questions in an appropriate manner.

Paraphrase a statement

Make inferences from what it is told.

Translate from one language to another.

Identify references - i.e. connect pronouns with persons.

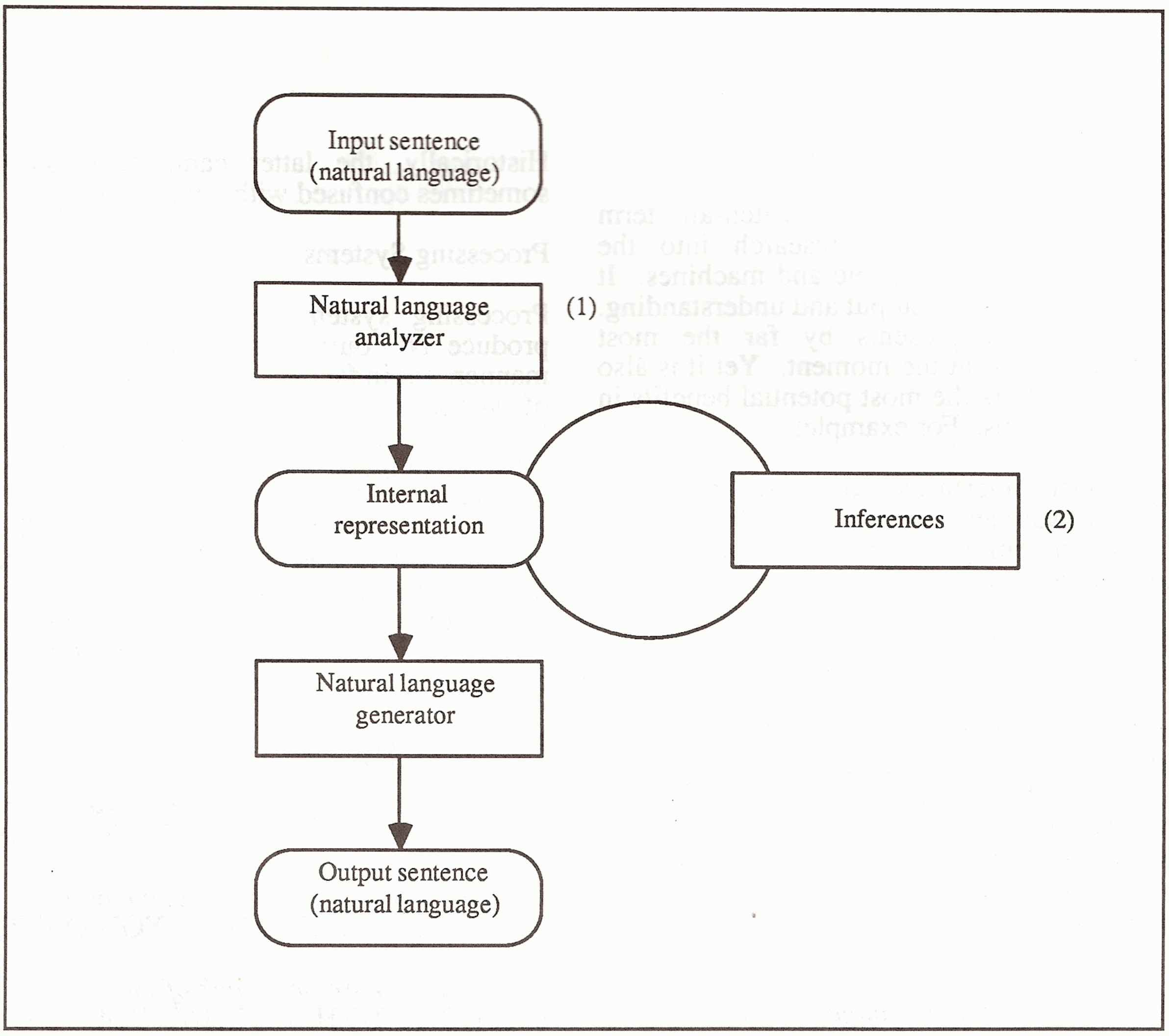


Figure 4.1

Understanding natural language

The key to NL understanding is that the system should be able to analyse the input and translate it into some internal representation language which is accessible to inferencing procedures and other intelligent operations.

Two kinds of analysis have traditionally been employed in order to do this. The first focusses on *syntax* (how sentences are constructed), the other on *semantics* (the meaning of words). The two approaches are complementary, in the sense that a working system requires both. They are combined to generate a paraphrase of an input expression in a special internal representation language.

Syntax is analysed by parsing - identifying what function each word plays in a sentence and whether the sentence is 'legal' in terms of a particular grammar. A variety of grammars (based on different theories of communication) have been used in parsing systems.

Semantic analysis focusses on the meaning of words and phrases and is done by associating words and their roles in a sentence with information about the problem domain stored in a knowledge base. What this means in practice is that the success or otherwise of semantic analysis is dependent on the quality of the domain-specific knowledge contained in the knowledge base. This implies that it may be possible to extract the meaning of sentences which relate to very specific areas or applications (tightly-constrained domains) - for example, database queries - but that computerised semantic analysis of everyday language is impossible with current or forseeable technology, simply because the domain-knowledge needed to conduct the analysis is too extensive.

Understanding speech

To the general problems described above, speech adds another layer of difficulty with its special problems of noise, pronounciation, the changes in phoneme sounds depending on whether they are spoken in isolation or with other words, the ungrammatical and incomplete nature of much of everyday speech, the problems posed by homonyms (words which are spelled differently but pronounced similarly), etc.

Summing up

In summary then, it appears that the era of machine understanding of unconstrained natural language is still a long way off, and certainly beyond the time-horizon of this Report. But we can realistically expect to see considerable growth in the use of NL-understanding systems in constrained applications where semantic analysis is possible because it is possible for the program to 'know' every relevant fact about the domain of the application. Examples of such applications are: 'intelligent' front ends to database, operating and other software systems. The general effect of such systems will be to make the user interfaces of most software packages less intimidating and more robust. This will have important benefits for computer-based training systems - as we shall see later.

As far as speech-driven systems are concerned, we can expect to see some *constrained* applications, but nothing capable of responding intelligently to conversational input on a wide range of topics. A speech-driven word-processor with reasonable performance and functionality (say a vocabulary of 10000 words and good adaptivity to different users), for example, is believed to be achievable before 1990, and IBM and an Alvey consortium already have working prototypes of this type of thing. Such systems will, inevitably, have some impact on the demand for secretarial skills, but the scale and timing of the effects will depend on economic and other factors.

Relevance to training

NL understanding systems potentially have two kinds of training implication.

(1) By making complex software simpler to use they may reduce the training requirements for certain kinds of jobs. For example, many organisations maintain complex database systems on which the organisation depends for its commercial survival. Sophisticated use of such databases, however, requires that operators are familiar with query languages with complicated syntax. For some such systems, it can take weeks or months for new operators to become proficient enough to conduct anything other than rudimentary inquiries. NL interfaces may reduce this problem.

(2) In a limited number of cases (e.g. speech-driven word processors), successful

implementations of constrained NL understanding systems may reduce the demand for certain types of skills and thus have an indirect effect on training.

Knowledge Elicitation

The knowledge embodied in an expert system may come from diverse sources - databases, textbooks, learned articles, etc. - but mainly it comes from one or more human experts. The task of *eliciting* knowledge from experts is therefore a central one in the construction of expert systems. It is also a task which is known to be difficult and problematic, and a great deal of effort and research has gone into ways of making the elicitation process more effective and productive.

Most elicitation techniques involve intensive discussions with, and interviewing of, an expert, followed by intensive analysis of transcripts of the interviews. This is sometimes called *protocol analysis*. Considerable expertise has been built up in the form of elicitation techniques, among them the following (Waterman,1985, page 158; see also Gamack and Young, 1984):

On-site observation - watching the expert on the job.

Problem discussion - exploring the kinds of data, knowledge and procedures needed to solve the problem.

Problem description - having the expert describe a prototypical problem for every category of answer in the domain.

Problem analysis - presenting the expert with a series of realistic problems to solve aloud, probing for the rationale behind his or her reasoning steps.

System refinement - having the expert present the knowledge engineer with a series of problems to solve using the rules (or other forms of knowledge) derived from earlier interviews.

System examination - having the expert examine and criticise the prototype system's rules and control structure.

System validation - present the cases solved by the expert and the prototype system to outside experts.

Implications for training

Much industrial training involves trainers who are not themselves experts in a particular discipline extracting the relevant knowledge from those who *are* domain experts in order to 'package' the knowledge in courseware of one form or another. In general, this elicitation process is conducted haphazardly within the training environment. Our conjecture is that elicitation could be much improved by conscious use of the techniques and tools which AI researchers have developed to assist the elicitation process.

Knowledge Representation

Much AI research has traditionally been concerned with *knowledge representation.* This is not just a matter of storing knowledge in the form of statements, but of "setting up a correspondence between a symbolic reasoning system and the outside world" (Bonnet, 1985, page 82).

Thus the knowledge that "John went to London" could be stored as a string of characters, but that would not enable a program to answer the question: "Who went to London ?" A better representation, from this point of view, would be to represent the knowledge in the following form:

Action: GO
Agent: JOHN
Source: ?
Destination: London
Tense: PAST
Means: ?

This includes information which might enable a program to elucidate the meaning of the phrase.

Numerous ways of representing knowledge have been proposed by AI researchers. The power of a representation is measured by (i) its ability to store information and (ii) make it accessable to inferential processing. Among the most widely-used representations are *rules, semantic nets* and *frames.*

Rules

Probably the most widely-used representation, and one which is suitable for capturing certain kinds of knowledge - particularly knowledge about recommendations, directives, diagnoses or strategies. Rules are expressed as IF...THEN statements of the form:

IF a flammable liquid was spilled
THEN call the fire brigade

IF the pH of the spill is less than 6
THEN the spill material is an acid

IF the spill material is an acid
AND the spill smells like vinegar
THEN the spill material is acetic acid

In a rule-based system, the domain knowledge is represented by rules of the above type. They are accessable to an inferential system which checks for instances where the IF portion of a rule is satisfied, in which case it 'fires' the rule and draws the appropriate conclusion. This may then trigger the IF sections of other rules, and so on and is called *forward chaining* in the jargon of the business. Alternatively, the inference system may look at the THEN portions of rules and, when it finds one which seems to meet a conclusion which it is trying to establish, looks to see if the premises (the IF clauses) are true. This is *backwards chaining* or goal-driven reasoning.

Rule-based representations are useful for some purposes, but have serious limitations. In particular, they appear to be vehicles mainly for *surface* rather than deep knowledge of the domain - i.e. they represent knowledge derived from empirical associations rather than from, say, scientific theory. This makes them specific to specific domains and difficult to generalise.

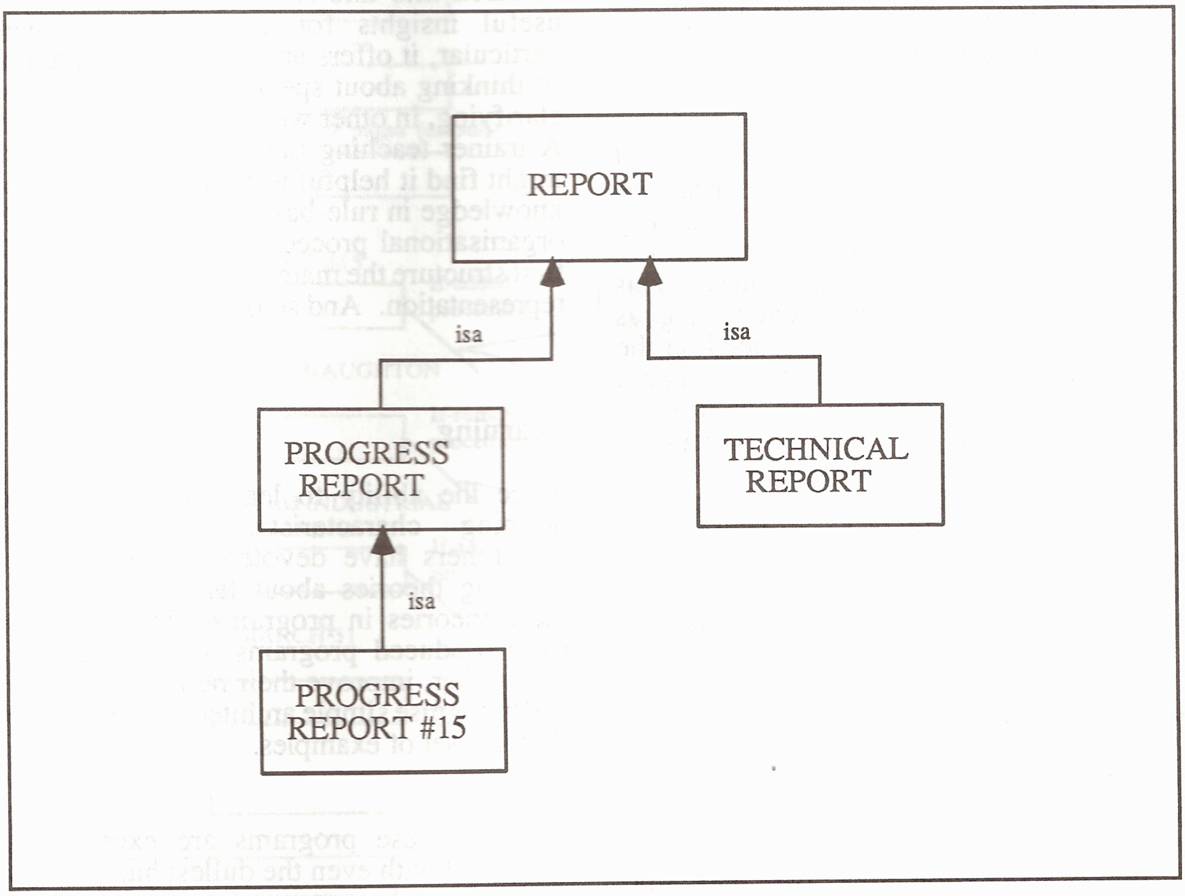


Figure 4.2

The concept of a written report

Semantic nets

Semantic nets were originally developed as psychological models of human memory, but are now used as a standard type of representation in some AI work. A net consists of points called *nodes* linked by *arcs* describing the relations between the nodes. The nodes stand for objects, concepts or events; arcs can be defined in a variety of ways, depending on the type of knowledge being represented. Common arcs used to represent factual knowledge include *is-a* (as in "John *is-a* man"), and *has-part* (as in "car *has-part* engine").

Frames

Frames are a way of representing knowledge about common concepts and situations. A frame is organised much like a semantic net - i.e. a set of nodes and relations organised in a hierarchy. Figure 4.3 shows a frame-type representation of the concept of a written report originally expressed as a semantic net in Figure 4.2 above.

The difference between a frame and a semantic net is that each node is defined as a collection of *attributes* and the *values* of those attributes. An important feature of a frame is that the 'slots' may contain not only assigned values or default values, but also *procedures* for obtaining values. This means that when a new frame is 'opened' - e.g. as a result of new information or a fresh need on the part of the system's user - various things can begin to happen automatically. The system, as it were, ransacks its knowledge to see what pieces of existing knowledge can be used to fill the vacant slots in the new frame. Frame-based systems are useful for problem domains where expectations about the form and content of the data play an important role in problem-solving - e.g. interpreting visual scenes or understanding speech.

The experience of AI research to date seems to be that there is no such thing as a catch-all representation for domain knowledge. Some representations are more suitable for some purposes, some for others. In addition, even within a single application area, one may find several different types of knowledge - each perhaps requiring its own representation. Gamack and Young (1984), for example, distinguished the following four types in knowledge about a fairly restricted technical domain (maintenance and fault-finding in technical plant):

(i) Knowledge of concepts and relations

(ii) Knowledge of routine procedures

(iii) Facts and heuristics

(iv) Classificatory knowledge.

In such a case, it might be sensible to represent (i) as a semantic net, (ii) and (iii) as rules, and (iv) as frames.

Relevance to training

Apart altogether from the fact that knowledge representation is an essential ingredient of any intelligent tutoring system (see Chapter 6), AI research into knowledge representation has some useful insights for conventional trainers. In particular, it offers an ordered and systematic way of thinking about specialist knowledge - a way of clarifying, in other words, what needs to be taught. A trainer teaching diagnostic skills, for example, might find it helpful to try and represent his or her knowledge in rule-based form. Someone teaching organisational procedures might find it helpful to first structure the material in terms of a frame-based representation. And so on.

Learning

Since the ability to learn is usually taken as a defining characteristic of intelligence, AI researchers have devoted considerable effort to devising theories about learning and embodying these theories in programs. In the process, they have produced programs that can 'learn' how to play poker, improve their performance at checkers and recognise simple architectural forms after being shown a set of examples.

Most of these programs are exceedingly crude compared with even the dullest human learner, and their relatively poor performance only serves to emphasise what a complex and sophisticated process learning is.

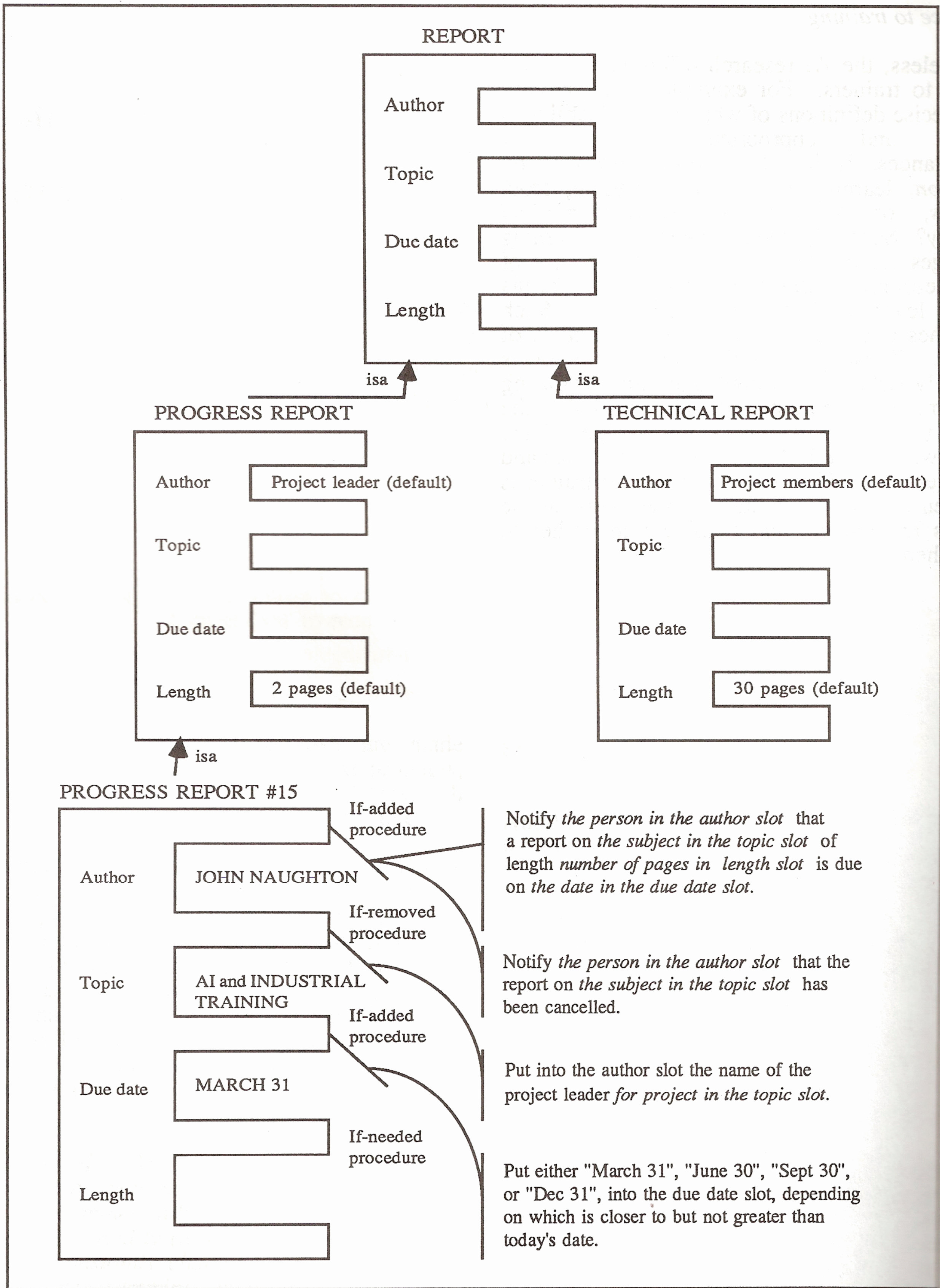


Figure 4.3

Frame-based representation

Relevance to training

Nevertheless, the AI research offers some useful insights to trainers. For example, it encourages more precise definitions of what *kind* of learning is required and appropriate in particular circumstances: is it rote learning, learning by *instruction*, learning by *analogy*, learning from *examples*, or learning by *exploration and discovery*? Secondly, the AI approach to learning encourages the trainer to try to model the state of the trainee's knowledge in order to see where his or her learning has been deficient. Such approaches are often based on 'buggy' models of learning. These assume that learning involves essentially 'debugging' knowledge acquired during the learning or training period. Initially, the student is seen as having a knowledge of the domain which is full of errors, inconsistencies and misconceptions. The task of the teacher/trainer is then seen as that of identifying the 'bugs' in the student's knowledge, and helping him or her to correct them.

Chapter Five

THE GENERAL IMPACT OF AI

Forecasts of the general impact of AI research are inevitably speculative because of the lack of established products and therefore of conventional markets. This chapter looks at the factors which will affect the evolution of markets for these products, and also at the views of leading AI practitioners on the state of their art and its likely prospects in the future.

Although some extravagant forecasts are made about the size of future markets for AI products, predictions about the general impact of AI research on the industrial scene are inevitably unreliable.

This is partly because of uncertainty about what is and is not an 'AI product'. Some experts maintain, for example, that the most enduring effect of AI will be the diffusion of its characteristic modes of thinking through the computer industry and through *that* to industry as a whole. Such views are summarised in statements to the effect that "Artificial Intelligence is that part of computer science that the rest of us can't afford to implement just yet". Or, as John Seely Brown of Xerox put it,

the real payoff of Artificial Intelligence during the next few years may not be in expert systems but rather in commercially exploiting the artificial intelligence mentality (a mentality for coping with ill-defined constantly changing problems) and the intelligent programming environments that have emerged to enable artificial intelligence researchers to cope with immensely complex programs.

Supply and Demand

Another problem with predictions of the likely market for AI products is that most forecasts emerge from the *supply* side of the equation - i.e. they are produced by or for *vendors*. There is very little reliable information about the *demand* side of the equation, mainly because it is impossible to estimate the demand for products or services which don't yet exist.

Key Factors

It is possible, however, to identify some of the factors which will influence the development of the market for AI products, in terms of positive and negative influences on supply and demand. A recent report by the Frost and Sullivan market research and consultancy organisation, for example, gives the following breakdown of these market factors (Frost and Sullivan, 1985):

Supply side

Positive factors:

New market opportunities for suppliers.

(The AI market opens up lucrative possibilities for vendors or appropriate products. To take just one example, it offers the computer manufacturers a way of selling high-performance, expensive workstations into companies which hitherto bought only mainframes and personal computers.)

State intervention

(The Japanese 'Fifth Generation' project is, like many such ventures in Japan, largely the brainchild of the industrial ministry MITI. A number of Western governments, concerned lest the project should give the Japanese a strategic hold over the information-technology industries of the future, have been actively promoting AI research and products.)

Falling equipment costs

(AI systems require powerful computers. The rapidly declining cost of computing power helps to bring AI applications to a wider community of users.)

Price competition

(This is simply a carry-over from most forms of IT, in the sense that applications developed in a research (and hence less cost-conscious) environment are eventually supplanted by cheaper and better systems. The market in advanced workstations is a good example. Three years ago, AI workstations cost between $50k and $100k. Today (1986), comparable systems may be bought for $10k - $20k.)

Technical breakthroughs

(Technological breakthroughs - e.g. in parallel processing - will have a positive effect on the supply side. However (see below), revolutionary breakthroughs - especially on the software side - are expected by very few, if any, of the leading researchers in the field.)

Presence and strength of suppliers

Intervention and funding by EEC

(The EEC, concerned lest Europe should lose out to the US and Japan in the IT industries of the future, has funded large research programmes like ESPRIT.)

Negative factors:

Scarce and expensive resources required for AI work.

(As observed earlier, the greatest bottleneck of all is the shortage of people skilled and experienced enough to work as knowledge-engineers.)

Working AI systems are large and expensive to build.

(...and to maintain. The Digital XCON system, for example, cost several millions of dollars, and requires continual updating and mainteneance (to the tune of several million dollars a year). Also, the problems involved in going from 'toy' or 'demonstrator' systems to commercial systems are not just those of 'scaling up'.)

Investment constraints

(Much applied AI research has long pay-back periods - longer in most cases than those allowed by British banks and financial institutions.)

Lack of agreed standards

(The AI world is characterised by large numbers of similar but independent products. There are, for exmaple, numerous different dialects of LISP and of Prolog, the predominant languages of AI programming, At present, no single corporation is powerful enough to impose a *de facto* standard of the kind that IBM imposed on the microcomputer market. To date, the most powerful pressure for standardisation has come from the US Department of Defense's DARPA agency.)

State concern about related issues

(For example, the impact of AI on jobs.)

The problem of creating a market for such products.

Protection of home industries.

Demand side

Positive factors:

National economic success.

(Economic growth will generate a demand for more sophisticated computing facilities.)

Growing familiarity with computers

(The rise of the home micro, together with the influx of computers into offices and schools means that computers are slowly but surely being regarded as just another technological device.)

Desire to establish a competitive edge.

(Some companies - for example in the pharmaceutical industry - are coming to regard specialist knowledge as a key strategic product. One such firm, ICI, already offers an expert system for pest management which is available to farmers via Prestel.)

Social acceptance of computer-based advice.

(Growing familiarity with computers may make people less suspicious of the idea of programs which embody *knowledge*.)

Deteriorating standard of manned-service alternatives to AI systems.

(In some cases (e.g. sophisticated diagnostics and fault-finding), it is increasingly difficult to provide a satisfactory service using conventionally-trained operatives. This is particularly the case in fast-moving industries. For example, it is said that the introduction of System-X telephone systems will generate a demand for technical skills in maintenance and installation that only knowledge-based systems cn actually deliver the required expertise to the desired spot in the field.)

The value of scarce and expensive expertise.

(Many organisations - e.g. in the financial sector - are beginning to realise that the expert knowledge and skill of their staffs are their most valuable, if also most intangible, asset.)

Negative factors:

Rejection of computer-based advice.

(Apart altogether from the phobic reaction which some people exhibit in relation to computers, there are legal and procedural issues surrounding the whole issue of advice-giving programs. For example, who would one sue for negligence, malpractice or incompetence, if the original legal advice was provided by an expert system.)

Inability to adopt new habits.

(Always a problem.)

High prices.

(Applied AI products are expensive, and likely to remain so in many cases.)

Ethical considerations.

(As above: is it ethical to refuse an individual a mortgage simply on the basis of an expert system designed to assess lending risks ?)

Rejection of machine intelligence.

(The very idea of 'artificial' or 'machine' intelligence is anathema to some people.)

Critical Factors

We can identify from the above some of the critical factors which will affect the diffusion of AI technology through industry. These are:

The possibility that fundamental theoretical breakthroughs in the field will make it easier to develop and implement AI-based products.

Whether there are likely to be significant shortages of personnel with appropriate skills for the diffusion of AI technology into industry.

Whether there are existing AI-based products with strong commercial or industrial potential in their present or realistically forseeable forms.

Theoretical breakthroughs?

By definition, major conceptual or theoretical breakthroughs are impossible to predict, but the best judgement of the leaders in the field (see later) seems to be that they are unlikely at present.

One of the reasons for this apparent consensus is a widespread belief that insufficient attention is being paid to fundamental research in the AI field. Most of the current glut of funding is for *applied* work of a relatively shallow and short-term kind, and most of the resulting products are in fact based on AI research done ten or even twenty years ago. The crucial question is whether sufficient people are today doing the kind of fundamental research needed to support the applications of twenty years hence. An instructive comparison is provided by biotechnology, another field with great commercial and industrial potential in which the applied research and development is underpinned by a massive theoretical research effort in molecular biology. If the problems of understanding intelligence and replicating it in machines are (as many people believe) at least as complex as those of genetic engineering, should we not be devoting a comparable amount of resources to fundamental research in AI ?

Skills shortages

Here we can be more specific. It seems highly likely that, if present trends continue, the growth of the market for AI products will be significantly restricted by the scarcity of suitably-trained staff - i.e. people with experience of AI work and products. To take just one example, it has been estimated at the time of writing (February 1986) that there are probably no more than 500 skilled knowledge engineers (i.e. staff who can elicit knowledge from experts and imbed it in software in such a way as to get acceptable levels of performance) *worldwide.* Some computer companies - notably Digital Equipment Corporation-- have already recognised this and have embarked upon ambitious training programmes to 'grow' their own experts rather than trusting to the vagaries of the job market or recruitment specialists.

This scarcity of appropriately skilled staff is rendered more serious by the extensive training period required to produce suitably qualified AI practitioners. One analyst, Karl M Wiig of Arthur D Little Inc., maintains that it takes at least 12 months' intensive training on sophisticated equipment to produce an AI programmer, and five *years* to produce a competent general AI practitioner who can take a market or organisational need and produce a working commercial system to meet it (Wiig, 1985). If these estimates are accurate, then they suggest that for substantial numbers of companies to be in a position to supply AI-derived products and services in 1991, they should already be embarked on ambitious R&D and training programmes of the kind in place in Digital. But there is no evidence that significant numbers of companies have programmes of the requisite scale at present, though many are 'exploring', in a tentative way, AI concepts and approaches.

Existing products with industrial potential

Here, the obvious candidates are expert systems, natural language systems, AI-enhanced robotics and computer-assisted training. Because of the importance of the latter, we will leave it to the next chapter.

As far as *expert systems* go, it seems probable that the next decade will see considerable expansion in the provision and use of such systems in a wide range of applications. But quite how they will affect the availability and demand for skills (and therefore training) is still difficult to predict. A crucial question is the extent to which such systems will be used as *replacements* for skilled personnel or as *intelligent assistants* ("power steering for the mind", as one commentator called them). The current bias is heavily in favour of the latter, and for most higher-level and professional skills is likely to continue that way for the time-span of this Report.

Natural language systems: We have seen that general-purpose natural language understanding systems are still a long way off. But the next decade is likely to see a plethora of NL systems designed to operate in domains and applications constrained enough to enable the system to do semantic analysis. The major applications of such NL systems will be concerned with providing 'intelligent front ends' to software and hardware - for example, database software and laboratory instruments respectively. These will have training implications, in the sense that equipment or software which currently requires intensive training will require less in future.

The other application area where NL systems are likely to make an impact is in the office - particularly in the field of speech-driven word-processing systems. As already remarked, such systems already exist as prototypes and there are strong commercial pressures to bring them to market. Several companies - for example, Kurtzweil and IBM - have products in this area which will shortly be commercially available. This development will have a significant impact on certain types of office work - particularly secretarial, and will therefore also have training implications.

The Views of AI practitioners

Practitioners' views about the achievements and future of AI

A recent issue of the journal *Artificial Intelligence* was devoted to an edited compendium of answers given by a variety of AI researchers to a questionnaire about the achievements and potential of their field. This informal survey revealed a wide range of opinions, with occasional bursts of agreement about topics such as the impact of commercial pressures on AI. Here is a selection of quotations from the journal.

On the 'state of the art'...

During the past ten years, AI has entered a new phase. The technical ideas that fueled the early excitement have been pushed to the point where we are experiencing the limits of their applicability. It is not at all obvious that significant theoretical advances can be made by continuing in the same direction. At the same time, the revolution in hardware size and cost has made it possible to apply those techniques with economic effectiveness in a huge range of situations. As a result, the field has several major growth areas. One emphasises applications and the resultant profits, with little pretense to theoretical advance; another leads to highly speculative and imprecise intuitions about learning, memory, etc. in an attempt to grasp at something that can get us beyond the limitations of the existing paradigm; another focusses on the design of better parallel ... hardware in hopes that difficulties are in the end only those of scale.

Terry Winograd.

The most significant advance in the last decade has been the appreciation of just how complex the nature of thinking is. We have come to understand what the issues are.

Roger Schank

AI has grown enormously in the last ten years, and the media - and the money-men - have now discovered it. But growth need not imply development. With a few exceptions, what we have seen is 'more of the same' (or, at most, 'better of the same'). The recent explosion of funding and publicity is due to commercial and political factors, not to intellectual advances in the field. The central problems of AI, and the theoretical basis of its achievements, have remained essentially the same. Most of the 'advance' has been in technological efficiency, not scientific understanding.

Margaret Boden

As far as I know very limited progress has been made with the problem of coping with natural language input which is really natural, that is, includes many kinds of slips, grammatical errors, incomplete sentences, etc.

Aaron Sloman

On the diffusion of AI tools to other areas...

Aaron Sloman suspected that "an enormous amount" can be gained under two headings: (1) simple expert systems to replace manuals and other documentation; (2) use of AI tools to build non-AI software.

"Current industrial interest in AI may not be so much in AI per se but rather in the programming methodologies (e.g. exploratory programming) and tools that the AI community has been so instrumental in creating. Thus, if AI is broken down, metaphorically, into the low, middle and high road approaches, it is the first two levels that are attracting industrial attention."

John Seeley Brown

On the future of AI ...

The industrial sponsorship is likely to spell disaster for AI. With rare exception, businesses want results that can be profitable in a year or two. With everyone hopping on the AI bandwagon, we are bound to see numerous industrial labs staffed with poorly-trained personnel. These labs will not be capable of producing very much beyond simplistic expert systems. Soon, AI will be characterised in the mind of business and government leaders in terms of those systems. I, for one, would not like to bet the future of AI on the potential usefulness of expert system building tools or expert systems built with those tools at random toothpaste companies.

Roger Schank

I wouldn't be surprised if what is now called AI will eventually subsume most of computer science in general.

Pamela McCorduck

The re-establishment of robotics has to be the second major advance. Its long-term impact for AI far exceeds the work in expert systems, for it will force AI to cope with the real physical world. However, it will be some time before the impact is felt in a major way...

My third candidate is the development of an understanding of the instructional process. For those mostly in AI rather than cognitive science, this will be identified as intelligent tutoring systems, and will perhaps seem a slender reed. But concomitant major advances have occurred in psychology in understanding cognitive skills and how they are acquired, including the different representations employed by novices and experts. This area provides an exemplary integration of scientific knowledge from AI and cognitive psychology. I think it marks an important advance."

Alan Newell

Areas of AI that are likely to see most progress in the next decade: Machine learning; problems of design and planning; reasoning under the guidance of several, qualitative and quantitative, models; theory formation and concept discovery; technologies for building and refining expert systems with knowledge bases of about 10K rules.

Saul Amarel

Progress in the next decade will come in discovering those domains in which the assumptions and techniques of AI are appropriate. Much work on expert systems has this flavour - the secret of success isn't in building the right program, but in finding the right domain. We will also begin to find better ways to integrate the kind of deduction done by AI systems with the reasoning done by people with a background of experience. The result may not be 'intelligent machines', but intelligent uses of machine capabilities.

Terry Winograd

The wave of commercial enthusiasm for AI was fueled by the possibilities created by microcomputers. It will find a substantial niche, but one that is far from revolutionary. There are many applications for AI techniques in industry, but the net impact will be like that of the introduction of a new useful technology (e.g. plastics), not a fundamental change in the way things are done. There is bound to be disillusionment, given the grandiose claims being made by many researchers, including a few of the recognised leaders in the field. The result won't be as total as it was, say, with machine translation, since the criteria for success are less well-defined. But it is likely that in spite of successful applications in many specialised areas, the public mood in ten years will be one of "What they promised didn't happen", instead of "We're on the way". This is because the public has been led to expect machines that really think, understand language, etc., not controllers for industrial processes or programs to diagnose engine flaws.

Terry Winograd

Conclusion

Estimates of the size and speed of development of the market for AI products differ widely and are inherently speculative. One recent report (Frost and Sullivan, 1985) predicts that the total market for commercial expert systems in Europe (hardware and software) will grow from over $18 million in 1983 to over $3800 million in 1990 (figures in constant 1984 US dollars). This implies a rapid take-up of the technology and is difficult to reconcile with the restrictions on growth due to the absence of skilled personnel in particular.

Chapter Six

ARTIFICIAL INTELLIGENCE AND COMPUTER-BASED TRAINING

Computer-Based Training (CBT) is an important and expanding technology for training, yet one which is currently limited by critical weaknesses in the available software. This Chapter examines the problems of conventional CBT, outlines some of the research which has been done on 'intelligent' computer-assisted instruction systems and discusses how this research might contribute to the enhancement of conventional CBT systems.

Computer-Based Training (CBT)

Computer-Based Training (CBT) is a general term used to describe a variety of activities and products whose common denominator is that they involve the use of a computer in some part of the training process. The training in question is not necessarily connected with the application or manufacture of computers, though the computing industry and its customers are probably the biggest users of CBT.

Generally, what is involved in CBT is the generation of teaching material ('courseware') by a trainer which is then made available on an individual basis to a trainee sitting at a computer terminal or microcomputer. The computer presents the student with successive 'frames' of textual and graphical material, and periodically tests his or her understanding of the material by asking questions (usually of the multiple-choice variety). On the basis of the student's answers, the program then branches to another set of frames (or recapitulates the previous frames if the student's answers and the courseware author's instructions indicate that course of action).

CBT has been available for over 20 years, and with the advent of powerful, inexpensive desktop computers and sophisticated display devices such as videotape and videodisk, has been expanding rapidly. This expansion seems set to continue, or even to accelerate. Laurillard (1986) found that 26 per cent of British companies were already users of CBT, and that about half of the remainder claim to be planning to introduce CBT in the near future. Comparisons of the findings of UK surveys with those conducted in the US suggest that CBT use in American companies runs at about twice the British rate.

The advantages cited for CBT over conventional face-to-face methods vary from application to application. To the company or training manager, potential benefits include cost, consistency, high throughput, speed and a solution to shortages of skilled trainers. In many areas - for example, the launch of a new product, the commissioning of a new manufacturing technology, the introduction of new office procedures, etc. - conventional CBT is a cost-effective way of getting the training message accross.

From the point of view of the trainee, CBT may also offer significant advantages. Trainees can learn at their own pace, for example; teaching material can be active rather than passive; for some people having to use a computer for learning is inherently motivating; and computer simulations can provide trainees with realistic representations of situations which might be too dangerous or expensive to organise in real life.

Criticisms of conventional CBT

CBT is now an established technology, with a range of products and vendors, annual conferences and all the other trappings of a maturing industry. However, from a pedagogic point of view, many of its products have been heavily criticised. Among the strictures which have been made, the following stand out:

(i) Much courseware is desperately unimaginative, amounting in many cases to little more than electronic page-turning.

(ii) Much courseware seems to be based on naive or non-existent models of how people learn. There is, for example, a strong streak of behaviourism in some CBT material, together with a tendency to confuse the ability to *name* concepts with evidence of understanding of them, and an almost total absence of interest in the teaching lessons to be learned from student error. The prevailing model is to display a gobbet of information, ask some Multiple Choice Questions to 'test' the student's 'understanding' of the material, and then to branch on the basis of student performance in the test. In so far as CBT vendors have anything approaching an educational philosophy, it sometimes seems to be merely that "we do it for half the price".

(iii) Conventional CBT programs are unable to do the things they teach. Thus a program that purports to 'teach' people about fault-finding in automobile engines does not itself 'know' anything about automobile engines. The student therefore does not have the option of saying "show me" to his or her computerised tutor. The absurdity of this situation is readily appreciated if one imagines a human teacher setting up to teach, say, integral calculus, without having any knowledge of integration.

(iv) Conventional CBT systems do not have natural language interfaces. That is to say, they cannot 'understand' entries from the student that do not fall within a rigidly prescribed pattern - usually menu-driven.

(v) Related to (iv) is the fact that most conventional CBT systems cannot respond sensibly or creatively to input errors. At the trivial level, this applies to elementary spelling and typing mistakes. At a higher level, conventional systems lack any means for making creative teaching use of student mistakes. This is especially important since often the most important diagnostic evidence presented to a tutor is provided by the errors a student makes.

(vi) Conventional CBT systems cannot improve their teaching in the light of experience.

(vii) Preparation of CBT material using conventional authoring languages is a labour- and time-intensive process using software tools which, by the standards of the computing industry, are rather cumbersome.

Consequences of CBT's limitations

Some of the above deficiencies are more serious than others, but overall they amount to a serious indictment of a technology on which great hopes are placed by the training industry. Some of the current limitations of CBT are essentially restrictions which render it less *efficient* or *effective* than it might otherwise be as a training medium. Thus the limitations of conventional authoring systems mean that large investments of trainers' time and energy have to be made in order to generate even the most elementary courseware.

But some of conventional CBT's limitations have the potentially equally damaging effect of *restricting the areas/subjects/topics that can be reached by the technology*. To see why, we need briefly to consider the age-old distinction between *education* and *training*.

Education versus training

The purpose of bringing up this distinction is not to re-open ancient philosophical debates but to try and identify why some topics and applications might lie beyond the reach of conventional CBT and why that lack of reach might be important in the future.

Education is usually seen as the process of teaching a person to think in abstract terms. It therefore places emphasis on theoretical knowledge and is the kind of thing a company seeks when it sends a promising manager on an MBA course. Education may, of course, make an individual capable of performing some practical task or achieving a practical goal, but that is essentially a side-effect of the process.

Training, in contrast, may involve some theoretical or conceptual material, but only does so in a practical context of teaching an individual to carry out a procedure or accomplish a particular goal. Little if any value is placed on conceptual knowledge for its own sake. A bank adopts this course when it puts all its junior managers through a Loan Analysis workshop.

Using this distinction, it is easy to see why conventional CBT has a considerable contribution to make to training - and equally why it has little to offer if abstract or theoretical material is an essential element in the teaching. The difficulty arises because many of the areas for which trainers would like to use CBT in the future are ones in which the mere teaching of procedures are not sufficient. For example, in an old-fashioned chemical plant it might not have been important for operatives to understand anything of the dynamics of the plant's overall behaviour simply because the plant was not an integrated system. But a modern operative in a new, computer-controlled and highly-automated plant might need to understand system dynamics; or he or she might need to have a good working knowledge of instrumentation and measurement theory. In other words, there are some technological areas for which procedural knowledge and training is not sufficient - which in turn implies that inflexible and 'unintelligent' CBT is likely to have little to offer in such environments.

The central weakness of conventional CBT lies in the way it represents the knowledge it purports or attempts to impart. This knowledge is stored in 'frames' which are essentially merely strings of characters and/or pictures. Knowledge stored in this way, however, is inaccessible to any kind of manipulation or logical inference within the machine. All the conventional system can do is store and display its frames in accordance with some scheme devised by the courseware author. But it cannot, for example, analyse the *content* of a frame and relate that to the contents of other frames or to some unexpected input that the pupil has just typed.

Yet, ultimately, some facility for doing this is required if CBT is to become more adaptive and 'intelligent'. And since knowledge representation is one of the central areas of AI research, it is clear that improvements in conventional CBT require an input from that field.

In fact, there is a long tradition of AI interest in the problem of devising tutoring systems. A number of intriguing systems have been built - mainly as research vehicles or prototypes. This research has also led to the evolution of a specification for an 'Intelligent Tutoring System' (ITS) - a kind of 'ideal type' which exists as a goal rather than an actual achievement. In the remainder of this chapter we will first of all examine this specification, after which we will briefly survey some of the tutoring systems which AI researchers have built. Finally, we will identify specific areas in which conventional CBT might be enhanced by the incorporation of AI concepts and techniques.

Concept of an Intelligent Tutoring System

The general concept of an Intelligent Tutoring System (ITS) is of a computer program which can provide an individual student with teaching comparable with that obtainable from a competent human tutor. At present - and probably for some time to come - it must be reiterated that this represents a goal rather than an achievement, though some interesting attempts at intelligent tutoring systems have been constructed (see later). At the very least, the interaction between an ITS and a student should be:

* highly adaptable to cope with individual differences between students;
* enjoyable and efficient in educational terms; and
* capable of allowing the student to control the interaction whenever possible.

These requirements lead to a specification for an ITS along the lines shown in Figure 6.1.

Figure 6.1 suggests that the essential components of an ITS are:

(1) *Domain expertise* - i.e. specialist expertise in the domain being taught by the system. This is necessary for several reasons; for example, it is required so that the trainee can, in extremis, ask the system to solve the problem he's been set by it; domain expertise is also required in order to be able to provide meaningful explanations to the student, and to enable the system (as compared to the courseware author) sometimes to generate

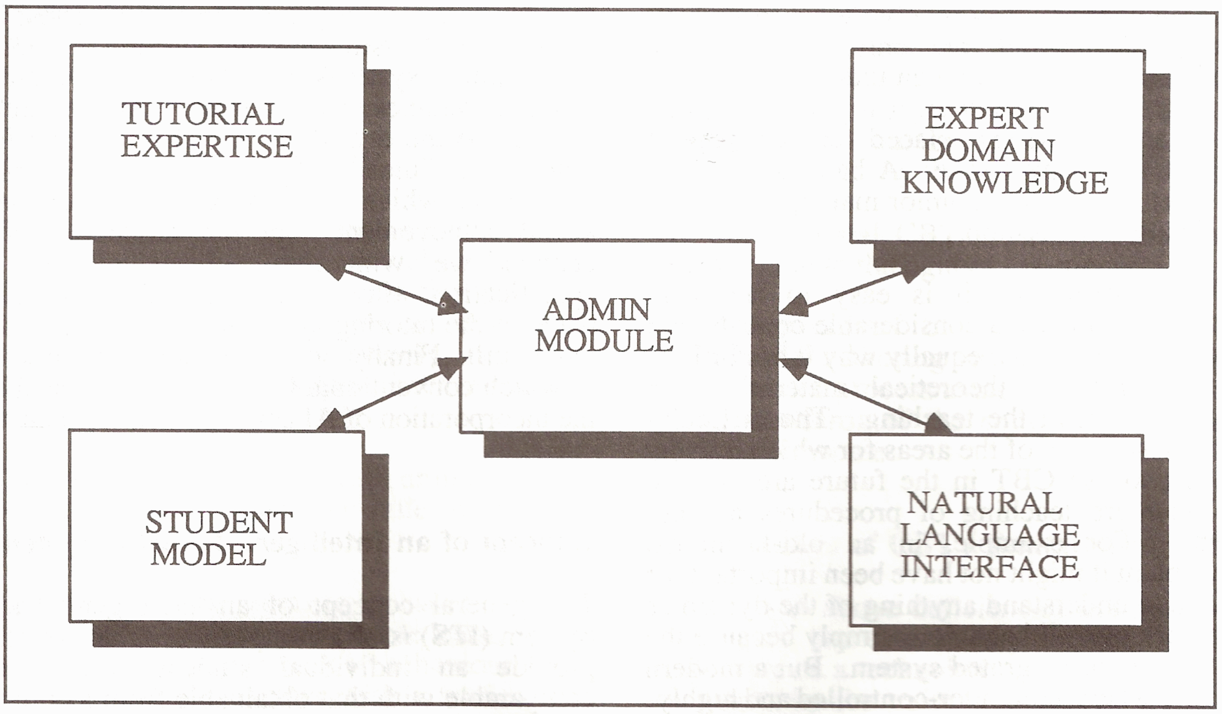


Figure 6.1

Structure of an Intelligent Tutoring System

problems which will test the student's grasp of the material or the system's model of the student's developing understanding, or both.

(2) *Teaching expertise* - i.e. knowledge about how to present different concepts to different kinds of learner, about judging progress, assessing the degree of cognitive overload experienced by the student, and so on.

Note:In many conventional teaching situations, both domain and teaching expertise are embodied in the same person - the teacher. In many training applications, however, there is a degree of separation between the two. Many industrial trainers, for example, cannot be experts in most of the material they present.

(3) *A student profile or model.* This is the central component of an ITS and is essentially a dynamic (i.e. changing) representation of the state of the student's knowledge of the subject being taught. It is used to decide how to assess the student, how to proceed with the tuition, which material to present next, which to recapitulate and so on. Most conventional CBT systems make little if any attempt at student modelling in this sense, and their failure to do so is probably the single most important limitation of the conventional approach.

(4) *A user interface* - i.e. a subprogram through which all interactions between the student and the other elements of the system are conducted. In principle, this would probably be based round a natural-language understanding system, though that might not be the most effective or efficient interface for some tutoring applications. What is important, however, is that the user interface of an ITS should be robust and 'intelligent' in a way that interfaces of conventional CBT systems generally are not. It would not, for example, be thrown by spelling mistakes and it would be capable of making intelligent guesses about the meaning of sentences and expressions inputted by the user.

Milestones in ITS research

Over the past decade and a half, a number of attempts have been made to implement in working systems some of the ideas embodied in the ITS specification. In a survey, O'Shea (1981) highlighted the following as significant:

Smallwood's geometry-teaching system (Smallwood, 1962) .

This is significant because it breaks from the standard question-and-branch modus operandi of most CBT systems by using past information about branching decisions. For a student at a particular level of mastery, the best block of material to be next presented is taken to be the best block for past students who had similar histories (i.e. similar paths through the branching network) to this student.

Leeds Arithmetic Teaching Programs (Woods and Hartley, 1971)

A series of arithmetic teaching programs was implemented and tested in Leeds in the late 1960s and early 1970s. Their significance derives from the fact that they were based on a model of the difficulty of arithmetic tasks and were able to generate and administer drill and practice in arithmetic problems of various levels of difficulty. The response-sensitivity of the programs stemmed from their ability to generate material suitable to a given student's level of competence; administer different types of feedback; and generate remedial material depending on student errors.

Kimball's Integration Tutor (Kimball, 1973)

This program teaches integration and attempts to implement the requirement that CBT systems should know what they teach. The student carries out integrations at a terminal and the program checks each transformation to see that it has been applied correctly. The program does not use pre-stored solutions to integration problems but uses two AI programs, one for integration (Moses' SIN), the other (Hearn's REDUCE) for algebraic simplification. Using these two, Kimball's tutor can solve all the symbolic integration problems which it sets for students. The program also has an archive of problems and solutions and will select examples for the student if requested. The teaching strategy is interventionist and based on the idea of 'student trouble thresholds'. When one of these is exceeded, the program intervenes and engages the student in dialogue.

Kimball's program is self-improving in the sense that if a student solution to an archive problem is 'better' (i.e. has fewer steps) than an archive solution, then this student solution is adopted and becomes the new archive solution to the problem.

SOPHIE (Brown, Burton and de Kleer, 1982)

Widely regarded still as effectively the state of the ITS art, the SOPHIE project was mounted for the US Air Force to teach electronic troubleshooting. Three systems were built in a five year period from 1973 to 1978, initially at the University of California at Irvine, later at Bolt, Beranek and Newman in Cambridge, Massachussetts. The domain is the troubleshooting of complex electronic circuits. The aim of the project was not to teach trainees to diagnose and locate faults on a *specific* piece of apparatus, but to produce a skilled troubleshooter with a sufficiently good conceptual understanding of electronics to be able to develop appropriate diagnostic steps on his own, to digest new information from technical manuals and to troubleshoot *unfamiliar* equipment. These aims are ambitious, but they are the kinds of things the next generation of CBT systems will need to be able to approach.

The SOPHIE system has several components. Firstly, it has an *expert troubleshooter* - i.e. a program capable of diagnosing and locating a wide variety of faults in electronic circuits. Secondly, it has a *coach* which is capable of watching the student's attempts at diagnosis and fault-finding and intervening when necessary. Thirdly, it has a sophisticated *natural language interface* which is capable of conversing with the student and which can understand terse and elliptical references to the domain. And finally, SOPHIE incorporates a simulated workbench which enables students to run their own experiments to explore the workings of the circuit under examination.

The system supports an extremely flexible interaction style. For example, it will allow the student to insert a fault in a part of the circuit and then watch the system's expert troubleshooter locate the faulty device. The student is then given the opportunity of locating the fault within the device, during which the coach will critique his or her work.

Comparison of ITS with conventional CBT

A central theme of this chapter is that CBT in its currently available commercial form is a problematic technology. A 'problem' is a discrepancy between an existing state of affairs and a desired one. The value of the ITS specification discussed earlier is that it gives us a yardstick against which to measure the limitations of the CBT technology which is currently available. Comparison of the ITS specification with conventional CBT suggests that four limitations are particularly important. They are:

* CBT's impoverished methods of knowledge representation.

* Lack of student modelling.

* The comparatively primitive tutoring styles available within conventional CBT.

* The non-generative nature of conventional CBT.

Let us consider each briefly in turn.

Knowledge representation

The prime advantage of an ITS, as specified, would be that the knowledge it was designed to impart would be represented internally in such a way that the tutoring software itself would be aware of its contents and be able to manipulate it. Thus, the software would have some way of 'understanding' the contents of frames and be able to relate them to the contents of other frames or to what the pupil types. This would enable an ITS to respond intelligently to inputs which, though unexpected (in the sense that they were never envisaged by the courseware author) are nevertheless sensible or interesting in the context of what is being taught.

As an illustration, consider the following situation (drawn from experience on a commercial product). A piece of conventional CBT courseware is teaching about thermocouples (i.e electrical devices for measuring temperature). The program displays some frames containing a description of what a thermocouple is, what it does, and what it looks like. It then displays a picture of a thermocouple with an arrow pointing to a point labelled "A" on the diagram. Point A, in fact, refers to the bimetallic strip which is the essential ingredient of a thermocouple. The student is posed the following question:

"What is the device shown at Point A ?"

In reply, the string "bimetallic strip" was typed in. The program however responded with a beep and the following message:

"Wrong - Point A shows a THERMOCOUPLE".

There are two points to be made about this. The first is the obvious one that the courseware author ought to have anticipated that some people would use the (perfectly correct) answer of "bimetallic strip" and ought not to have been penalised for so doing. More importantly, however, is the fact that the software itself ought to have been able to recognise that "bimetallic strip" is a relevant and correct concept in the context of what it is attempting to teach. But a conventional CBT system has no way of doing this, because it does not represent its knowledge in a way that would make that possible.

Knowledge representation is therefore an essential advantage of an ITS. However, there are grounds for thinking that a competent ITS might need to have more than one representation of its knowledge. This is because the representations needed so that the program can actually do what it teaches may be different from the representations needed to provide meaningful explanations to students. This goes back to a point made at the end of Chapter Three, where we observed that IF...THEN rules may be an effective way of representing expert knowledge about electronic troubleshooting, but cannot be used as the basis for an explanation of the system's actions because they embody surface rather than deep (theoretical) knowledge about electronic components.

Student modelling

An important attraction of an ITS is that it would overcome the inflexibility inevitably inherent in most conventional CBT material which arises because the courseware author has to envisage all possible branching paths during the creation (or maintenance) of the courseware. He or she thus has to make assumptions about the state of the student's understanding and knowledge at each stage in the study - assumptions which, in all probability, not fit any individual student terribly well.

If one thinks about any interaction in which learning takes place, there are two individuals and two cognitive structures involved. The teacher comes to the interaction with a well-constructed (if idiosyncratic) model of the domain and a set of teaching methods which he or she can perform (Elsom-Cook, 1986). The pupil, in contrast, also brings some (preconceived or other) knowledge of the domain, plus some learning methods which he or she has acquired in the past. The purpose of the interaction is to organise some negotiation between the individuals involved as a result of which changes take place in the student's model of the domain (and also, on occasion, in the teacher's). During the interaction, the good teacher will constantly be trying to assess the state of the student's understanding of the domain - in terms of concepts which have been grasped, concepts which have not been understood, and concepts which the student thinks s/he has understood but which in fact have been incorrectly understood. The teacher will, in effect, constantly be revising a *model* of the student.

An ITS would seek to do the same thing, though in a more explicit fashion, by means of a student model. Various kinds of student model are possible (Figure 6.2).

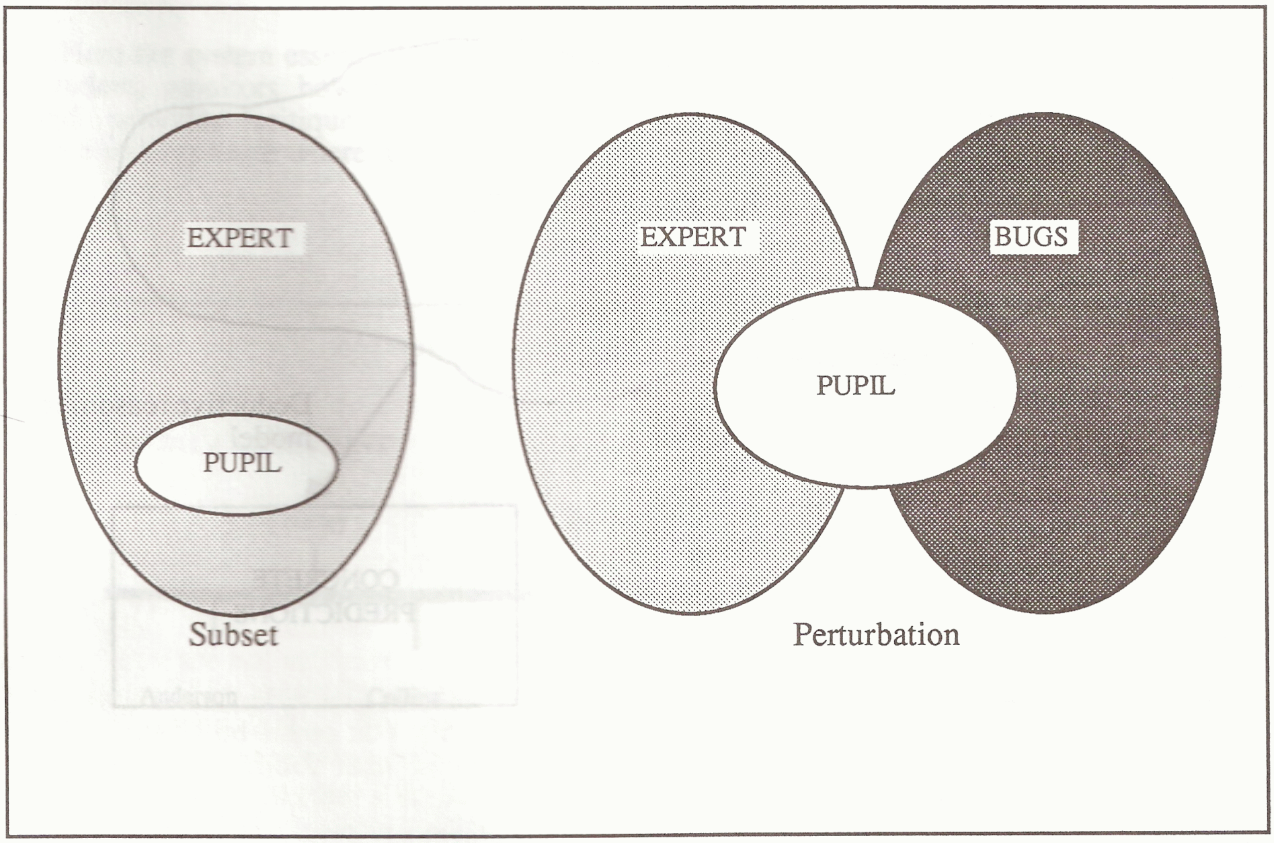


Figure 6.2

Two approaches to student modelling

In one, the student's understanding of the domain is seen as a subset of the teacher's: the student's understanding is essentially correct but incomplete, and the goal of the tuition process is gradually to enlarge the student subset. An alternative approach is to see the student's knowledge as intersecting both the teacher's understanding and a set of incorrect concepts (often called 'bugs). Teaching is then seen as the process of 'debugging' the student's knowledge. Some researchers (e.g. Elsom-Cook, 1986) regard the student set in both approaches to be essentially unknowable and favour a more empirical approach which essentially involves trying to determine the upper and lower bounds of a student's understanding at any given time in the interaction (Figure 6.3).

In this, the teaching program infers from observations of the student's behaviour what the limits on his understanding are. These inferences are then used to generate predictions about what the student would do if confronted with a specific type of problem or puzzle to solve. The program would then pose such a problem and observe the student's response. And so on.

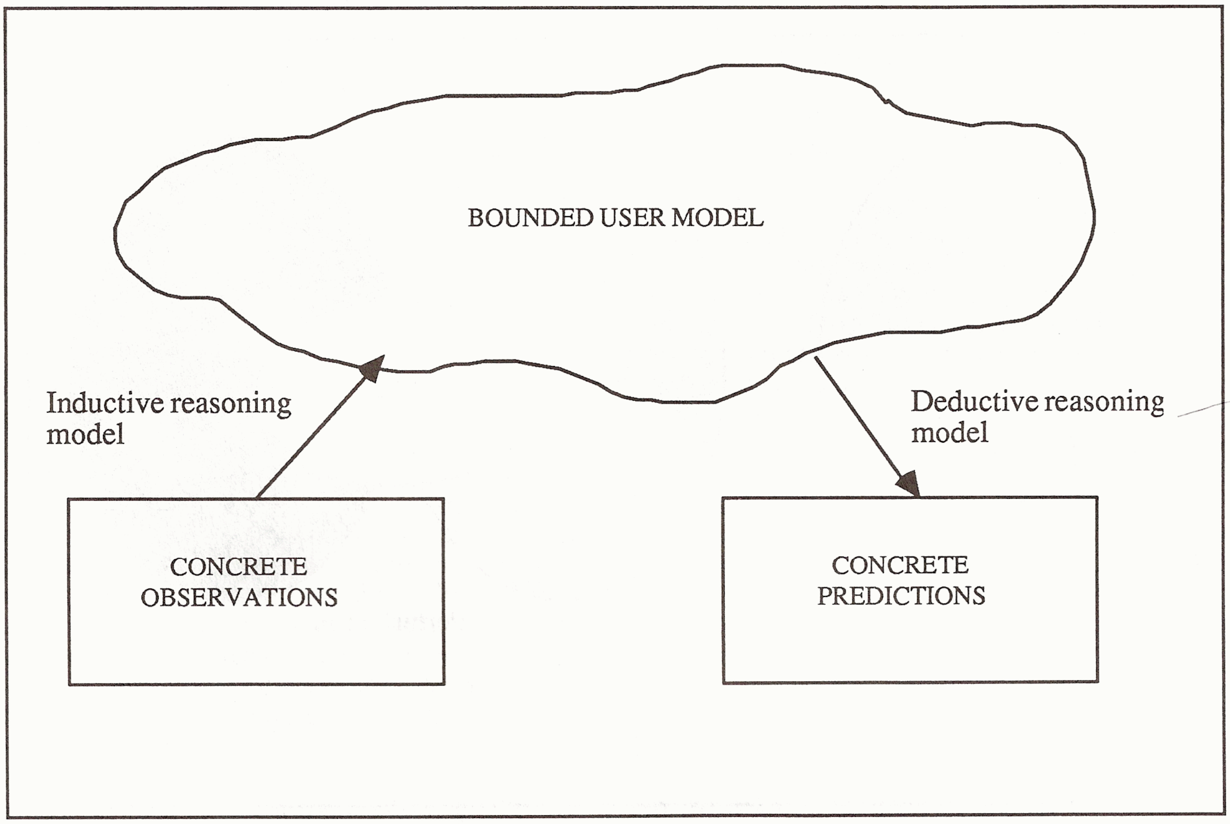


Figure 6.3

Bounded student models

Tutoring styles

Tutoring style is in fact an amalgam of two separate but related things - teaching style and *interaction* style. The former refers to the way in which a system (or a teacher) presents its material, while the latter refers to the ways in which the teacher and the learner interact.

Teaching styles:

Three particular styles can be distinguished in tutoring systems:

Expository. The system presents and explains its material to the student, who is expected to receive it passively - except for periods when he or she is expected to answer questions indicating whether the material has been understood.

Tutorial. Here the system essentially sets problems for the student, monitors how he or she solves them, and provides critiques of the pupil's performance and approach where relevant.

Drill and practice. The system concentrates on providing the student with plenty of typical small problems (perhaps based on teaching experience or task analysis) in order to give practice at rapid problem-solving in the domain.

There is a depressing conformity in the teaching styles of conventional CBT systems. As observed earlier, the vast mass of commercially-available courseware adopts an impoverished version of the expository style - i.e. it presents students with instructional material in a predetermined sequence of frames, poses some questions at suitable intervals and branches to other frame-sequences as a result of the scores achieved by the student. The style of such programs is therefore exceedingly rigid: the system determines what shall be presented when and how. If 'explanations' are provided they are based on 'canned' text.

Interaction styles:

Elsom-Cook (1986) has classified a number of well-known tutoring systems in terms of their interaction styles on a spectrum which ranges from 'total constraint' to 'total freedom' (Figure 6.4).

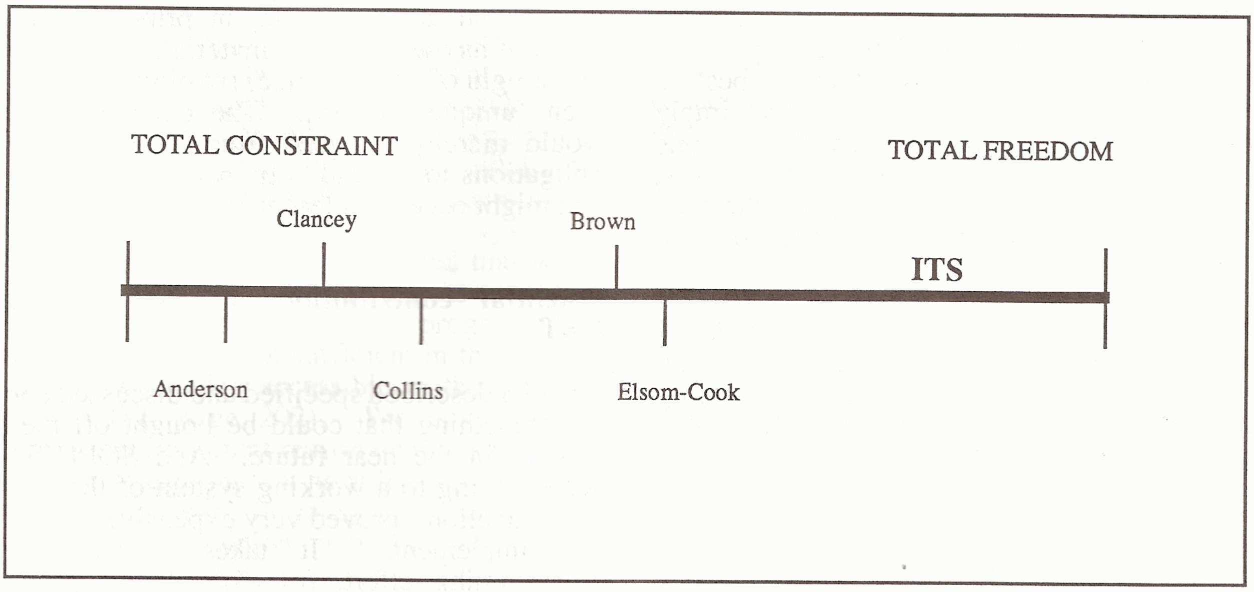


Figure 6.4

Tutoring styles of a range of systems

This scheme identifies a number of different types of interaction style:

Reactive systems are ones which process a single utterance by the pupil and react to it without reference either to the context or to any larger structures in the dialogue. Most existing tutoring systems operate in this manner.

Mixed-initiative systems attempt to recognise responses in which the pupil is attempting to take control of the interaction and allow that takeover to occur. They also look for opportunities to take control of the interaction. These systems require some model of conversation structure, but in most implemented systems this model is very simple.

Goal-directed, planning interaction mechanisms are ones which maintain a complete goal-structure for the dialogue and attempt to satisfy tutoring goals through that interaction. They have a higher level of 'awareness' of what they are saying, and of the context in which it is said. They are based on AI research on models of conversation and are currently "very rare" (Elsom-Cook, 1986).

This brief consideration of tutoring styles prompts a number of thoughts. The first is that the teaching and interaction styles of conventional CBT are inherently rather restricted and rigid. The teaching style is overwhelmingly expository and the interaction style is reactive. This is a problem because different students may respond best to different tutoring styles. Some students simply want to be told what to do and allowed to get one with it, while others want to argue, discuss, negotiate and interrupt whenever they feel the need to. Also, different material may require different styles.

Ideally, an intelligent tutoring system should be able to accommodate to these varying needs. But it ought to be said that the problems of building robust mixed-initiative systems seem formidable and have not yet been overcome.

Generative CBT

An important advantage sometimes cited by ITS enthusiasts is that such systems are 'generative' - i.e. capable of generating much of their own teaching and assessment material, or at least of improvising sensibly within given constraints. This is seen as a potentially a great boon to authors of courseware.

Designing a piece of conventional CBT courseware requires the author to structure the domain in an unnatural branching model. Material has to be divided into chunks of approximately equal size and are then connected into a structure by selection questions and problems. Sometimes, the material to be taught is not unduly distorted by being cast in this framework; but domains in which concepts are highly-interrelated are likely to be unsuitable for such treatment.

Having cast his or her material into the branching model, the courseware designer must then try to envisage all the possible situations which could arise in the use of the material, and to produce a segment of courseware (or a branching strategy) to cover that situation.

In principle, an ITS offers a way of overcoming this problem. The structure of the domain knowledge in an ITS would be explicitly represented, rather than being implicit in the whole program as in conventional CBT and CAL systems. This makes such structures closer to the cognitive structures of real teachers and allows direct manipulation of those structures. The capability of manipulating domain knowledge also means that an ITS could, in principle, generate much of its own teaching material, and could do so in the light of each (unique) set of interactions with each (unique) student. The courseware author would thereby be freed from his or her current obligations to try and anticipate every eventuality that might occur in a teaching session.

Potential contributions of AI research to CBT

The ITS described specified and discussed above is not something that could be bought off the shelf now or in the near future. And SOPHIE - the nearest thing to a working system of the requisite sophistication - proved very expensive to construct and implement. It takes *man-years* of programming effort, for example, to produce a single *hour* of SOPHIE-type tutoring performance. Consequently, full-blown ITS are not currently a cost-effective technology, and even when cost is not a constraint, the construction of such systems is beyond the technical expertise of all but the largest corporations.

Nevertheless, comparison of the ITS ideal with the reality of conventional CBT highlights a number of discrete areas where (i) improvement or enhancement is urgently required, and (ii) AI research has tangible techniques to contribute.

This leads to the conjecture that one of the most fruitful areas for future research could be into whether significant improvements could be made to conventional CBT systems by the incorporation of ideas, techniques or systems from AI research. Following on our earlier analysis, the following areas would seem worth concentrating on:

(i) Knowledge representation. We have seen that conventional CBT usually represents and stores knowledge in a form that renders it inaccessible to inferential procedures. This could be improved by borrowing techniques for knowledge representation from AI research. Specifically, the difficulty that most CBT systems cannot themselves do what they purport to teach might be overcome to some extent by imbedding expert systems within a tutorial program. This is, in fact, currently an idea being explored by a number of organisations. STC Idec Ltd., for example, has developed an expert consultation system for diagnosing faults in wave soldering, and is now seeking to incorporate this system in a tutorial system for training personnel in the diagnosis of soldering defects.

However, there is some earlier research along these lines which suggests that merely imbedding an expert system in a tutorial system may not in itself be enough to improve the tutoring performance of the overall system. For example, the GUIDON system (Clancey, 1982) was based on the notion of transferring the expertise of the MYCIN system to the student. The experience suggested that while "MYCIN-like rule-based expert systems constitute a good basis for tutorial programmes", nevertheless "they are not sufficient in themselves for making knowledge accessible to the student" (Clancey, 1982, page 202). We have already touched on one important reason why this might be so, namely that *explanation* is at least as important as domain expertise in tutorial systems, and that the knowledge representation methods adopted in functional expert systems may not lend themselves to the provision of explanations that make sense to novices.

(ii) More robust user interfaces. There are two aspects to this - one trivial, the other more profound. The trivial one is that some conventional CBT have user interfaces which are both fragile and hostile. They are *fragile* in the sense that they cannot handle even spelling mistakes; they could be much improved simply by borrowing some of the methods AI researchers have developed for handling user input in a creative and helpful way.

At a more profound level, CBT interfaces are *hostile* in that they are 'dumb' or 'unintelligent'. That is to say, they cannot handle natural language exchanges and tend to hide behind the strategy of constraining the student to menu-driven or multiple-choice interactions with the program. Neither in general can conventional systems recognise that an unexpected (i.e. unforseen by the courseware author) input might be relevant to the material being taught, and might even constitute a correct answer to a question.

While acknowledging that natural language is not always the most efficient way for a human being to interact with a machine, it is already the case that where the domain is fixed and the vocabulary used in the interaction is dominated by technical and other terms relevant to the domain, AI researchers can deliver impressively sophisticated language-understanding front-ends. The CBT industry may be able to benefit from this research.

(iii) Student modelling. This is a key area, and one intimately related to the question of knowledge representation discussed above. There is widespread agreement that maintaining a dynamically updated model of the student is essential if more responsive and intelligent tutoring systems are to be devised. Some conventional CBT systems do, in fact, attempt to maintain a student model of sorts - for example, by making branching decisions dependent not on the last set of answers given by a student, but on some weighted function of his past performance, or the performance of his group.

At the other extreme are the full-blooded AI attacks on the student modelling problem, with machine-induction algorithms attempting to induce hypotheses about the pupil's understanding of the material from observations of his or her behaviour. The problem of student modelling is an extremely intractable one, and no one - in the AI community or elsewhere - would claim to have solved it. But conventional CBT authors and designers ought at least to (i) acknowledge the importance of student modelling, and (ii) examine which of the existing approaches to modelling might be feasible within their own tutoring systems.

Contribution of AI to face-to-face training

Artificial Intelligence is not so much an academic discipline as a way of thinking about problem-solving and computing. AI, in this sense, is really *a distinctive cast of mind* - one which sees knowledge and its representation in computable form as the key to problem-solving. Although AI research would appear to have something directly to offer computer-based training, there are reasons for thinking that taking an AI view of conventional training problems might also yield some insights.

As an illustration, consider the emphasis conventionally placed on behavioural approaches to training. This is fine so long as the performance expected of the trainees is in activities which are well-designed and constrained - as they are, for example, in manufacturing industry. But for more open-ended tasks and work settings, different approaches may be needed. Thus, effective training of salesman and saleswomen cannot be done merely by decomposing the selling activity into a large number of small tasks - opening, presenting, negotiating, closing, etc. - and teaching each of them using behavioural methods. Such an approach may be useful for beginners, but for more advanced training it is inappropriate.

A different approach is immediately suggested by looking at this training problem from the perspective of a knowledge engineer. This would emphasise the importance of eliciting from skilled sales personnel knowledge about how they work. Good salespeople do have some general knowledge about such things as opening, probing, presenting, closing, and so forth - all the things conventionally taught in sales training courses. But a more important kind of knowledge which they have is very specific knowledge about the right kinds of questions to ask particular types of customer to elicit the kinds of responses they can use to continue their sales presentation. They know particular ways to contact particular types of customer. They know very detailed things about their products (and about competitors' products) that are important to certain types of customer. In other words, good salespeople are like experts in every walk of life: they have worked with a particular domain over an extended period of time, and in the process have picked up a large number of practical heuristics.

When one asks what kind of sales training is needed in order to produce such individuals, the answer is that they are programmes which start by eliciting from skilled practitioners the heuristics which they habitually (perhaps unconsciously) use, and then structure these into teaching material which can be delivered in conventional forms.

This is just a single example to illustrate a more general point. This is that training - even of the conventional face-to-face kind - is ultimately a process in which specialist knowledge of some kind is extracted from some source, structured into training material and delivered to trainees. This process is known to be inefficient, partly because of the mystery surrounding how people learn, but also partly because the elicitation and structuring of the material is often deficient. The tools which AI researchers have developed for knowledge elicitation and representation can almost certainly improve the process.

Chapter Seven

CONCLUSIONS AND RECOMMENDATIONS

The main findings of this Report are:

1. The AI revolution is not just around the corner. Though there will be an increasing 'trickle-down' effect resulting in AI concepts, tools, approaches, etc. finding their way into software products (e.g. heuristic search in databases, natural language interfaces, intelligent front ends and so on), the fundamental problems which impede the development of generally intelligent machines will not be solved within the time frame of this report. This conclusion is reinforced by the fact that relatively little 'pure' or fundamental research is currently being done in AI.

The emerging market for AI products will, like other markets, be determined by the interplay of demand and supply factors. At the moment, the demand is small (some would say negligible) partly because awareness of AI research is still slight in many sectors of industry. Likewise, the history of previous tendencies to oversell AI research induces caution even in those who are aware of what's happening in AI labs. On the supply side, a critical factor would seem to be a chronic shortage of skilled AI practitioners - a shortage which is already chronic and likely to get worse.

2. Contrary to popular belief, there is much more to AI than expert systems. There is, for example, knowledge representation and elicitation, inference, search, learning, planning, game-playing, vision and language understanding. Moreover, the non-IKBS parts of AI research may be at least as valuable to the training community as anything which emerges from applied work in expert systems..

3. Expert systems are nevertheless important because potentially they offer a way of enabling computer-based tutoring systems to do what they purport to teach. In that sense, such systems offer a direct way of remedying one of the worst defects of conventional CBT. *However*, simply imbedding expert systems in CBT is not automatically an improvement in itself (as the GUIDON research showed) for if the system cannot explain its reasoning in terms that are meaningful to a student, then it isn't much good as a tutor.

4. The insights provided by AI research into a range of problems - learning, knowledge articulation and representation - could potentially be useful in conventional (non computer-based) training. Applying a knowledge engineering approach to eliciting the specialist knowledge to be taught, for example, may lead to more efficient and better structured training programmes.

5. The most obvious application of AI research in the computer-based training and computer-assisted learning fields would be the construction of so-called 'intelligent tutoring systems'. However, fully-fledged implementations of such systems do not appear to be cost-effective at the present time, though that will undoubtedly change in due course.

The main effect of AI research on CBT will be inenhancing that technology in specific ways and on various levels. For example:

* improving the delivery of CBT material to the student by means of

- more robust and intelligent user interfaces

- having domain knowledge included in program and all knowledge represented explicitly

- student modelling leading to more sensitive and adaptive teaching

- variable teaching styles

* more sophisticated authoring environments with software and hardware tools comparable with those currently available in AI labs.

* generative CBT - i.e. awareness on the part of the program of the content of its knowledge which enables it to *generate* some of its own assessment material and adjust the content and style of its presentation of teaching material. Among other things, this would reduce the responsibility of the conventional courseware author to try and forsee all possible interactions between the system and the student.

Recommendations

It is recommended that the Manpower Services Commission should:

1. Support projects aimed at providing specific enhancements to conventional CBT systems using AI ideas, techniques, and models e.g.

- imbedded expert systems
- more robust user interfaces
- student modelling
- explicit representation of domain knowledge.

2. Support AI-type analysis of training problems and materials - for example, a knowledge-engineering approach to the structuring of material to be taught in a training course.

3. Ensure that all MSC-supported projects should be feasible with existing AI tools - i.e. that none should require fundamental research in AI but should take and use existing software and hardware products 'off the shelf'.

4. Ensure that all supported projects should be ones for which there is a proven existing need. Typical examples might be skills transfer from one word-processing package to another, learning new safety procedures, training for introduction of new manufacturing technology, etc.

5. Sponsor expert systems development projects in specific training-related fields - for example, training needs analysis, choice of suitable domains for CBT, etc.

6. Encourage a more professional approach to courseware development within the CBT industry by emphasising the importance of proper tools, maintenance and long-term investment - in other words, hasten the end of the 'garage programming' metaphor in CBT authoring.

7. Sponsor on-going awareness programmes about AI in training.

8. Support the development of delivery vehicles for AI-enhanced CBT.

REFERENCES

Bonnet, Alain: *Artificial Intelligence - Promise and Performance,* Prentice-Hall, 1985.

Brown, J.S., Burton, R.R. and de Kleer, J.: "Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III", in Sleeman, D. and Brown, J.S. Eds.: *Intelligent Tutoring Systems*, Academic Press, 1982.

Clancey, W.J. : "Tutoring rules for guiding a case method dialogue", in Sleeman, D. and Brown, J.S. (Eds):*Intelligent Tutoring Systems*, Academic Press, 1982.

Clanon, Jeff: "Growing your own Artificial Intelligence Resources - Digital Equipment Corporation's Expert Systems Training and Apprentice Program", Digital Equipment Corporation, 1985.

Elsom-Cook, Mark: "Artificial Intelligence and Computer Assisted Instruction", Institute of Educational Technology, The Open University, 1986.

Frost and Sullivan Ltd, "Expert Systems in Europe", Report Number E711, February 1985.

Gamack, J. and Young, R.M.: "Psychological Techniques for Eliciting Expert Knowledge", *Proceedings of the Fourth Technical Conference of the British Computer Society Specialist Group on Expert Systems*, Warwick, December 18-20, 1984.

Harmon, Paul and King, David: *Expert Systems - Artificial Intelligence in Business*, Wiley, 1985.

Kidd, Alison: "What do users ask ? - Some thoughts on Diagnostic Systems" in Merry, Martin, Ed.: *Expert Systems '85*, Cambridge University Press, 1985.

Kimball, R.B. : *Self-Optimising Computer-Assisted Tutoring - theory and practice,* Technical Report (Psychology and Education Series), Institute for Mathematical Studies in the Social Sciences, Stanford University, June 1973.

Laurillard, Diana: "Introducing Computer-based Learning", *Open Learning*, Vol 1, No 1, February 1986, 10-12.

McCorduck, Pamela: *Machines Who Think*, WH Freeman, 1979.

O'Shea, Tim: "Intelligent Systems in Education", *Infotech State of the Art Report,* Series 9, Number 3, Pergamon Infotech Ltd., 1981.

Schon, Donald: *The Reflective Practitioner,* Temple Smith, 1983.

Scown, Susan, J.: *The Artificial Intelligence Experience - an introduction,* Digital Equipment Corporation, 1985.

Smallwood, R.D.: *A Decision Structure for Teaching Machines*, MIT Press, 1962.

Waterman, Donald A.: *A Guide to Expert Systems,* Addison Wesley, 1985.

Wiig, Karl M.: "Building Knowledge-Based Systems for the Financial Community -Experiences and Lessons Learned", *Fifth Technical Conference of the British Computer Society Specialist Group on Expert Systems*, Warwick, December 17-19, 1985.

Woods, P. and Hartley, J.R.: "Some learning models for arithmetic tasks and their use in computer-based learning", *British Journal of Educational Psychology*, 41 (1), 1971, pages 35-48.