Report Series
No. 37

Expert Systems

September 1983

The Butler Cox Foundation

THE BUTLER COX FOUNDATION

REPORT SERIES NO. 37

EXPERT SYSTEMS

ISSUED SEPTEMBER 1983

Abstract

Expert systems are computer systems that help in tackling difficult decision-making problems. As well as using facts, they attempt to embody judgement, rules of thumb and human experience, and they go through a rudimentary form of reasoning to offer possible solutions. Most pioneering work in expert systems has been done in universities in the United States; now the subject has been taken up by industry there and active groups have been established in the United Kingdom and continental Europe.

The purpose of this report is to provide Foundation members with a basic understanding of expert systems. It attempts to remove the myth attached to expert systems by giving a simple explanation of their main features, by placing them in context, and by indicating their potential applications.

The report concludes that expert systems represent a new software technique, still at a primitive state of development. These systems will not revolutionise data processing during the next five years.

Research team

The report was researched and written by a team of three Butler Cox consultants who have taken an active interest in the expert systems field over recent years:

Charles Chang: a consultant specialising in the strategic planning of information systems, with an emphasis on data management, high productivity tools for systems development and end-user computing.

Yair Melamud: a consultant specialising in information and logistics management. He has carried out extensive research into expert systems and other aspects of artificial intelligence.

David Seabrook: a consultant with extensive experience of systems development. He has been involved in the supervision of numerous Foundation reports covering many aspects of information technology, and has carried out research in the expert systems field.

Photoset and printed in England by Flexiprint Ltd., Worthing, Sussex.

THE BUTLER COX FOUNDATION

Butler Cox & Partners
Butler Cox is an independent management consultancy and research organisation, specialising in the application of information technology within commerce, government and industry. The company offers a wide range of services both to suppliers and users of this technology. The Butler Cox Foundation is a service operated by Butler Cox on behalf of subscribing members.

Objectives of The Foundation
The Butler Cox Foundation sets out to study on behalf of subscribing members the opportunities and possible threats arising from developments in the field of information systems.

The Foundation not only provides access to an extensive and coherent programme of continuous research, it also provides an opportunity for widespread exchange of experience and views between its members.

Membership of The Foundation
The majority of organisations participating in the Butler Cox Foundation are large organisations seeking to exploit to the full the most recent developments in information systems technology. An important minority of the membership is formed by suppliers of the technology. The membership is international with participants from Belgium, Denmark, France, Italy, the Netherlands, Sweden, Switzerland, the United Kingdom and elsewhere.

The Foundation research programme
The research programme is planned jointly by Butler Cox and by the member organisations. Half of the research topics are selected by Butler Cox and half by preferences expressed by the membership. Each year a short list of topics is circulated for consideration by the members. Member organisations rank the topics according to their own requirements and as a result of this process, members' preferences are determined.

Before each research project starts there is a further opportunity for members to influence the direction of the research. A detailed description of the project defining its scope and the issues to be addressed is sent to all members for comment.

The report series
The Foundation publishes six reports each year. The reports are intended to be read primarily by senior and middle managers who are concerned with the planning of information systems. They are, however, written in a style that makes them suitable to be read both by line managers and functional managers. The reports concentrate on defining key management issues and on offering advice and guidance on how and when to address those issues.

Additional report copies
Normally members receive three copies of each report as it is published. Additional copies of this or any previous report (except those that have been superseded) may be purchased from Butler Cox.

Previous reports
No. 1 Developments in Data Networks
No. 2 Display Word Processors*
No. 3 Terminal Compatibility*
No. 4 Trends in Office Automation Technologies
No. 5 The Convergence of Technologies
No. 6 Viewdata*
No. 7 Public Data Services
No. 8 Project Management
No. 9 The Selection of a Computerised PABX
No. 10 Public On-line Information Retrieval Services*
No. 11 Improving Systems' Productivity
No. 12 Trends in Database Management Systems
No. 13 The Trends in Data Processing Costs
No. 14 The Changing Equipment Market
No. 15 Management Services and the Microprocessor
No. 16 The Role of the Mainframe Computer in the 1980s
No. 17 Electronic Mail
No. 18 Distributed Processing: Management Issues
No. 19 Office Systems Strategy
No. 20 The Interface Between People and Equipment
No. 21 Corporate Communications Networks
No. 22 Applications Packages
No. 23 Communicating Terminals
No. 24 Investment in Systems
No. 25 System Development Methods
No. 26 Trends in Voice Communication Systems
No. 27 Developments in Videotex
No. 28 User Experience with Data Networks
No. 29 Implementing Office Systems
No. 30 End-User Computing
No. 31 A Director's Guide to Information Technology
No. 32 Data Management
No. 33 Managing Operational Computer Services
No. 34 Strategic Systems Planning
No. 35 Multifunction Equipment
No. 36 Cost-effective Systems Development and Maintenance

**These reports have been superseded.*

Future reports
No. 38 Selecting Local Network Facilities
No. 39 Trends in Information Technology
No. 40 New Ways in Presenting Information

EXPERT SYSTEMS

CONTENTS

EXPERT SYSTEMS

REPORT SYNOPSIS

No topic in computing is more confusing to management at the present time than that of "intelligent knowledge-based systems" in general and expert systems in particular. Despite (or, perhaps, because of) the space accorded the subject in the computing press, the present achievements and future significance of these systems is far from clear. We know that these and other techniques of artificial intelligence (AI) may well emerge in Japanese fifth generation systems in perhaps ten years' time — but what do they mean today for organisations looking ahead over the next few years?

Expert systems are computer systems that help in tackling difficult decision-making problems. As well as using facts, they attempt to embody judgement, rules of thumb and human experience, and they go through a rudimentary form of reasoning to offer possible solutions. Most pioneering work in expert systems has been done in universities in the United States; now the subject has been taken up by industry there and active groups have been established in the United Kingdom and continental Europe.

Introducing the subject in this report (chapter 1), we define an expert system as a computer system containing organised knowledge, both factual and heuristic, that concerns some specific area of human expertise; and that is able to produce inferences for the user. Among early milestones in AI research that are relevant to expert systems were the development of the LISP symbol-processing language by John McCarthy at Massachusetts Institute of Technology; and the application of AI techniques to discrete problems in chemistry and medicine by the heuristic programming team led by Edward Feigenbaum at Stanford University.

One key difference between a conventional computing system and an expert system is that the expert system uses inexact reasoning. It holds facts and heuristics (and normally also the processing logic) in a knowledge base, which is interpreted by a separate reasoning mechanism or inference engine. Two of the most widely used symbolic languages are LISP, much favoured in the United States; and PROLOG, a higher-level language based on predicate logic and developed in France and the United Kingdom.

In a total of 41 expert systems listed in chapter 2 of the report, only one has had a genuine commercial impact. This is R1, developed jointly by Carnegie-Mellon University and Digital Equipment Corporation for the purpose of configuring the DEC VAX 780 computer system. Several other expert systems are notable technical successes. We describe four state-of-the-art examples: Stanford's DENDRAL and MYCIN, SRI International's PROSPECTOR, and DEC's R1. Our examination of the current state of the technology includes also eleven case studies (see chapter 3), of which five come from the United States, four from the United Kingdom and two from France.

Experience to date points to a number of limitations in present-day expert systems:

- The area of knowledge (the domain) which any one system can handle is small and specialised.
- The systems take many years to construct.
- They are large and expensive to construct.
- Only a few domains have been tackled, and these have been those where an industry sector or government agencies can afford the high risk.
- In the domains that have been tackled, the human experts are scarce and expensive.
- The ability of the systems to explain their reasoning is relatively poor.
- Most present-day systems have scant built-in knowledge of their own assumptions, and so can be used only by experts.

Suitable applications for expert systems, in our view, will be found in the broad areas of training, advice, and intelligent interfaces. In training, knowledge-based systems have an obvious contribution to make to computer-aided training in areas where the knowledge is highly specialised and difficult to acquire.

Providing advice is a very broad field. Fault diagnosis, insurance broking, tax guidance and general counselling based on regulations are among the suitable cases for expert systems, which also could strengthen the power of computer-based decision support systems.

Intelligent interfaces could be provided to existing databases and software. Using complex software is a very knowledge-intensive activity, and manuals are complex and often poorly written. Expert systems could giver users a much better understanding and would have a broad market in this field.

A number of other special fields offer opportunities for the introduction of expert-system techniques, notably those where traditional techniques have proved inadequate. Examples include voice recognition and natural language processing.

A basic problem in building expert systems is the difficulty of acquiring the information necessary to structure the knowledge base. Substantial resources are needed in terms of staff — domain experts with the time and willingness to participate, knowledge engineers skilled in extracting the domain expertise and presenting it to the system, and skilled computer systems designers who can think in terms of the user interface. Maintaining a large knowledge base will be just as difficult as constructing it; by definition the knowledge base will change as new knowledge emerges.

Specialist hardware and software resources also will be needed, pushing the total cost of developing a typical expert system to perhaps one million dollars or more.

Looking forward from today's narrow and specialised expert systems to a future when expert systems may become more generally useful and cost-effective, a number of improvements are called for (see chapter 5). Acquisition of knowledge and understanding of natural languages must be improved, together with the understanding of how to represent knowledge. We need better understanding of how to deal with uncertainty; and of the processes of human judgement, reasoning and perception. Better ways of identifying domains, more trained and skilled people, and lower-cost hardware are also needed. Only the last of these seems certain to be attained.

Over the next two or three years, therefore, the main existing limitations will remain. Large, expensive and high-risk expert systems will be tailored to the specialised needs of the few companies that can afford them. In parallel, small experimental systems will be developed whose value will be mainly educational. A substantial investment will be needed to achieve significant benefits.

We believe that Foundation members should consider expert systems applications only in cases where knowledge is already available in some written form; where the application area calls for continuous update of logic rules; where the system can be developed in a modular way; where there is a clear incentive for a user to use the system; and where a user is going to be able to maintain and improve the knowledge base.

Expert systems, we conclude, will not revolutionise data processing during the next five years. They represent a new software technique, still at a primitive state of development.

PREFACE

Expert systems are computer systems that help with important and difficult decision-making problems which only a few experts do well. Expert systems are not the same as conventional computer systems. As well as using factual information these systems attempt to embody judgement, rules of thumb and experience of experts in the field. A reasoning process is then used to provide a possible solution to a problem.

Work on expert systems has been underway for the past 15 years. Research grew out of the specialised field of artificial intelligence, which has itself been actively pursued over the past 25 years as a part of computer science.

Despite its comparatively lengthy heritage, the subject of expert systems has only recently attracted general interest amongst the commercial computing fraternity. That interest stems from a number of important developments over the past two or three years. In particular, expert systems have recently been shown to equal or even better human experts' performance in some fields; cheaper hardware means that expert systems application could become economic; and the widely publicised Japanese plan to overtake IBM as the world's leading computer force has generated much interest.

At a conference held in Tokyo in October 1981, the Japanese announced their plans to research and develop a so-called fifth generation computer architecture, in which expert systems would have an important part to play. Since that time, interest in expert systems has grown, not only amongst an expanding circle of specialists, but also amongst management services staff who are anxious to know whether these systems present new opportunities for their organisations.

To many people in management services, the subject of expert systems is bewildering. The reasoning process itself is not an easy one to grasp. Jargon terms abound. Experienced staff are few and far between. Expert systems having the capacity to do jobs that are genuinely worthwhile seem very expensive. Nonetheless, more and more management services staff are looking for guidance on the subject. On the one hand they want to satisfy their curiosity. On the other, they want to know whether there are applications in their business where expert systems can be usefully exploited — either now, or in a few years time when the subject may have advanced beyond its present experimental stage.

Purpose of this report and intended readership

The purpose of this report is to provide Foundation members with a basic understanding of expert systems.

The report attempts to remove the myth attached to expert systems by giving a simple explanation of their main features, by placing them in context, and by indicating their potential applications.

Because expert systems have received extensive (and sometimes misleading) coverage in the computing press, the report should be made available to anyone in the information systems function with an interest in the subject. A formal background in computer science is not a prerequisite.

Scope and structure of the report

We have written the report with the business user in mind, and its emphasis is on the commercial implications of expert systems. It is not a technical report and so does not cover in detail the technical issues associated with expert systems. We have covered these issues in a general and simplistic way. There is a danger here that the reader may underestimate the technical difficulties involved in building expert systems. These difficulties cannot be overstated.

We begin in chapter 1 by giving a general introduction, placing expert systems in context and explaining the reasons for recent developments.

Chapter 2 then outlines the main features of expert systems, gives a summary of principal expert systems developed by 1983 and describes in detail four major systems that represent the current state of the art.

Next, in chapter 3, we review experience of users with expert sytems and look at recent development work by major suppliers of expert systems.

In chapter 4 we discuss potential application areas. The contents of this chapter can be used to help identify potential expert system applications and to estimate development costs.

In chapter 5 we look to the future and describe the developments that need to take place before expert systems can make a significant impact on the commercial world. We then assess the likelihood of these developments taking place in the next five years.

Finally in chapter 6 we provide a concise set of guidelines for companies wishing to explore the potential for expert systems.

The comprehensive glossary at the end of the report will help guide the reader through the maze of new jargon and terms.

For the reader who wishes to study the subject in depth, the bibliography provides a guide to current literature.

Approach to the research

Over the past year the computer press has given extensive coverage to expert systems. Some of this information was factual, some interpretive, but most of it contained views and opinions. We decided therefore to concentrate our research on discussions with individuals who had direct experience of expert systems development or use. Discussions were held with over 20 leading experts in this field, including those listed below:

Dr Gerald Barber, Institut National de Recherche En Informatique et Automatisme (INRIA), B.P 150 78150, Rocquencourt ,France. Telephone: 3331 954 9021

Dr Mike Barratt, SPL International Research Centre, The Charter, Abingdon, Oxfordshire, England. Telephone: 0235 24112

Dr Richard Duda, Head of Expert Systems Fairchild Camera and Instrument Corporation, 4001 Miranda Avenue, Palo Alto, USA. Telephone: 415 857 1501

Dr John Fox, Imperial Cancer Research Fund, PO Box 123, Lincoln's Inn Fields, London, England. Telephone: 01 242 0200

Alex Goodall, Expert Systems Ltd, 9 West Way, Botley, Oxford, England. Telephone: 0865 242206

David C Hawkins, Racal Expert Systems, 21 Jubilee Way, Chessington, Surrey, England. Telephone: 01 397 5281

Brian Johnson, BL Advanced Systems Technology, Grosvenor House, Prospect Hill, Redditch, Worcs., England. Telephone: 0527 64274

Dr Karen Sparck Jones, Computer Laboratory, University of Cambridge, Corn Exchange Street, Cambridge, England. Telephone: 0223 352435

Peter Jones, Tymshare UK, Brettenham House, Lancaster Place, London, England. Telephone: 01 379 7822

Dr S Jerrold Kaplan, Vice President, Business Development, Teknowledge Inc, 525 University Avenue, Palo Alto, USA. Telephone: 415 328 4870

Dr T Kehler, Vice President and Director of Applied Artificial Intelligence, IntelliGenetics, 124 University Avenue, Suite 300, Palo Alto, USA. Telephone: 415 493 7250

Professor Frank Land, Department of Computer Science, London School of Economics, England. Telephone: 01 405 7686

Professor John McDermott, Department of Computer Science, Carnegie-Mellon University, Pittsburg, Pennsylvania 15213, USA. Telephone: 412 578 2000

Professor Donald Michie, Machine Intelligence Research Unit, University of Edinburgh, Hope Park Square, Meadow Lane, Edinburgh, Scotland. Telephone: 031 667 1011

Steve Owsianka, CCTA, Riverwalk House, 157-161 Millbank, London, England. Telephone: 01 211 5940

Barry Parker, ISIS Systems Ltd, 11 Oakdene Road, Redhill, Surrey, England. Telephone: 0737 71327

Sal Pinto, Mars Group Services, 132-133 Fairlie Road, Slough, Berkshire, England. Telephone: 0753 30721

Gerry Piper, International Computers Limited, Fairview Road, Stevenage, Herts., England. Telephone: 0438 56111

Gordon Scarrott, Independent consultant, 34 Parkway, Welwyn Garden City, Herts., England. Telephone: 070 73 23073

Dr Read Smith, Schlumberger-Doll Research, Old Quarry Road, Ridgefield, Connecticut, USA. Telephone: 203 431 5508

Dr Mark Stefik, Palo Alto Research Centre, Xerox Corporation, 3333 Coyote Hill Road, Palo Alto, USA. Telephone: 415 494 4012

Professor Lofti Zadeh, Computer Science Division, University of California, Berkeley, California 94720, USA. Telephone: 415 642 6000

Professor Gian Piero Zarri, Centre National de la Recherche Scientifique (CNRS), Laboratorie Informatique pour le sciences de l'Homme, 54 Boulevard Raspail, 75270 Paris, France. Telephone: 331 544 3849

CHAPTER 1

INTRODUCTION

In this chapter we begin by establishing the historical and technical background of expert systems, placing them in the context of artificial intelligence. Next we describe what an expert system is, looking at the same time at two closely related subjects: knowledge engineering and heuristics. We then examine a topic which is fundamental to expert systems — that of inexact reasoning — introducing terms such as Bayesian probability, certainty factors and fuzzy logic. The discussion leads from a general explanation of the main components to a description of their role in fifth-generation computer systems.

ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS

Artificial intelligence is concerned with enabling computers to mimic the characteristics that make people seem intelligent. That statement raises the question of what is meant by human intelligence. Attempting to define human intelligence is of doubtful value, at least in the context of this report, because of its difficulty. What we can say is that intelligence appears to be an amalgam of many different information-processing and information-representing capabilities. (Information itself, of course, is communicated knowledge.) Intelligence includes many abilities — to reason, to infer, to theorise, to prove, to acquire knowledge, to apply knowledge, to pursue, to communicate ideas, to learn, and finally to teach.

The central goals of research into artificial intelligence are to make computers more intelligent, and so more useful, and to understand the principles which make intelligence possible.

Artificial intelligence began to become an active field of research within computer science (or possibly halfway between computer science and psychology) in about 1955. Since that time, the study of artificial intelligence has embraced a wide range of topics, including problem solving, theorem proving, game playing, pattern recognition, search methods, heuristics, linguistics (syntax and semantics), learning and teaching.

Early on, researchers in artificial intelligence came to the conclusion that traditional mathematical techniques would not be suitable for their work. Richard Duda, head of expert systems at Fairchild Camera and Instruments Corporation, when interviewed by our researcher said:

"There are very strong limits on what numerical methods can do. It is just unthinkable to use an operational research method for recognising continuous speech, it is just not appropriate."

Symbol systems, which manipulate collections of symbolic structures, were considered to be more suitable for encoding intelligence-exhibiting processes.

One direct result of this was the development in the early 1960s by John McCarthy of Massachusetts Institute of Technology (MIT) of the LISP language. LISP is a computer language designed for manipulating symbolic expressions in a recursive way. It enables researchers to encode and to explore intelligence-exhibiting processes. Since the 1960s LISP has gone on to become the prime language of artificial intelligence in North America. It is used in the implementation of many expert systems.

Artificial intelligence research was almost totally exploratory in the early years. Researchers tried to explore possibilities rather than produce results. Because of the open-ended work on seemingly intangible problems, artificial intelligence became isolated and unpopular within the computer science fraternity. But as years went by researchers came under increasing pressure to deliver practical results. The pressure came both from within the academic world and from outside sponsors (such as the United States government).

In the mid-1960s, a new school of thought arose within artificial intelligence, led by Edward Feigenbaum. Feigenbaum suggested several reasons why artificial intelligence was not making reasonable progress. The problems being addressed were too large and too vague, and they involved too many unknowns and too many interactions. Feigenbaum believed that a more useful application of artificial intelligence techniques would be to specific problems. His view was that a carefully chosen specific problem would provide sufficient complexity to make research meaningful and interesting. Furthermore, the development of new ideas or techniques would be applicable to other specific problems. In time, the specific solutions and techniques could find applicability to more general problems.

Feigenbaum's approach, to reduce problems to a manageable size, did not immediately generate enthusiasm within the artificial intelligence community. His work was regarded as uninteresting by many researchers in this field. But the Stanford Heuristic Programming Project (HPP), led by Feigenbaum, was responsible for the development of a significant number of expert systems, and has probably contributed more than any other single project to the credibility of artificial intelligence. The systems developed by the HPP include DENDRAL, MYCIN, MOLGEN, PUFF and UNITS, all of which are described in chapter 2.

At Stanford University most of the work was carried out in narrow well-defined areas of medicine and chemistry. Among the reasons for this were that most of the information was available in publications; the relevant experts were often university staff; and public financing was available.

At Carnegie-Mellon University, development work was more commercially orientated. With collaboration and financial support from Digital Equipment Corporation, the R1 system was developed (a detailed description is given on page 19). A pragmatic approach was used to avoid some of the more difficult problems associated with expert systems.

The development of the PROLOG programming language is the most exciting innovation to come from Europe. It is based on work done by R Kowalski, interpreting what is known as "Horn clause predicate logic", although the language concept was first developed and implemented by A Colmeraner's research group at the University of Aix in Marseilles. PROLOG is described on page 13.

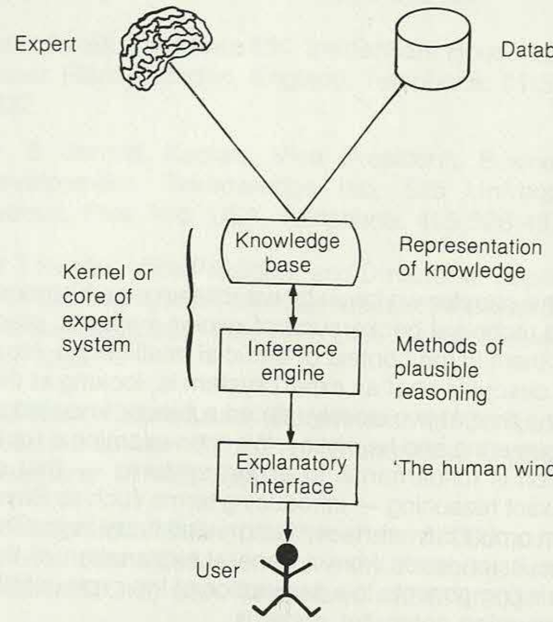
According to Buchanan (see reference 1), all research into artificial intelligence is relevant to the understanding and constructing of expert systems. His view is that expert systems will continue to be severely constrained until we understand better how to represent many concepts that have been central to artificial intelligence research for over 20 years. This may not be achieved in the next 20 years.

WHAT IS AN EXPERT SYSTEM?

It is a characteristic of immature fields of scientific research that the specialists are unable to agree on a definition of their chosen field. The study of expert systems is no exception. A typical expert system is illustrated in Figure 1.1.

This figure illustrates how expert knowledge is acquired from two sources: human experts, and databases of facts and figures (such as text books, reference books and handbooks). Needless to say,

Figure 1.1 Components of an expert system



(Source: Forsyth, R., Expert Systems Now, Hexadecimal Press Seminar, June 1983)

the quality and usefulness of an expert system is directly proportional to the quality and the organisation of the knowledge acquired. Figure 1.1 also illustrates three key components of an expert system: the knowledge base, inference engine and explanatory interface.

Definition of expert systems

For the purposes of this report we have adopted the following working definition:

"An expert system is a computer system containing organised knowledge, both factual and heuristic, that concerns some specific area of human expertise; and that is able to produce inferences for the user."

It is instructive to compare our definition with three others, which are typical of the many definitions available. The first has been proposed by the British Computer Society's Specialist Group on the subject: "an expert system is regarded as the embodiment within a computer of a knowledge-based component, from an expert skill, in such a form that the system can offer intelligent advice or take an intelligent decision about a processing function. A desirable additional characteristic, which many would consider fundamental, is the capability of the system, on demand, to justify its own line of reasoning in a manner directly intelligible to the enquirer. The style adopted to attain these characteristics is rule-based programming."

The second typical definition is due to Bramer (see reference 2): "an expert system is a computing system which embodies organised knowledge concerning some specific area of human expertise, sufficient to perform as a skilful and cost-effective consultant."

The third typical definition has been noted by Jones (see reference 3): "an intelligent knowledge-based system (of which expert systems are a sub-class) is a system for carrying out a single task, but a task of sufficient complexity to imply working with large, incomplete, uncertain or rapidly changing knowledge; with tentative inference procedures for exploiting this knowledge in reacting to variegated and unreliable inputs."

These three typical definitions have the merit of being somewhat more explicit than our own, but also the disadvantage of being rather more unwieldy.

DEFINITION OF TERMS

There are a number of terms associated with expert systems that appear continually in the press and in other publications. In the main these terms have originated from within the world of artificial intelligence. It is helpful to define them at this early stage. We will discuss some of these terms in more detail later in the report. (A full list of terms and definitions appears in the glossary at the end of the report.)

Backward/forward chaining

These terms are used to describe alternative control strategies used by the reasoning mechanism (inference engine). In forward chaining, the program starts by satisfying a set of conditions, then moves forward towards some (possibly remote) conclusion. In backward chaining, the program starts by assuming a conclusion or goal, then works backward trying to satisfy all the conditions leading to that goal.

Bayesian probability

A probability theory exploiting the elementary theorem known as Bayes' rule. This rule establishes a numerical relationship between a hypothesis and observed evidence.

Empty shell

This term (which has its roots in Empty MYCIN, EMYCIN) describes a generalised expert system package emptied of its knowledge. It provides a structure or framework for a designer to build a new knowledge base. The empty shell also provides the inference mechanism, together with its predetermined control strategy.

Fuzzy logic

A method for handling inexact information by attempting to quantify non-numeric (value) judgements. Fuzzy logic deals with the rules of manipulating fuzzy sets which are sets of values corresponding to a logical statement called a fuzzy proposition, for example:

The logical statement: X is a large number could correspond to the fuzzy set:

($X\varepsilon$ (0,10), .1) — the probability of x being a value between 0 and 10 is .1

($X\varepsilon$ (10,1000), .2) — the probability of x being a value between 10 and 1000 is .2

($X>$ 1000, .7) — the probability of x being larger than 1000 is .7.

Heuristics

The term heuristics describes the informal, judgemental knowledge of an application area that constitutes the "rules of good judgement" in the field. Heuristics also encompass the knowledge of how to solve problems efficiently and effectively, how to plan steps in solving a complex problem, how to improve performance, and so forth. This type of knowledge has typically been accumulated by experts in the field and represents their experience.

Horn clause sub-set of predicate logic

Horn clause sub-set of predicate logic is used as a basis for the PROLOG logic programming language. Horn clauses, named after Alfred Horn, are used to express information in a way that can be used to solve problems. A Horn clause sentence is either a simple assertion, such as "John likes Mary", or an implication such as "Mary likes X if X likes Mary" (Mary likes anyone or anything who likes her.)

Inference engine

The problem-solving algorithm, or rule interpreter, and its method of applying to the problem the relevant knowledge in the knowledge base.

Knowledge base

A database of knowledge in which both facts and heuristics are represented as individual elements of knowledge about a particular field (domain).

Knowledge engineering

Knowledge engineering is the process of building a specific expert system by assembling the requisite knowledge. The process is concerned with representing knowledge in such a way that it can be used by a system, and be meaningful to a user. Knowledge engineering is also concerned with acquiring and testing knowledge to ensure that it is, in the context of a given problem, both internally consistent and complete.

List processing language

Used to manipulate strings, rather than characters, and to manage their storage.

Pattern matching

The process of matching the conditions that the program needs to satisfy with the data available in the database (or supplied interactively by the user).

Predicate calculus

A widely studied formal language of symbol structures. Some of its concepts are relevant to symbolic computing and are used for defining structures and the relationships between things. Predicate calculus also allows for functions and logical connections.

Production rule

A common approach to representing the domain knowledge needed for an expert system. Also called an IF-THEN rule or alternatively a situation-action rule. A production rule states that if a certain kind of situation arises, a certain action can be taken.

Symbol

A string of characters such as Apple, Table, Five, 3.14159, etc.

Symbol structure

A type of data structure containing symbols (also known as a list structure). A symbol structure can be used to represent information and is especially useful when the information is not numeric. For example:

(ON BLOCK 1 BLOCK 2) means the item defined as BLOCK 1 is on top of item BLOCK 2

(PLUS 5 S) means add 5 to S

(PART-OF E D) means E is part of D.

Expert systems compared with conventional systems

As our working definition indicates, an expert system is a computer system that enables a user to apply the knowledge of an expert (both factual and heuristic) in a narrow well-defined field, to a given problem. Some conventional computer systems, however, could be said to fit that description. One example is a financial forecasting system written in a financial modelling language by a financial specialist that can be used to make forecasts in the face of uncertainty. Yet the financial system is clearly not an expert system in the true sense. The differences between the two are set out in Figure 1.2.

A note of warning: The term expert system is sometimes applied to general artificial intelligent languages such as LISP and PROLOG. That is erroneous, and is equivalent in conventional computer systems to referring to COBOL as an application package.

INEXACT REASONING

One of the key differences between a conventional computer system (such as for payroll processing) and an expert system (such as MYCIN which is used in medical diagnosis) is that an expert system makes use of inexact reasoning. Inexact reasoning can be illustrated by MYCIN. If a blood test with the agent X indicates a positive reaction, then MYCIN concludes (and advises the user) that it is likely (with a .75 certainty) that the patient suffers from disease Y. Expert systems handle inexact reasoning in three

Figure 1.2 Differences between conventional systems and expert systems

Conventional systems	*Expert systems*
Contains orderly and deterministic processes	Contains heuristic and rule of thumb processes
A single input goes through a single mechanism or algorithm to produce a correct output	Multiple common (often redundant) inputs go through overlapping mechanisms to produce multiple plausible solutions
Program code and data are kept separately. The recipes for manipulation, and the structure of information, are intermixed in the code	The manipulation rules are held separately (in the inference engine) from the structure of information (in the knowledge base)
The embedded manipulation and structure makes it difficult to modify complex systems	A separate knowledge base can be amended relatively easily
No reasoning or explanation given to the user about why a particular input resulted in a particular output (except through a roundabout mechanism of program trace in debugging tools)	The knowledge base is intentionally made visible to the user

(Source: Quinlan, J. R., Fundamentals in the knowledge engineering problem, Introductory Readings in Expert Systems, D. Michie (editor) Gordon Breach Science Publications, 1982) main ways — by Bayesian probability, certainty factors and fuzzy logic.

Bayesian probability

Bayesian probability is a statistical approach to inexact reasoning. It is based on deriving probabilities of events from odds-in-favour of prior related events. So it depends on the availability of sufficient statistical data to calculate the odds in favour, and also on the independence of that data. Data from observations is collected and analysed to derive these probabilistic values. By way of an illustration, consider a medical diagnosis. The symptoms and laboratory tests of people suffering from certain diseases are recorded. From these records, the odds-in-favour of a particular disease occurring may be calculated for certain combinations of symptoms and test results. The calculated factors relating the observations to possible diseases can be built into the knowledge base. The end result is that, in the case of a single patient showing certain symptoms and test results, the conditional probability that the cause is disease X can be calculated.

The main difficulty with Bayesian probability is the large amount of data that is required to determine all the conditional probabilities. Further problems are raised by the need for assumptions about the independence of the observations.

Certainty factors

An alternative approach to inexact reasoning has been developed by Shortliffe at Stanford University. It is called the method of certainty factors, and it is this method that is used in MYCIN (see description in chapter 2). The principle behind the method of certainty factors is that experts provide an assessment of a 'belief' that a particular hypothesis is true, together with a separate 'disbelief' that the hypothesis is true. Both the belief and the disbelief factors are valued in the range 0.0 to 1.0. Surprisingly, the belief factor is not usually equal to 1.0 minus the disbelief factor, because of the nature of the conditional circumstances that are involved.

For example, a hypothesis involving certain symptoms and test results could cause the experts to record a belief factor of .95 that disease X is present. Essentially, this means that the experts are 95 per cent certain that those symptoms indicate disease X. The same experts, on the same occasion, could legitimately record a disbelief factor of 0.2 — indicating a 20 per cent certainty of some other disease. The certainty factor is simply computed as the difference between belief and disbelief: 0.95 minus 0.20 = 0.75. In other words, given the symptoms and test results, there is a 75 per cent chance that the patient has disease X.

A common objection to the method of certainty factors is that it has the formality of Bayesian probability, without that method's rigorous scientific and mathematical basis.

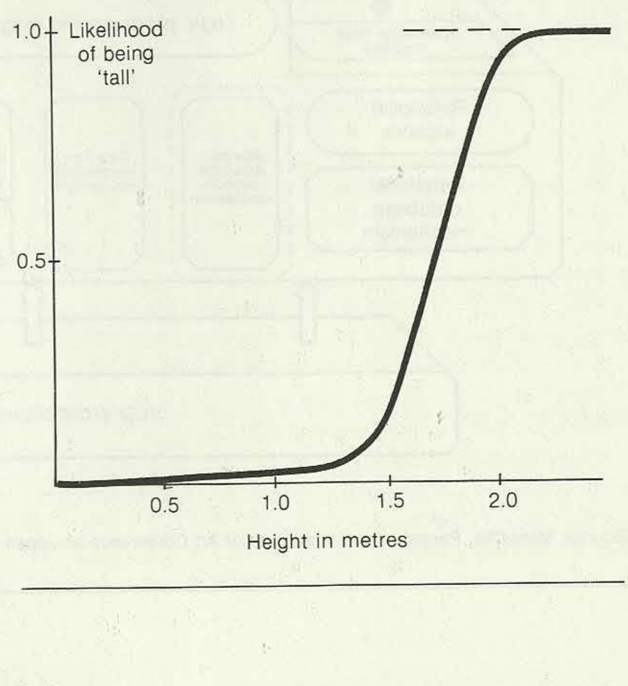
Fuzzy logic

Fuzzy logic was first developed by Lotfi Zadeh at the University of California at Berkeley in 1965. The principle behind fuzzy logic is that most human reasoning is not only inexact, but non-numerical as well. Many values are expressed as qualifications rather than quantifications. Thus a person may be generally agreed to be very tall, interest rates to be moderately high and so forth. Fuzzy logic enables such qualifications to be translated into quantifications for analysis and manipulation.

For instance, consider the case in which inferences need to be drawn depending on the human attribute of "tallness", but where the actual measurement in height is not readily available. First, an analysis is required to map between the height of people in a given population and a quantified measure of tallness. Of course, many non-numerical attributes are subject to national, cultural and other differences. Thus a tall Japanese may appear short to a Norwegian. Figure 1.3 shows a mapping function for tallness that is valid in Western Europe. There is a 67 per cent chance that a male adult 1.8m in height will be considered tall in Western Europe. Thus if our input to a system is "very tall European male" the system using its fuzzy set can correlate this qualification to a likely height in metres.

Zadeh now contends that no expert systems can be built without using fuzzy logic. Others disagree. On

Figure 1.3 Fuzzy logic: the likelihood of an adult male being regarded as tall in Western Europe

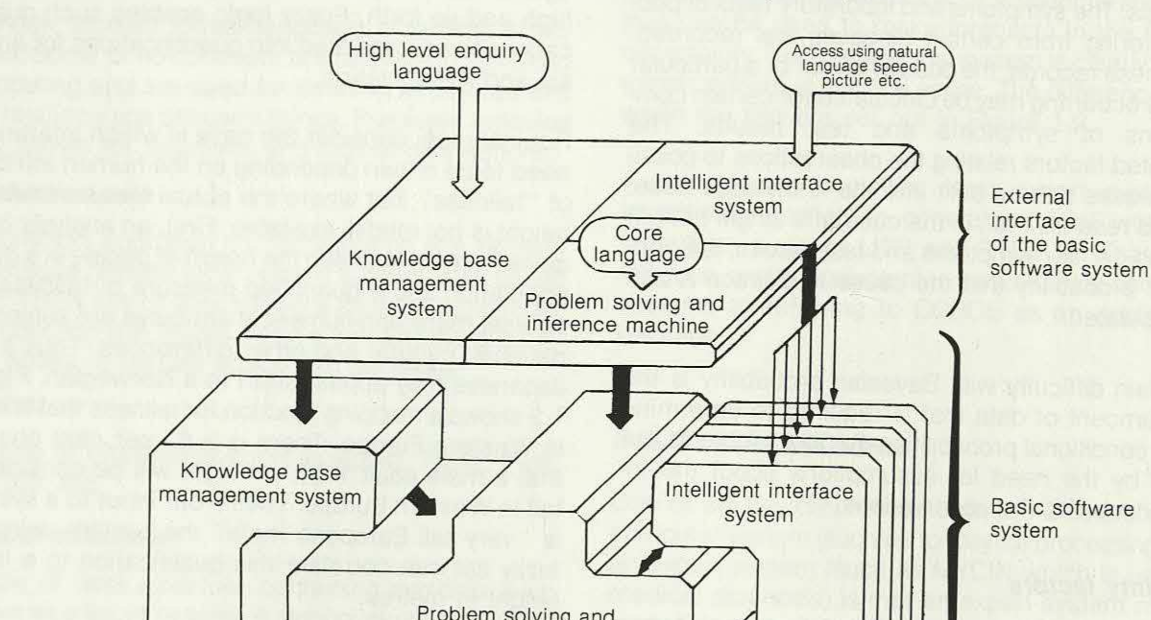


the one hand, they accept the need for encapsulating non-numerical reasoning. But at the same time they contend that a similarly viable and useful set of rules and results can be achieved through a combination of numerical processes (such as Bayesian probability or certainty factors) and iterative trial-and-error processes involving an expert from whom the necessary knowledge is acquired. Clearly inexact reasoning does not yet have a sound theoretical foundation.

JAPAN AND THE FIFTH GENERATION

In October 1981, the Japanese announced that they were embarking on a fifth-generation computer systems project. The project is planned to run for about 10 years and to result in commercially viable products by the mid-1990s. The logic of the term "fifth generation" is that it represents a significant advance over what has gone before.

Figure 1.4 Conceptual representation of the Japanese fifth generation computer system



(Source: Moto-Oka, Pergamon Infotech State of Art Conference on Japan and the Fifth Generation, September 1982)

Fifth-generation computing incorporates the principles of artificial intelligence since, in future, non-numeric data processing will play a more important role in information processing. Three years of study by the Japanese Government, industry and research organisations led to this very ambitious programme which aims to achieve worldwide leadership for Japanese industry in information technology. The approach has been to start almost with a blank sheet of paper and re-think the conventional computer design philosophy. This approach was adopted because the research and development targets of the programme cannot be handled within the framework of conventional computer systems.

In his presentation to the Pergamon state of the art conference on Japan and the fifth generation (see reference 4) Moto-Oka says that the functions of the project may be roughly classified as follows:

—Problem-solving and inference.

—Knowledge-base management.

—Intelligent interface.

These functions will be realised by making individual software and hardware systems correspond. The Japanese fifth-generation project plans to combine research on very large scale integration (VLSI), parallel processing, pattern recognition, logic programming and knowledge-based systems.

Figure 1.4 shows the basic conceptual structure of the Japanese fifth generation computer system.

Two years after its inception, the fifth generation project continues to attract attention from academia and industry, and from both the computer and the general business community. There is a difference of opinion over whether the aims and direction are realistic or misguided, and whether the Japanese call for international co-operation is genuine, or merely a means of acquiring expertise quickly and relatively inexpensively.

Without doubt, the project has generated new interest in and awareness of artificial intelligence systems. In turn, this interest has spurred efforts in North America and Europe to put more emphasis into researching and developing expert systems. This may well be beneficial for information technology and society.

CHAPTER 2

FEATURES AND EXAMPLES OF EXPERT SYSTEMS

Having introduced in chapter 1 the subjects of artificial intelligence and expert systems, we turn now to a more detailed discussion of expert systems. We begin by setting out the distinguishing characteristics of expert systems. Next we look at expert system structures, and at two main languages — LISP and PROLOG. Then we summarise the features of 41 principal systems that are in existence today, and describe four of them in detail.

CHARACTERISTICS OF EXPERT SYSTEMS

In chapter 1 we defined an expert system as a computing system embodying organised knowledge about some fields of human expertise in both heuristic and factual forms, and capable from that knowledge of producing useful inferences.

Expert systems differ from conventional computer programs in a variety of ways. A conventional program is an integral unit. The whole is determined analytically in advance. In contrast, the knowledge base of an expert system normally contains pieces of knowledge that are independent of each other. Each piece can be altered separately. This is an important, if somewhat imprecise, characteristic of expert systems. We can identify eight common characteristics of expert systems that set them apart from conventional computer systems.

Explanation facility

Many people involved with expert systems agree that the most significant distinguishing characteristic of an expert system is its explanation facility, sometimes known as its "human window". Most expert systems designed today can explain their own line of reasoning. They can answer queries posed by the user, either during or at the end of a question-and-answer session. The form of user query can be WHY (are you asking me for this information), or HOW (was some particular conclusion reached). User queries of both the WHY and HOW form are illustrated in Figure 2.l.

Facts and heuristics

The performance level of an expert system is primarily a function of the size and quality of the knowledge base that it possesses. The knowledge represented in the knowledge base consists of both facts and heuristics about the application area (called the domain). An important distinguishing feature of an expert system is the inclusion of heuristics. They are rules of good judgement — the commonsense, informal, judgemental knowledge of actions to take when a situation arises.

Extendable knowledge base

The knowledge base of an expert system can be easily extended or modified. Knowledge is represented in the knowledge base as a collection of individual and independent pieces of fact and heuristic. Any one of these pieces can be changed, or pieces removed

Figure 2.1 Explanation facility

The following fragment of dialogue between the user and MYCIN-TEIRESIAS is an example of the system's explanatory capabilities. The dialogue begins with a question from MYCIN to the user. After each question asked by the user, TEIRESIAS prints an explanation of its understanding of the meaning of the question as a means of avoiding any possible misunderstandings.

MYCIN: Is the patient's illness with ORGANISM-1 a hospital-acquired infection?

USER: WHY *(WHY is it important to determine whether or not the infection with ORGANISM-1 was acquired while the patient was hospitalised?)*

TEIRESIAS: [1.0] This will aid in determining the category of ORGANISM-1.
It has already been established that
[1.1] the gram stain of ORGANISM-1 is gramneg, and
[1.2] the morphology of ORGANISM-1 is rod, and
[1.3] the aerobicity of ORGANISM-1 is facultative
Therefore, if
[1.4] the infection with ORGANISM-1 was not acquired while the patient was hospitalised then there is strongly suggestive evidence (.8) that the category of ORGANISM-1 is enterobacteriaceae
Also: there is weakly suggestive evidence (.1) that the identity of ORGANISM-1 is pseudomonas-aeruginosa
[RULE 37]

USER: HOW 1.3 *(HOW was it established that the aerobicity of ORGANISM-1 is facultative?)*

TEIRESIAS: The following were used:
[3.1] RULE 27 indicated there is strongly suggestive evidence (.8) that the aerobicity of ORGANISM-1 is facultative
[3.2] RULE 27 indicated there is weakly suggestive evidence (.2) that the aerobicity of ORGANISM-1 is anaerobic
Since this gave a cumulative CF of (.8) for facultative, and (.2) for anaerobic, it has been established that the aerobicity of ORGANISM-1 is facultative.

USER: WHY DIDN'T YOU CONSIDER STREPTOCOCCUS AS A POSSIBILITY FOR ORGANISM-1?

TEIRESIAS: The following rule could have been used to determine that the identity of ORGANISM-1 was streptococcus: RULE 33
But clause 2 ('the morphology of the organism is coccus') was already known to be false for ORGANISM-1, so the rule was never tried.

or added, with no great impact on the remainder of the knowledge base. This contrasts with the traditional procedural program, where a small change to the logic can have serious effects on the system.

Processing logic in the knowledge base

In most expert systems the processing logic is an integral part of the knowledge base. The facts and the rules are sufficiently comprehensive normally to contain all the information that is logically required to make any decision. All the reasoning mechanism (inference engine) does is to ensure that relevant facts and rules are retrieved at the right time. In a traditional computer system, the processing rules (logic) are embodied in the program and the data is held in a separate store.

Inference engine

The reasoning mechanism of an expert system, which is known as the inference engine, is separate from the knowledge base. The inference engine inteprets the knowledge base and so is sometimes also known as the 'rule interpreter'. The inference engine tests the individual rules or pieces of knowledge by pattern matching, activating them when there is a match. The rule interpreter uses a predetermined control strategy for searching through the rules and deciding which rules to apply ('enabling' the rules).

Question-and-answer session

The search is normally carried out through a question-and-answer (consultation) session. The system can ask the user for guidance or for further information when it is unable to deduce the next step. Moreover, the expert system is able to explain its own line of reasoning. This is why the user interface is so important, and why so much effort (up to 80 per cent of development time in some cases) goes into the design of the user interface.

Domain expertise

Most of the serious applications of expert systems to date are designed to be used by, and to benefit, people who are experts in the field (domain). This is because the consultation session relies on the user having a highly developed understanding of the subject matter. The reason for this is that many of the questions call for a judgemental answer. Only a person knowledgeable in the field (called a domain expert) and familiar with the domain's assumptions and ambiguities, is able to place in context the final suggested solution, or the advice or answer offered by the system.

Plausible solution

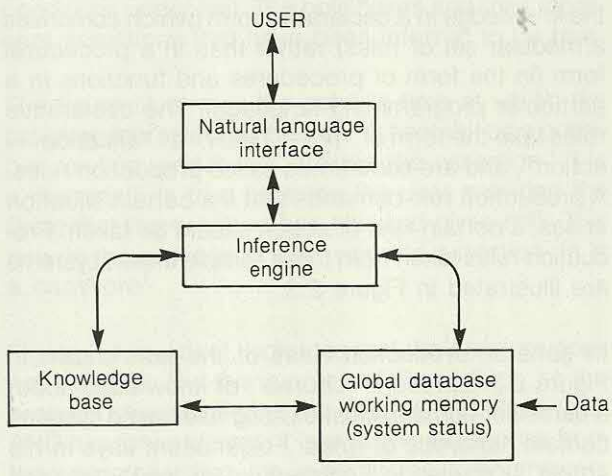
A typical expert system provides its user at the end of a consultation session with a possible or plausible solution to the problem posed at the outset. Rather than being presented in the form of a definite answer, the solution takes the form of advice to the user. The solution may not be right. But it is more likely to be right than other plausible solutions. It is the outcome of deduction involving both the facts and the heuristics stored in the knowledge base.

STRUCTURE OF EXPERT SYSTEMS

Feigenbaum has described the common structure of an expert system (see reference 5) as follows: "the basic structure of an expert system normally consists of a knowledge base and an inference procedure. The knowledge base contains the facts and heuristics. The inference procedure consists of the processes that work over the knowledge base to infer solutions to problems, to do analyses, to form hypotheses and so forth. In principle, the knowledge base is separable from the inference procedure".

A typical expert system normally has (in addition to the knowledge base) a working memory for keeping track of the status of the problem, for inputting data, and for recording the relevant history of what has been done so far. The structure of an expert system is outlined in Figure 2.2.

Figure 2.2 Structure of an expert system



Knowledge base

The knowledge base is a database of knowledge in which both facts and heuristics are represented as individual pieces of knowledge about a particular field (domain).

The use of heuristics is a characteristic of expert systems. The heuristics relate to the specific domain in question. They are acquired from domain experts — people who are experts in the field. They are rules of good judgement which will be used to produce an effective and efficient solution to a problem. This knowledge, together with facts about the domain, is normally organised and represented in the knowledge base.

In his article "Expert System" (reference 6), Michie states: "Expert systems are not, and owing to the complexity of their tasks cannot be, either procedure-driven in the ordinary sense or data-driven, although they can all be fairly described as database-driven. The great bulk of the database, however, is typically made up of rules which are invoked by pattern-matching with features of the task-environment and which can be added to, modified or deleted by the user. A database of this special type is ordinarily called a knowledge base, and its existence determines that there are three different user-modes for an expert system in contrast to the single mode (getting answers to problems) characteristic of the more familiar type of computing:

—Getting answers to problems (user as client).

—Improving or increasing the system's knowledge (user as tutor).

—Harvesting the knowledge base for human use (user as pupil)."

The best form of representing knowledge in the knowledge base is open to question. The topic is still being debated and is the subject of active research. The most common approach, however, is to encode the knowledge in a declarative form (which comprises a modular set of rules) rather than in a procedural form (in the form of procedures and functions in a particular programming language). The declarative rules take the form of "IF — THEN" or "situation — action", and are sometimes called production rules. A production rule demands that if a certain situation arises, a certain kind of action should be taken. Production rules taken from three sample expert systems are illustrated in Figure 2.3.

In general, production rules of the sort shown in Figure 2.3 represent "chunks" of knowledge about a particular domain. Most existing rule-based systems contain hundreds of rules. Feigenbaum says in his article "Knowledge Engineering for the '80s" (see reference 5): "the performance level of an expert system is primarily a function of the size and quality of the knowledge base that it possesses". The process of building the knowledge base is currently a painstaking and lengthy task. Highly trained computer scientists work with domain experts in an attempt to represent organised heuristics. This process tends to be an iterative, incremental one of encoding heuristics into rules. Even after initial implementation, continuous refinement of the knowledge base is undertaken to improve the performance of the system.

DEC's R1 system for configuring VAX hardware systems illustrates the point. From its original 500 rules, the knowledge base of R1 has grown to 2,500 rules. As Feigenbaum has said, "the problem of knowledge acquisition is the critical bottleneck problem in artificial intelligence".

Inference engine

In addition to the knowledge base containing rules, a mechanism is needed for manipulating the rules to form inferences, to make diagnoses and so forth. In order for a system to reason, it must be able to infer new facts from what it has been told already. The rules have the following general form:

IF: antecedent , antecedent , ...,

THEN: consequent with certainty , consequent with certainty , ..., consequent with certainty

The antecedents can be thought of as patterns that can be matched against entries in the database, and the consequents as actions that can be performed (or conclusions that can be deduced) if all the antecedents match.

Figure 2.3 Typical production rules from three expert systems

Production rule used by the R1 system to configure DEC's VAX system

IF: the most current active context is assigning a power supply,
and a unibus adaptor has been put in a cabinet,
and the position it occupied in the cabinet (its nexus) is known,
and there is space available in the cabinet for a power supply for that nexus,
and there is an available power supply,
and there is no H7101 regulator available

THEN: add an H7101 regulator to the order.

Production rule used by the MYCIN system for medicial diagnosis

IF: the site of the culture is blood, and
the identity of the organism is not known with certainty, and
the stain of the organism is gramneg, and
the morphology of the organism is rod, and
the patient has been seriously burned

THEN: there is weakly suggestive evidence (0.4*) that the identity of the organism is pseudomones.

*(*The number 0.4 indicates the degree to which the conclusion follows from the evidence, on a scale of 0 to 1.)*

Production rule used by the PROSPECTOR system in mineral exploration

IF: there is Hornblende pervasively altered to biotite

THEN: there is strong evidence (*320,0.001) for potassic zone alteration.

**Bayesian probability theory is used here to determine probabilities at each stage. The number 320 indicates how sufficient the evidence is for establishing the hypothesis if the evidence is, in fact, present. A larger value means greater sufficiency. The number 0.001 indicates the degree of necessity of the evidence for establishing the hypothesis. A smaller value means greater necessity. Both these numbers determine the adjustment to be made to the current probability estimate of the hypothesis, in view of the evidence.*

Most of the mechanisms needed for manipulating the rules to form inferences can be regarded as a form of theorem prover. However there is little commonality in detailed system architecture. Expert system design is at present unique to the domain and to the designer's approach. Different designers often use different techniques to produce an efficient system. Problem solving is carried out by searching through all the possible solutions. But the number of candidate solutions is usually so great that an exhaustive search is not feasible.

Two main approaches are used to overcome the difficulties associated with search in complex problems. The first approach is to find ways to search efficiently through the logical alternatives (the search space). The second approach is to find ways to transform a large search space into smaller, more manageable spaces that can be searched efficiently.

A detailed discussion of search methods and control strategies is beyond the scope of this report. Interested readers should consult reference 7. A simplified explanation of the main strategies used is given below.

Goal-driven control strategy (backward chaining)

A goal-driven control strategy, also called backward chaining, means searching backwards from the goal. In this type of search the system works from goal to sub-goal. Using the action side of the rules to deduce the condition side, the system proceeds in a hierarchical search trying to satisfy all the conditions necessary to meet the chosen goal. Each rule is tested in turn. If the antecedents for a rule match all the existing facts in the database, the rule is applied ('fired'). If an unmatched antecedent is encountered, matching it becomes a new sub-goal, and the procedure is applied recursively. If there are no rules in the knowledge base to establish the new sub-goal, the system asks the user for the necessary facts and enters them in the knowledge base. The behaviour of the system is therefore directly related to the goals it is trying to achieve. The goal-driven, backward-chaining strategy is also known as top-down reasoning or consequent reasoning.

Data-driven control strategy (forward chaining)

A data-driven control strategy, also called forward chaining, means searching forward after starting from a given set of conditions. In forward chaining the system simply scans through the rules until one is found whose antecedents match assertions in the knowledge base. The rule is then applied, the knowledge base updated, and the scanning resumed. This process continues until either a goal state is reached, or no applicable rules are found. The behaviour of this strategy is directly related to the facts about the problems entered in the knowledge base, and the strategy is known as data-driven, bottom-up, or antecedent reasoning.

Bi-directional control strategy

To improve the efficiency of the search, sometimes both backward chaining and forward chaining are used. This strategy is called bi-directional control strategy and involves searching from both ends of the knowledge base, and (hopefully) meeting somewhere in the middle. Such combined search is applicable to complex problems when the search space is large.

An example of a simple rule-based system

We now describe a simple expert system which was built to demonstrate the backward-chaining procedure used in MYCIN. The system is rule-based — the knowledge is encoded in situation-action rules. The system includes 15 rules for identifying animals. The knowledge base is too small and too simple for serious use but the system illustrates the back-chaining inference process.

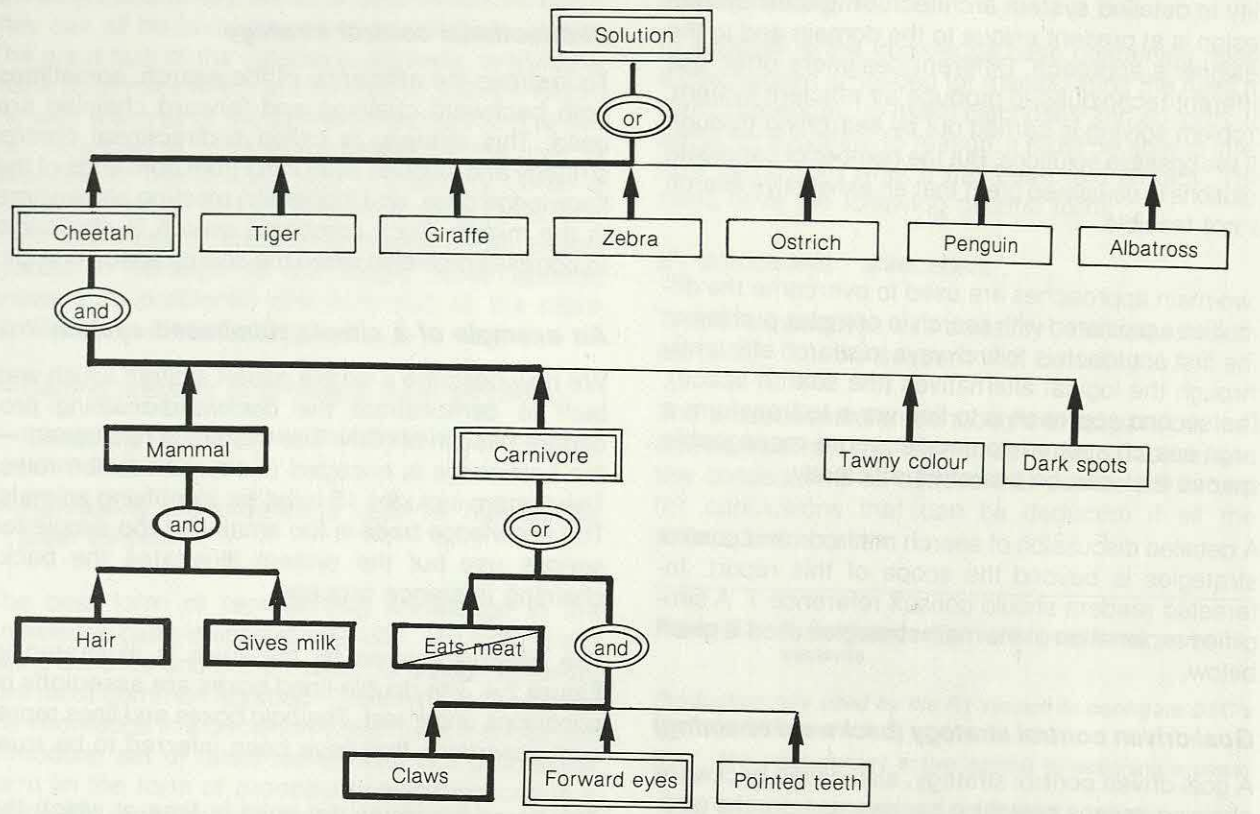
The network formed by the rules is illustrated in Figure 2.4. The double-lined boxes are assertions or conditions under test. The bold boxes and lines represent assertions that have been inferred to be true.

The figure illustrates the point in time at which the program has selected the goal "cheetah" as a possible solution, and it has inferred the assertion "it is a mammal" is true because the user provided the facts that the creature has hair and gives milk. It is now in the process of checking the assertion "it is a carnivore".

Figure 2.4, overleaf, indicates that the creature does not eat meat. But this is only an OR condition, so the program checks whether the creature has claws (yes) AND has forward eyes; it is awaiting a response from the user regarding the eyes. If the user answers "yes", then the program will go on to check if the creature has pointed teeth. If the answer again is yes, then the creature is inferred to be a carnivore. The program will go on to check for tawny colour, dark spots and so forth.

On the other hand, if the creature neither has forward eyes nor pointed teeth, then it is not a carnivore so cannot be a cheetah. Under these circumstances, the inference engine would backtrack to try an alternative goal — tiger, which is the next goal in sequence — as a possible solution. The progam would continue in this way until either it achieves the goal, or it exhausts all possible goals. In other words, the eventual outcome of the back-chaining process is either to identify one or more plausible solutions, or to identify no possible solution.

Figure 2.4 A backward-chaining inference mechanism



(Source: Lewis, J. W. and Lynch, F. S., British Computer Society Technical Conference on Expert Systems, September 1982)

EXPERT SYSTEM LANGUAGES

Because knowledge can best be represented in the form of symbolic declarative expressions, the inference process is, in the main, a pattern-matching process of symbols or lists. List processing languages are therefore the natural software environment for building expert systems.

Symbols and structures

In their discussion of physical symbol systems, Newell and Simon (see reference 8) define a symbol as a physical pattern that can occur as a component of a symbol structure. Moreover, they regard a symbol structure as being composed of a number of symbols related in some physical way, such as being next to each other. We believe that for simplicity it is usually sufficient to think of symbols as strings of characters, and of symbol structures as a type of data structure (called a list structure) containing symbols. The following character strings are examples of symbols:

Apple;Transistor-I3;Running;Five;3.I4I59

The following are examples of symbol structures:

(On Block I Block 2);(Plus 5 X);(Same-as (Father of Pete) (Father-of (Brother of Pete)))

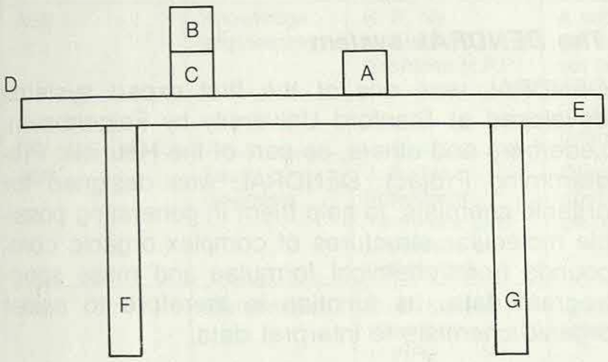
One of the early contributions to computer-based research in artificial intelligence was the invention of list-processing languages for symbolic computation. These languages provide facilities for manipulating lists, and facilities for managing their storage. Our discussion of symbols and symbol structures emphasises how they can be used to represent knowledge.

Predicate calculus is a widely studied formal language of symbol structures which can be used for representation in a computer. An understanding of predicate calculus is an essential foundation for understanding the representation of knowledge and the inferences that can be made from the knowledge. A detailed discussion of predicate calculus is beyond the scope of this report but, nevertheless, we provide a simple illustration of its concepts.

Figure 2.5 shows a sketch of a table with some blocks on it. The figure also shows some symbol structures representing the 'information' in the sketch. These symbol structures are written in a syntactic variation (prefix format) of predicate calculus. They are made

Figure 2.5 Predicate calculus representation

This figure shows a table with blocks on it and a predicate calculus representation for the information.



A, B and C are defined as blocks:	(IS — A A BLOCK) (IS — A B BLOCK) (IS — A C BLOCK)
D defined as table: F, G defined as table legs:	(IS — A D TABLE) (IS — A F TABLE-LEG) (IS — A G TABLE-LEG)
E, F, G defined as part of D:	(PART — OF E D) (PART — OF F D) (PART — OF G D)
Define the physical relationship between the items:	(ON A D) (ON C D) (ON B C)

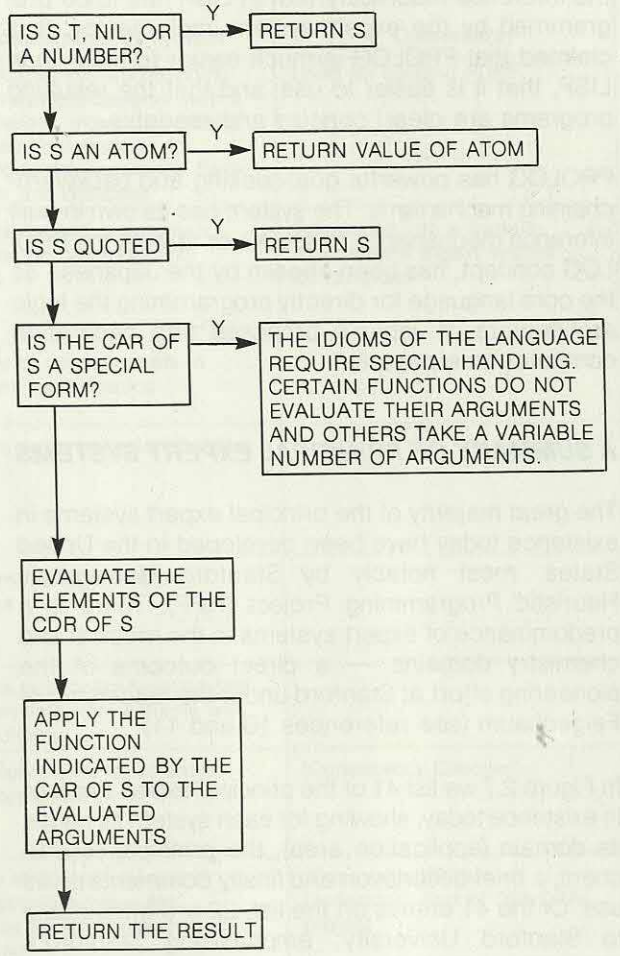
up of terms and predicate symbols. Terms are used for the names of things and predicates represent relations between things. In Figure 2.5, A,B,C,D,E,F,G, BLOCK, TABLE, TABLE-TOP and TABLE-LEG are terms, and IS-A, PART-OF, and ON are predicate names. Predicate calculus is a branch of logic and simple predicates like those in Figure 2.5 are called propositions or atomic formulae.

The LISP language

The LISP language is one of the standard vehicles for encoding symbolic processes. Like other languages in its class, LISP is an invaluable aid for processing descriptions. When we interviewed Duda, head of expert systems at Fairchild Camera and Instrument Corporation, he emphasised this point with the following words: "if you are going to do one of these jobs (expert systems) you just cannot do it in Pascal. You would spend all your time fighting the limitations of compilers and storage. So the artificial intelligence people have invested an effort in building these nicer environments".

The simple flowchart in Figure 2.6 helps to explain what LISP can do with an expression that has been entered.

Figure 2.6 LISP evaluation process of a symbolic (S) expression, displayed as a flowchart



Note: T, NIL corresponds to true, false.
Note: CAR and CDR are basic symbol-manipulating functions.
Note: An atom is an S-expression whose value is stored in a table rather than the result of some computation.

LISP is the main symbolic language used by academies and commercial organisations in the United States.

The PROLOG language

PROLOG is a declarative language based on logical relationships between objects.

PROLOG was developed and first implemented by Alain Colmeraner's research group in Marseilles. It was originally devised for the purpose of implementing a natural language question-answering system. It is now based on Kowalski's procedural interpretation of Horn clause predicate logic (see reference 9). PROLOG is a higher level language than LISP.

A PROLOG program can be regarded simply as a collection of statements of fact, called clauses. The appropriate clause is selected by a pattern-matching operation. Pattern matching is the sole data-manipulation operation. PROLOG provides much of the inference machinery that in LISP has to be programmed by the expert system implementor. It is claimed that PROLOG is much easier to learn than LISP, that it is easier to use; and that the resulting programs are clear, concise and readable.

PROLOG has powerful goal-seeking and backward-chaining mechanisms. The system has its own in-built inference mechanisms. PROLOG, or at least the PROLOG concept, has been chosen by the Japanese as the core language for directly programming the logic architecture of Japan's proposed fifth generation computer (see page 6).

A SUMMARY OF PRINCIPAL EXPERT SYSTEMS

The great majority of the principal expert systems in existence today have been developed in the United States, most notably by Stanford University's Heuristic Programming Project (HPP). There is a predominance of expert systems in the medical and chemistry domains — a direct outcome of the pioneering effort at Stanford under the leadership of Feigenbaum (see references 10 and 11).

In Figure 2.7 we list 41 of the principal expert systems in existence today, showing for each system its name, its domain (application area), the principal researchers, a brief description and finally comments on its use. Of the 41 entries on the list, 22 are attributable to Stanford University, emphasising Stanford's domination of this field.

It is instructive to note that, of the 41 expert systems listed in Figure 2.7, only one (R1) has had a genuine commercial impact. R1 was developed as a collaborative venture between Carnegie-Mellon University and DEC for the purpose of configuring DEC's VAX 780 computer system. When we interviewed McDermott of the Department of Computer Science at Carnegie-Mellon University in Pittsburgh, he claimed that R1 is now repaying its initial investment every three months. The development effort, however, was based on seven years' previous research at Carnegie-Mellon. Apart from R1, other systems listed in Figure 2.7 have proven to be successful from a technical standpoint. PROSPECTOR, PUFF and the DIPMETER ADVISOR system are all notable in this respect. But it is not yet clear what commercial success they have had, nor whether in the future they will repay the investment made to develop them.

DESCRIPTION OF FOUR EXPERT SYSTEMS

In this section we choose four of the 41 expert systems listed in Figure 2.7, describing each of them in turn. The four are: DENDRAL, MYCIN, PROSPECTOR and R1. These systems represent the state of the art at the time of writing this report.

The DENDRAL system

DENDRAL was one of the first expert systems developed at Stanford University by Feigenbaum, Lederberg and others, as part of the Heuristic Programming Project. DENDRAL was designed for organic chemists, to help them in generating possible molecular structures of complex organic compounds from chemical formulae and mass spectrogram data. Its function is therefore to assist organic chemists to interpret data.

It is claimed that DENDRAL rivals expert human performance for a number of molecular families. Indeed, DENDRAL is still regarded by some people as the most successful expert system ever built. Work started on DENDRAL in 1965. A basic aim was to use heuristic knowledge to limit the search for solutions, which explains why DENDRAL is sometimes also known as HEURISTIC DENDRAL.

DENDRAL runs with what is known as a "plan, generate and test" control strategy. It first derives the necessary constraints on the molecular structure, then systematically generates structures that satisfy these constraints. Finally, it tests the proposed structures by predicting the mass spectrogram readings, rejecting those that disagree with the experimental results.

The knowledge base contains rules for deriving constraints on molecular structure from experimental data, rules for predicting mass spectrogram readings from structures, and a highly sophisticated procedure for generating candidate structures to satisfy the constraints.

Originally, DENDRAL was custom-designed. As time went by, the knowledge base changed significantly. Its designers found that large parts of the system had to be rewritten. In an effort to avoid this, DENDRAL's designers looked for ways to increase the rate of transfer of expertise from chemists into the DENDRAL system. An extension of the DENDRAL project, known as META-DENDRAL, aims to achieve that through the inclusion of "higher-level" rules, which are used to examine data and to discover rules for determining molecular structures from mass spectrometry data. Plausible rules are first generated from an analysis of experimental data. These rules are then refined. As a result, some new rules have been discovered and previously known ones successfully rediscovered — all within a fixed and very limited vocabulary.

Figure 2.7 Principal expert systems

Name of system or project	Application area	Principal researcher(s)	Brief description of system	Comments
AGE	Knowledge engineering	H. P. Nii N. Aiello (Stanford H.P.P.)	A sophisticated system providing the expert system designer with a set of separate, interconnectable, pre-programmed modules for implementing the knowledge base, interpreter and database	Used for building PUFF
AL/X	Fault diagnosis	J. Reiter (Intelligent Terminals Ltd./ University of Edinburgh)	Diagnoses causes of automatic shut-downs on oil production platforms	Developed into a domain-independent expert system building tool
AM	Mathematics	D. B. Lenat (Stanford H.P.P.)	Discovery of new concepts in elementary mathematics	
BACON.4	Science	P. Langley G. Bradshaw (Carnegie-Mellon University)	Discovers empirical scientific laws	
CASNET	Medicine	S. Weiss, C. Kulikowski (Rutgers University)	Long-term management of glaucoma	
CENTAUR	Medicine	J. S. Aikins (Stanford H.P.P.)	Interprets pulmonary function test measurements from patients with lung disorders	
CONCHE	Science	I. H. Chisholm D. H. Sleeman (University of Leeds)	An intelligent aide for scientific theory formation	'Consistency Checker'
CONGEN	Science	R. E. Carhart (Stanford H.P.P. /University of Edinburgh)	Aids the structural chemist in finding possible molecular structures for an unknown compound	'Constrained Structure Generator'. Part of DENDRAL project
CRIB	Fault diagnosis	T. R. Addis (International Computers Limited)	Diagnosis of faults in computer hardware and software	
CRYSALIS	Science	E. A. Feigenbaum R. S. Engelmore (Stanford H.P.P.)	Infers the structure of a protein from a map of electron density derived from x-ray crystallographic data	
DART	Engineering	(Stanford H.P.P.)	Diagnosing hardware faults in computer systems. (Under development.)	Joint project with I.B.M.
DIPMETER ADVISOR	Geology	MIT/ Schlumberger	Inferring sub-surface geological structures from oil well dipmeter readings	
DENDRAL (HEURISTIC DENDRAL)	Science	E. A. Feigenbaum J. Lederberg B. G. Buchanan et al. (Stanford H.P.P.)	Identification of organic compounds by analysis of mass spectrograms	First expert system began in 1965. Concerned with using knowledge to limit search
EMYCIN	Knowledge engineering	W. Van Melle (Stanford H.P.P.)	A domain-independent version of MYCIN, usable for developing rule-based consultation programs for many fields	'Essential MYCIN' (Used for PUFF, SACON, ONCOCIN etc.)
EXPERT	Knowledge engineering	S. Weiss C. Kulikowski (Rutgers University)	A system for designing and building models for consultation	
GAMMA	Science	D. R. Barstow (Yale University)	Interpreting gamma ray activation spectra	

Continued on next page

Figure 2.7 continued

Name of system or project	Application area	Principal researcher(s)	Brief description of system	Comments
GUIDON	Education	W. J. Clancey (Stanford H.P.P.)	Case-method tutor designed to improve a student's ability to diagnose complex problems in medicine and science	Exploits the MYCIN knowledge base to teach students both facts and problem-solving strategies
HEADMED	Medicine	J. F. Heiser	Psychopharmacology advisor	Constructed using EMYCIN (270 rules)
INTERNIST	Medicine	J. D. Myers H. E. Pople (University of Pittsburg)	Diagnosis in internal medicine	Can involve multiple instances of 500 different disease types
LOGIN	Engineering	E. A. Feigenbaum H. P. Nii (Stanford H.P.P.)	Development of tools to complement or supplement signal processing programs (initially by adding geological information)	
MACSYMA Advisor	Mathematics	M. R. Genesereth (M.I.T.)	An automated consultant for MACSYMA (an algebraic manipulation system)	
MDX	Medicine	B. Chandra-sekaran (Ohio State University)	Performs diagnoses related to cholestasis	
METADENDRAL	Science	B. G. Buchanan (Stanford H.P.P.)	Induces rules for determining molecular structure from mass spectrometry data	Part of DENDRAL project
MOLGEN	Science	J. Lederberg N. Martin P. Friedland M. Stefik et al (Stanford H.P.P.)	Provides intelligent advice to a molecular geneticist on the planning of experiments involving the manipulation of DNA	
MYCIN	Medicine	E. Shortliffe (Stanford H.P.P.)	Diagnoses certain infectious diseases and recommends appropriate drug treatment	(400 rules)
ONCOCIN	Medicine	E. Shortliffe A. C. Scott (Stanford H.P.P.)	Assists in the management of cancer patients on chemotherapy protocols for forms of lymphoma	'Oncology Protocol Management System'. Constructed using EMYCIN
PROSPECTOR	Geology	P. Hart R. Duda (SRI International)	Aids geologists in evaluating mineral sites for potential deposits	
PSYCO	Medicine	J. Fox (Imperial Cancer Research Fund, London)	Diagnoses dyspepsia	Experimental Production System compiler (Initial application)
PUFF	Medicine	J. C. Kunz (Stanford H.P.P.)	Analyses results of pulmonary function tests for evidence of possible pulmonary function disorder	In routine use at Pacific Medical Center Hospital, San Francisco. Constructed using EMYCIN (250 rules)
R1	Computing	J. McDermott (Carnegie-Mellon University)	Configuring the VAX/780 computer system	Development is known as XSEL
RAFFLES	Fault diagnosis	T. R. Addis (International Computers Limited)	Diagnosis of faults in computer hardware and software	
RITA	Knowledge engineering	R. H. Anderson (Rand Corporation)	Provides the user with a language for defining intelligent interfaces to external data systems	'Rule-directed Interactive Transaction Agent'
RLL	Knowledge engineering	R. Greiner D. B. Lenat (Stanford H.P.P.)	Provides user with a flexible set of facilities as a tool for building his own knowledge representation language	'Representation Language Language'. (Developed from UNITS.)

Continued on next page

Figure 2.7 continued

Name of system or project	*Application area*	*Principal researcher(s)*	*Brief description of system*	*Comments*
RX	Medicine	R. L. Blum (Stanford H.P.P.)	Derives knowledge about the course and treatment of chronic diseases from a database of patient information	
SACON	Engineering	J. S. Bennett R. S. Engelmore (Stanford H.P.P.)	Advises structural engineers in using the structural analysis program MARC	'Structural Analysis Consultant'. Constructed using EMYCIN (170 rules)
SECS	Science	W. T. Wipke	Proposes schemes for synthesising stated organic compounds	
SU/X	Engineering	H. P. Nii and E. A. Feigenbaum (Stanford H.P.P.)	Forms and updates hypotheses about location, velocity etc. of objects from primary signal data (spectra)	
TEIRESIAS	Medicine	R. Davies (Stanford H.P.P.)	Knowledge acquisition program used with MYCIN	An ambitious attempt to develop a tool for the process of acquiring knowledge
UNITS	Knowledge engineering	M. Stefik (Stanford H.P.P.)	Interactive language providing general-purpose facilities for knowledge representation. Used for MOLGEN plus other small applications	(Originally developed as part of MOLGEN.) (Now being superseded by RLL)
VLSI	Engineering	(Stanford H.P.P.)	Assistance in the design of very large scale integrated circuits. (Under development)	Joint project with Stanford Centre for Integrated Systems
VM	Medicine	L. M. Fagan (Stanford H.P.P.)	Provides diagnostic and therapeutic suggestions for critical care patients needing mechanical assistance with breathing	'Ventilator Management'. Operates in real-time, with time-dependent relations (120 rules). Used in the intensive care unit of the Pacific Medical Center, San Francisco

(Source: Adapted from Bramer)

The MYCIN system

MYCIN is an expert system that is designed to diagnose bacterial infections and to recommend antibiotic therapy. Starting in 1975, the system was developed by Shortliffe and others as part of the Heuristic Programming Project at Stanford University. MYCIN is particularly concerned with blood infections and meningitis infections.

A MYCIN session begins with a consultation with a physician. During the course of this dialogue, the physician supplies the system with information relevant to a particular case. MYCIN's control strategy is backward chaining. The search starts with the various possible organisms as goals to be achieved, and tries to satisfy the necessary conditions by matching them with available data.

MYCIN's knowledge base comprises over 500 production rules. About half of them apply to blood infections and about half to meningitis infections. Each rule takes the form IF (condition) THEN (conclusion). Rules adopting this format are sometimes known as situation-action rules.

For each rule, the expert estimates on a scale from 0.1 to 1.0 the degree of certainty attached to a conclusion. These degrees of certainty are then converted automatically to probability values. MYCIN has a very simple and basic model of inexact reasoning, used to manipulate these certainty factors and to produce an overall "cumulative certainty factor". The following illustrates a typical rule:

IF:

1. The site of the culture is blood, and 2. The gram stain of the organism is gramneg and 3. The morphology of the organism is rod, and 4. The patient is a compromised host.

THEN:

There is suggestive evidence (0.6) that the identity of the organism is pseudo-aeruginosa.

MYCIN backchains from the goals, testing the 'truth' of the conclusions of the relevant rules. It does this in one of two ways: either by deducing the condition from previous conclusions, matching the condition with available data, or alternatively by asking the user for additional information.

The output from MYCIN is a list of possible alternative diagnoses, for each of which it assesses a degree of certainty. MYCIN also recommends treatments to cover all the possibilities it identifies. The recommendation takes a form that is illustrated by the following:

"My preferred therapy recommendation is as follows:

In order to cover for items 1, 2, 3 and 4:

Give the following in combination:

1 AMPICILLIN

Dose: 3.5q (28.0ml) q4h IV (calculated on basis of 50mg/kgl).

2 GENTAMICIN

Dose: 119mg (3.0ml, 80mg/2ml ampule) q8h IV (calculated on basis of 0.7mg/kg) plus consider giving 5mg q24h Intrathecal.

Comments: Monitor serum concentrations.

Since high concentrations of penicillins can inactivate aminoglycosides, do not mix these two antibiotics in the same IV bottle."

MYCIN's explanation facility can exhibit selected portions of the chain, as requested by the user. It can answer a variety of queries during or after the consultation dialogue, such as 'WHY (are you asking me for this information)?' or 'HOW (was some particular conclusion reached)?' One interesting form of query is shown in Figure 2.8. To answer it, MYCIN must keep track not only of acceptable lines of reasoning, but also of the invalid lines it has explored and the reasons for their rejection.

Figure 2.8 An illustration of MYCIN's explanation facility

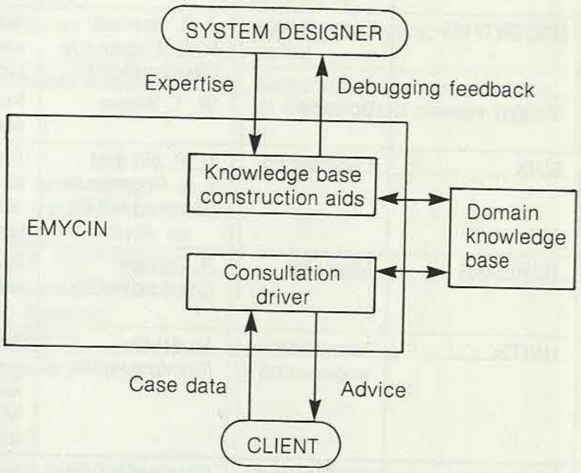
USER: WHY DIDN'T YOU GIVE TETRACYCLINE FOR E.COLI IN REC-1

MYCIN: TETRACYCLINE was discounted for ITEM-1 (RECOMMENDATION-1) because there is evidence that this e.coli is not sensitive to it.

An extension of the MYCIN project known as EMYCIN (Essential MYCIN) has been developed more recently. EMYCIN is domain independent. It is a framework, designed to help build MYCIN-like systems. A basic assumption of EMYCIN is that production rules are an appropriate framework for representing the new knowledge base, and that a backward chaining or goal-directed interpreter is an appropriate inference mechanism.

EMYCIN helps systems designers to structure knowledge of a problem. Once a designer has used EMYCIN to build a new knowledge base, EMYCIN interprets the knowledge base with the inference engine. These two main functions are illustrated in Figure 2.9.

Figure 2.9 The main functions of EMYCIN



The PROSPECTOR system

PROSPECTOR is a consulting aid for evaluating the mineral potential of a site or region. It is a rule-based expert system. Knowledge about a particular type of ore deposit is encoded in a computational model representing observable geological features and their relative significance. The PROSPECTOR system was developed by Duda, Hart and others at SRI International in California. PROSPECTOR is being developed in a modular way, with each module corresponding to a deposit type.

Work begins with a description of deposit characteristics. The description is then developed into a model that is progressively refined with more data from different finds.

The sponsor of PROSPECTOR is the US Geological Survey which is responsible on a continuing basis for assessing mineral deposits. At times the task becomes particularly critical — as in deciding which areas in Alaska to set aside for wilderness and which to earmark for development. Here, the challenge was to assess beforehand the mineral potential of the land.

The US Geological Survey estimated that for work in Alaska at least 150 types of deposit needed to be assessed. This information was available in a variety of independent reports, each written by a geologist specialising in one particular ore. Information on about 36 deposits is built into the PROSPECTOR system, but only half had been refined to an operational stage by mid-1983. Duda has estimated the amount of work needed to establish a reliable rule base for just one deposit type as six months. When interviewed by our researcher, he commented: "If you wanted to do it for 200 types, that is 100 man-years work. It is simply a question of manpower.

"What you see here is a knowledge-acquisition problem. That is why our researchers are working on learning and language comprehension. The proper goal for the next 20 years is to have the secretary type the information in, and the rules generated automatically. But that is long-term research". For this reason, the United States Geological Survey is concentrating on critical deposits such as uranium.

Users of PROSPECTOR provide the program with information observed in the field. A question-and-answer session is then carried out with the user supplying additional information, where it is available, on demand. PROSPECTOR is able at any point to provide the user with an explanation of its reasoning, or the conclusions it has reached so far.

The eventual output from PROSPECTOR is an indication of the 'level of certainty' by which the presence of a particular form of deposit in a given site is supported by available evidence. An excerpt from a consultation with PROSPECTOR is shown in Figure 2.10. In general, even the field evidence is indeterminate. The user expresses certainty about a piece of evidence on a scale from minus 5 to plus 5. PROSPECTOR converts the value into probability values.

Initial results of PROSPECTOR have been very promising. The cost of a typical consultation session works out at the surprisingly low figure of $15.

Figure 2.10 An excerpt from a typical consultation with PROSPECTOR

The following questions are intended to establish the nature of the petrotectonic setting. (These questions will not be pursued unless a positive answer is given to the next question.)

12 — To what degree do you believe that the target area is in a continental-margin mobile belt? $\boxed{4}$

13 — Do you know the age of the continental-margin mobile belt? $\boxed{\text{WHY}}$

The vast majority of porphyry copper deposits are younger than 200 million years. Major age groups include:

1. An older group of 200 to 150 m.y. (million year) porphyries in British Columbia;
2. An important younger group of 70 to 50 m.y. deposits in the Mideast, British Columbia (Skeena Arch), and Arizona-New Mexico-Sonora (Laramide) copper province;
3. The major belt of 60 to 4 m.y. deposits in the Andes of Argentina, Chile, Peru, Ecuador, and Colombia.

<The explanation continues with additional information>

13 — Do you know the age of the continental-margin mobile belt? $\boxed{\text{YES}}$

14 — What is the age of the continental-margin mobile belt?

$\boxed{\text{CRETACEOUS = 65-136 MILLION YEARS}}$

What is your confidence in this estimate? $\boxed{3}$

<The consultation continues on with many additional questions.>

Note: □ indicates user answers.
Note: The numerical answer to question 12 indicates the user's degree of certainty about a fact, using a −5 to +5 scale. If the user types "WHY" instead of answering a question, PROSPECTOR explains the question's geological significance.

The R1 system

R1 is an expert system that is used for configuring DEC VAX-11 computer systems. Following seven years of basic research at Carnegie-Mellon University, work on R1 began at the university in December 1978. The development of R1 has been described by McDermott (see reference 12). Less than one year after commencing development, R1 had matured to a point where it could be used regularly by DEC's manufacturing organisation to configure VAX-11/780 systems. At that time, DEC established its own independent group responsible for the maintenance and continuing development of R1. By 1982, the group had grown to a strength of 14, and no longer needed the support of the specialists at Carnegie-Mellon.

Almost all the VAX systems now being delivered within the USA are configured using the R1 system. When it was first implemented in 1979, R1 had 500 rules. Today it has grown to 2,500 rules. This underlines how refining and enhancing an expert system is a very lengthy process. It can take longer than the original effort required to develop the structure and rule base, and can require more resources.

R1 does not handle inexact reasoning. Rather, all the rules and questions-and-answers are deterministic. Two main advantages are claimed for the expert-system approach to the formation of the rule base, in preference to a traditional decision-table approach. The first is that the task of maintaining and updating the database (a seemingly endless task for a manufacturer) becomes more manageable. The second is that the system can be developed incrementally. "It is another approach to software engineering" was the description given to our researchers by Duda, head of expert systems at Fairchild Camera and Instrument Corporation.

The approach of the designers of R1 was to break into sub-tasks the problems of configuring a computer system. The sub-tasks are performed in the following sequence:

—Correct mistakes in the purchase order.

—Put components into CPU cabinets.

—Put boxes into unibus cabinets and put components in boxes.

—Put panels in unibus cabinets.

—Lay out system on floor.

—Do the cabling.

R1 is a data-driven system. It starts from a customer's order, then goes on to match rules to data in the database. R1 takes a set of components as input and produces diagrams showing what the spatial relationships among the components should be. Though R1 knows almost nothing about the sub-task of selecting a set of components to satisfy a functional specification, it does understand that certain components may require other components in order to be configured. If the set of components it is given is incomplete in this sense, it adds whatever components are required to make the set configurable. R1 recognises the acceptable ways in which components can be associated under various conditions. It uses this knowledge to construct a single configuration that satisfies all of the organisational constraints. Because its knowledge is sufficient to enable it to recognise what to do at each step, it performs this task with almost no search. In other words, it seldom needs to backtrack.

R1 is implemented in OPS5 which is a general-purpose, rule-based programming language built on top of LISP and designed to ease the representation of knowledge in production rules form. The language was developed at Carnegie-Mellon University and implemented on DEC10 and DEC20 machines.

Figure 2.11 shows an English translation of a sample rule from R1, using the OPS5 language.

Figure 2.11 English translation of a sample R1 rule

Assign UB modules except those connecting to panels 4

IF: The most current active context is assigning devices to unibus modules
and there is an unassigned dual port disk drive
and the type of controller it requires is known
and there are two such controllers neither of which has any devices assigned to it
and the number of devices that these controllers can support is known

THEN: Assign the disk drive to each of the controllers
and note that the two controllers have been associated and that each supports one device.

CHAPTER 3

EXPERT SYSTEMS IN THE UNITED STATES AND EUROPE

So far in this report we have examined what expert systems are, how they work and what they provide. In this chapter our focus is on user experience. We look at case studies of user experience under three main headings: expert systems in the United States, expert systems in the United Kingdom and expert systems in France.

EXPERT SYSTEMS IN THE UNITED STATES: FIVE CASE STUDIES

Of the 41 principal expert systems listed in Figure 2.7 on pages 15 to 17, more than three-quarters have been developed in the United States. Both the scale and the scope of research into expert systems, and practical experience with them, is greater in the United States than it is in Europe. In the United States the typical expert system is large, expensive, custom-made and specialised. In this section we describe five case studies of American work on expert systems: DEC's XSEL development of the R1 system, IntelliGenetic's KEE (Knowledge Engineering Environment), Schlumberger's DIPMETER ADVISOR, Teknowledge's intelligent interface to statistical software, and Xerox PARC's LOOPS.

DEC's XSEL

XSEL has evolved from R1, the system for configuring VAX computers discussed on page 19. XSEL complements R1, and is designed to assist sales staff in configuring computer systems to fit the needs of customers.

DEC differs from most computer manufacturers in the degree of flexibility it allows its customers in component selection. Rather than marketing a range of standard systems, each with a limited number of options, DEC markets processors with relatively large numbers of options. One of the results of this marketing strategy is that many of the systems sold are unique. Consequently each poses a distinct configuration problem. A computer system configurer has two tasks: to ensure that the system is complete (all components are present), and to determine what the spatial relationship among the components should be. Because a typical DEC system has about 100 components, each having a large number of possible interrelationships, a lot of knowledge is required to achieve the optimum configuration.

Configuring a particular computer involves two main elements: selecting the components needed to fulfil the customer's order, and organising the selected components to form a complete working system. XSEL is concerned with the first of these tasks, and R1 with the second. Having selected the set of components that satisfies the requirements imposed by an application, XSEL then informs R1 of its selection and provides any additional information R1 needs to tailor the configuration to the application. Because part of the task of R1 is to ensure that all support components are included in the order, there is no need for XSEL to concern itself with support components. XSEL permits the customer to specify a processor, an amount of primary memory, whatever software is desired, and all the necessary peripherals. This skeletal order is then passed to R1 to configure and produce a complete order.

XSEL performs consistency checks to ensure that the components ordered are compatible. Customers may specify some of the components in terms of total capability required of the system, rather than by type or name.

Specialist staff at Carnegie-Mellon University are assisting DEC with the development of XSEL. At the time of writing this report in mid-1983, XSEL was being tested and evaluated in field trials.

IntelliGenetic's KEE

IntelliGenetics of Palo Alto, California, is one of the first companies to apply artificial intelligence techniques to commercial applications. The company specialises in computer software and hardware for biotechnology (total staff strength is about 39). It also applies the techniques of artificial intelligence to the development of products to assist in commercial planning, decision making and information management.

IntelliGenetics believes that the most important technical developments to emerge from research into artificial intelligence are the software development 'environments' which had to be created to facilitate productive research. These software environments or tools include highly sophisticated user interfaces, high-level knowledge representation languages and powerful debugging techniques. Dr Kehler, Vice President and Director of Applied Artificial Intelligence at

IntelliGenetics, described these developments as the "industrialisation of artificial intelligence — producing revolutionary new software development tools".

The IntelliGenetics' approach is to further develop these tools and use the artificial intelligence technology to develop specialised products for individual clients.

Kehler emphasised the importance of a close working relationship between knowledge engineering companies and their clients. He said: "Anyone who has a problem that involves in-house expertise, which may save ten million dollars a year, is going to be very cautious about going outside and talking to people about the solution to that problem, because that problem is on the competitive edge. The notion of a knowledge engineering company developing an application product on its own is ridiculous".

A software development tool which IntelliGenetics is introducing to the market at the time of writing this report is KEE (Knowledge Engineering Environment). KEE can store, organise and manipulate knowledge. Unlike some equivalent tools stemming directly from university research, KEE has been engineered specifically as a commercial product.

KEE is designed to function as a foundation for larger knowledge-based systems. An illustration is provided by IntelliGenetics' development, using KEE, of an intelligent control system for a fermentation process. This system is capable of controlling all the stages in the fermentation process, from laboratory experiments and pilot plant systems to the control of entire production plants at three sites in the United States. The system cost over $150,000 to develop.

Schlumberger's DIPMETER ADVISOR system

The DIPMETER ADVISOR system is the result of a four-year effort by Schlumberger in the United States to apply expert system technology to the problem of interpreting oil-well logs.

Schlumberger is a £6 billion international concern, whose main activity is the collection and interpretation of data from oil wells. The company manufactures the measuring equipment, collects the data and interprets the results on behalf of oil companies. The DIPMETER ADVISOR system has been developed by Schlumberger-Doll, the company's research centre in Connecticut. About 250 staff are employed at the centre, including physicists, geologists and computer scientists. Of these, about 10 are involved in research into artificial intelligence.

An oil-well log is the record of geological structures associated with an oil well. Oil-well logs are prepared by lowering instruments into the bore hole, then recording the measurements registered by the instruments as they are raised again to the surface. The resulting logs are sequences of values indexed by depth. Logging instruments, known as tools, measure a variety of petro-physical properties. One such device, the dipmeter tool, measures the conductivity of rock in a variety of directions about the bore hole. Variations in conductivity can be correlated and combined with measurements of the inclination and orientation of the tool to estimate the magnitude and tilt of various formation layers penetrated by the bore hole.

The type of information provided by a dipmeter tool is invaluable in defining hydrocarbon reservoir structures, and designing methods to drain the reservoirs. Factual information of this sort can be combined with expert knowledge of local geology and with rock properties measured by other logs. From this data, a skilled interpreter can deduce a great deal of information — about the geological history of deposition, the composition and structure of the beds, the presence or absence of hydrocarbons and the optimum locations for future wells. Unfortunately, skilled interpreters are a scarce resource. Yet Schlumberger's commercial success in a region is directly related to the level of interpreting expertise it can provide there. The purpose of the DIPMETER ADVISOR system is to raise the performance of less experienced interpreters.

The DIPMETER ADVISOR system attempts to emulate the best human performance in dipmeter interpretations. It makes use of dipmeter patterns, together with local geological knowledge and measurements from other logs. The system has four main components:

- —A knowledge base, where knowledge is represented in the form of production rules.
- —An inference engine that applies the rules in a forward-chaining control strategy.
- —A set of geological models.
- —A menu-driven user interface.

The importance of the last of these four components, the user interface, was stressed at our research interview by Dr. Read Smith of Schlumberger-Doll. He said that this component had accounted for more than half the total development effort.

Initial versions of the DIPMETER ADVISOR system were written in the INTERLISP language, and operate on the Xerox 1100 Scientific Information Processor (known as the Dolphin). Production versions will run on the Xerox Star (average unit price around $35,000). Schlumberger's intention is to order some 150 to 200 Star machines for implementing DIPMETER ADVISOR systems in different geographical regions.

Teknowledge's intelligent interface to statistical software

Teknowledge of Palo Alto, California, employs some 80 staff (mostly computer scientists) and is one of the world's largest companies specialising in knowledge engineering. Teknowledge's activities span the design, development and continuing support of commercial expert systems. The company specialises in large bespoke projects for individual clients. It is also developing expert system software development tools, designed for more general application. Teknowledge offers customised education and training programmes for its clients, as well as more general educational courses for the public.

One of Teknowledge's expert systems is for a client requiring sophisticated analyses of geophysical data. Scientists employed by the client company have at their disposal a database containing more than 100 statistical software packages. In total, the database comprises more than one million lines of Fortran. There are subtle distinctions between the different packages. Experts take years to learn how best to select the programs, and how to 'tune' individual programs by adjusting the parameters.

The purpose of Teknowledge's intelligent interface expert system is to capture the expertise of the scientists on behalf of the client company. Scientists using the system begin by entering data describing their particular problem, either directly from a mainframe or from a LISP-based workstation. Using its knowledge base, the system first classifies the data, then selects the appropriate statistical package together with the relevant parameters. The system then analyses the data, and finally provides advice on the basis of the analysis.

Xerox PARC's LOOPS

LOOPS is the name of a new software product using the techniques of artificial intelligence which is being developed by the Knowledge Systems Area of Xerox's Palo Alto Research Centre (PARC). According to Mark Stefik of PARC, during the past two years the knowledge systems area has moved away from building expert systems, to concentrate instead on the development of software for the organisation of knowledge.

LOOPS integrates a number of knowledge-representation styles used in the field of artificial intelligence: object-oriented programming, which deals with programs as communicating entities with hierarchies of properties; rule-based programming, which uses IF-THEN constructions; access-oriented programming, which is driven by data; and the language LISP itself. The whole LOOPS system has been developed using INTERLISP-D, a dialect of LISP. The development of LOOPS itself was aided by an expert system known as the LOOPS TESTER which, using its knowledge base, can infer what effects may be caused by changes or extensions to LOOPS. Great effort has gone into the design of the user interface of LOOPS. It makes extensive use of interactive graphics featuring display windows. Display features include 'browsers' (tree charts), and gauges (a display of gauges is illustrated in Figure 3.1, overleaf). Powerful debugging tools are also available to the user.

Xerox is using LOOPS to build expert systems for internal use, in different areas within PARC. In one application, an expert system developed using LOOPS is assisting in the design of photocopiers. In another application, LOOPS is used in the development of a planning and decision support system aimed at improving the efficiency of office work.

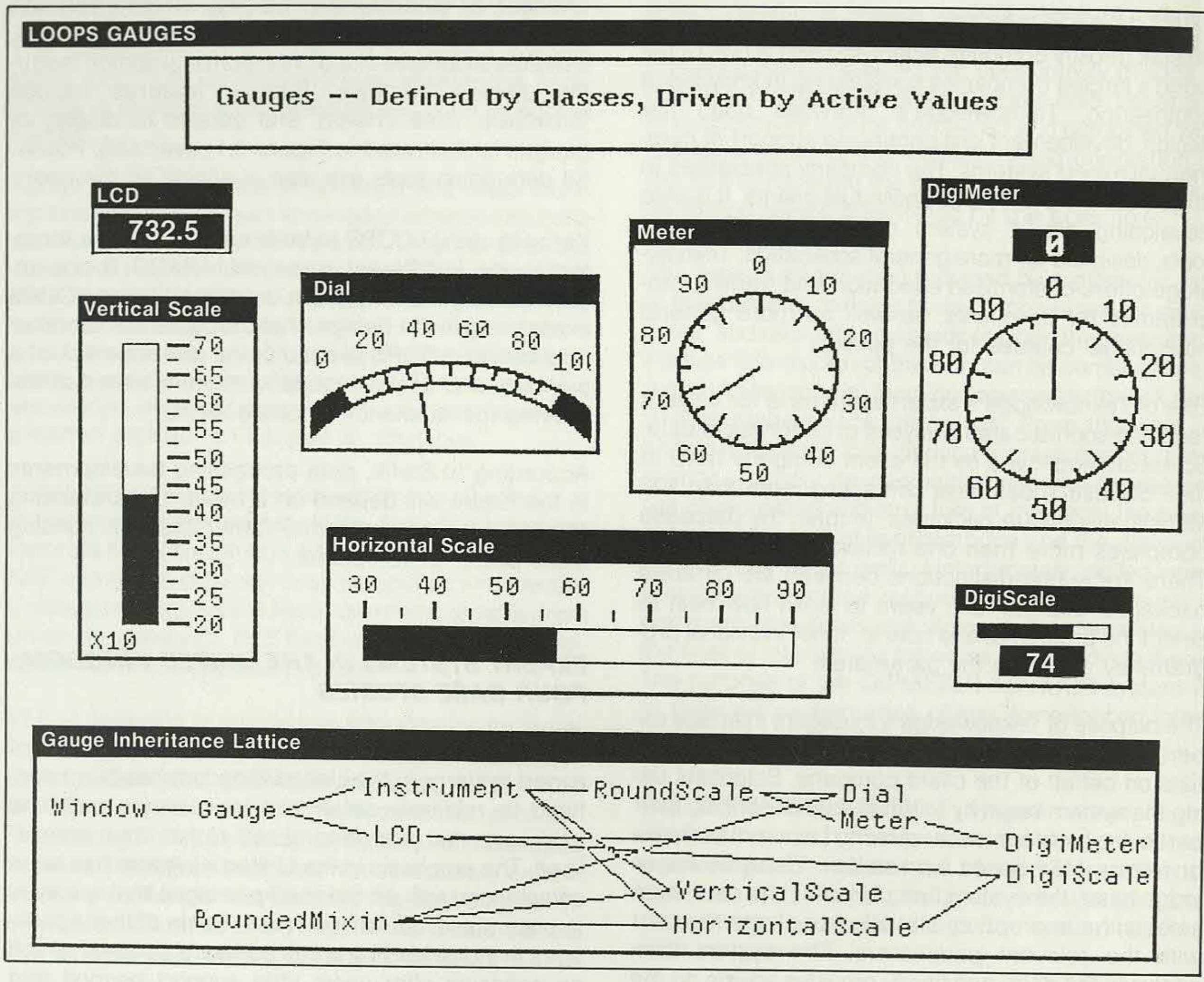
According to Stefik, data processing developments in the future will depend on a better understanding of programming problems, rather than on building more powerful computers.

EXPERT SYSTEMS IN THE UNITED KINGDOM: FOUR CASE STUDIES

In sharp contrast to the United States, experience of expert systems in the United Kingdom has been confined to relatively small and inexpensive software packages that are generalised rather than specialised. The emphasis in the United Kingdom has been on self-help, with off-the-shelf packages that are easy to understand, but limited in use. Some of these packages are provided by small software houses, which on occasion offer users little support beyond that necessary to implement the package on the user's host computer. Other packages are developed and provided by established software companies on behalf of clients.

Most expert systems packages are derivatives of the EMYCIN system described in Chapter 2. Packages from small software houses are typified by MicroExpert, supplied by ISIS Systems, and AL/X supplied by Intelligent Terminals Limited. SPL International supplies a similar package called SAGE, which has extended facilities. SPL supports SAGE through its knowledge engineering group and offers its clients both software development and support. The three packages offer varying degrees of flexibility and facilities. Prices range from $1,200 for a basic implementation of MicroExpert (on a microcomputer) to $30,000 for a full implementation of SAGE on a mainframe. While the suppliers agree on the supremacy of PROLOG as an expert system language, each of the three packages we have mentioned is written in a standard commercial programming language — a consequence of the present limitations of PROLOG.

Figure 3.1 Gauge display of LOOPS



PROLOG itself is available from two suppliers: Expert Systems Limited (whose product is PROLOG-1) and Logic Programming Associates (whose product is MicroProlog). Both can be implemented either on Z80 CP/M microcomputers, or on DEC PDP 11 and VAX minicomputers.

In contrast, ICL and RACAL Expert Systems provide specialised products. ICL introduced in June 1983 an expert system called CONSULT, operating on the PERQ workstation. CONSULT is a result of research and development in collaboration with British Steel on a fault diagnosis system; and is a microcomputer implementation of the resulting product. ICL also announced in mid-1983 the setting up of a new knowledge engineering group. RACAL Expert Systems was set up to develop and market a product which will assist in the interpretation of oil-well logs. The product is in direct competition with Schlumberger's DIPMETER ADVISOR system described on page 22.

On the other hand, Cambridge Consultants Limited and Tymshare UK regard expert systems as simply a new software tool to be used in appropriate applications. Cambridge Consultants develop bespoke systems for individual clients, while Tymshare offers expert systems building facilities as part of a decision support package called REVEAL which is available to users of the Tymshare time-sharing service.

We now describe four case studies that typify the experience with expert systems in the United Kingdom: that of BL Systems Limited with MicroExpert; the CCTA with MicroProlog; the Imperial Cancer Research Fund with in-house development; and Mars Group Services with MicroExpert.

BL Systems Limited

BL Systems Limited (BLSL), a wholly owned subsidiary of BL Cars Limited, provides specialist systems support for the parent company. BLSL's advanced systems department purchased the first version of MicroExpert in 1981, as part of a continuing policy of exploring and evaluating new developments in information technology. Since purchasing the product, BLSL has developed several different expert systems, two of which have been demonstrated to our researcher.

The first of these systems is a diagnostic product for investigating the causes of a car failing to start. The knowledge for building the knowledge base was extracted from manuals and from the personal experience of relevant experts. At the time of writing in mid-1983, the knowledge base consisted of 90 rules. The way the system works is by first assuming a type of fault (the goal), then exploring all the conditions necessary for this fault to be substantiated. The process is carried out in an interactive question-and-answer session, during which the user supplies the system with additional information where necessary (see Figure 3.2). The consultation process is laborious and tedious.

The second expert system demonstrated by BLSL has been designed both to advise and to assist motor car sales people by providing them with information on which of 21 models in the Allegro range of small family cars is best suited to a particular customer. Interestingly, ethical issues have been highlighted in this development. These issues relate to the question of whether a sales person should use the system to convince a customer of the suitability of a specific model, when the underlying logic could be designed to select the model that the sales person prefers to supply. The system's knowledge base consists of 68 rules extracted from experts in this field (domain).

At the time of writing the potential users of these two systems were not enthusiastic about them, for two reasons. One was the poor user interface. The other and more important reason was the simple fact that neither system provided its users with genuine advantages over what they had beforehand (manuals, charts and their own personal experience). The view of our interviewee, Brian Johnson, was that the current generation of commercial expert system packages is suitable only for experimenting and learning, and not for building commercial products.

Figure 3.2 Example of dialogue exploring the cause of a car failing to start

The current goal is whether or not THERE IS A FAULT IN THE ELECTRICAL SYSTEM. To explore this goal — type [Y]. Type [?] for options. $\boxed{Y}$

How certain are you that THE STARTER MOTOR TURNS THE ENGINE AT NORMAL SPEED [– 5 . . 0 . . 5]? $\boxed{W}$

Note: The user types "W" to ask why this question is being asked.

THE STARTER MOTOR TURNS THE ENGINE AT NORMAL SPEED — being true greatly weakens, being false greatly strengthens the hypothesis that:— THERE IS A FAULT IN THE ELECTRICAL SYSTEM.

How certain are you that THE STARTER MOTOR TURNS THE ENGINE AT NORMAL SPEED [– 5 . . 0 . .5] ? $\boxed{5}$

Note: Answers to this type of question are given using an 11 point scale. 5 is definitely true, – 5 is definitely false and 0 represents don't know.

Is it the case that THE CAR HAD BEEN RUNNING BUT CUT OUT [Y . . ! . . N] ? $\boxed{N}$

This goal was whether or not THERE IS A FAULT IN THE ELECTRICAL SYSTEM. Certainty Factor is – 5.00. Answered.

Note: This message is saying that there definitely is not (– 5) a fault in the electrical system. "Answered" indicates that all relevant questions about this goal have been asked.

The current goal is whether or not THERE IS A FAULT IN THE IGNITION SYSTEM. To explore this goal — type [Y]. Type [?] for options.

Note: The system continues until all goals have been investigated or the user wishes to end the session. At the end of the session a report is produced.

Note: □ indicates user answers.

Central Computer and Telecommunications Agency (CCTA)

The CCTA is part of the United Kingdom Government Treasury department. The broad purpose of the Agency is to encourage the use of computing throughout central government, with the aim of improving efficiency and effectiveness. The CCTA has only limited facilities itself, but provides for the government a central focus for expert knowledge of developments in computer-related technology, both inside and outside central government.

Within the CCTA, the Future Concepts Branch is responsible for evaluating new developments in information technology, and expert systems are one of its areas of interest. The aim of the Branch is to learn enough about expert systems to promote and support their use within government departments at the appropriate time.

One of the experimental applications of expert systems in government departments was in prototyping a system for assessing social benefits in the Department of Health and Social Security (DHSS). The questions of who is entitled to a benefit, what form a benefit should take, and what the value of the benefit should be, are complex ones involving a large number of selection criteria which change at frequent intervals. At present, minor amendments to the DHSS system specification can mean many person-years of programming effort.

Using MicroProlog, however, 90 per cent of the rules were written by DHSS staff (with assistance from Imperial College) in a mere 10 days. Modifying the knowledge base is easy, said Mr Owsianka, because of the independent nature of the structure. His belief was that this application provides a convincing demonstration of the power of the PROLOG language for building an expert system prototype of a large computer system. "You do not need to have a complete design and specification before you start writing the application program — it is easy to expand, modify and refine the knowledge base as you go along". The CCTA hopes to apply the lessons and experience gained in this DHSS development in other government departments.

Imperial Cancer Research Fund

The Imperial Cancer Research Fund (ICRF) is a charity supported entirely by private funds. Ninety-three per cent of the money collected is spent directly on research on cancer, providing continuing support for over 800 scientists and associated workers in ICRF's laboratories and hospital-based units.

Within the ICRF, a small team of experts (numbering up to seven, with backgrounds in artificial intelligence, systems analysis and programming) is conducting research and development work on knowledge-based systems and their application in cancer-related fields. The work is being undertaken on a DEC 20 computer, with a view to eventually implementing some applications on microcomputers.

One of the applications that has now reached an advanced stage is the Terminal Care System (TCS). The purpose of TCS is to advise general practitioners on the management and care of patients suffering from terminal cancer. Treating patients at this stage is a difficult and stressful problem. Most general practitioners are uncertain both of the nature of the disease, and of how to manage the practical aspects of ordering the appropriate medicines.

TCS was developed in-house and is written in PROLOG. It has a knowledge base containing some 200 rules that have been established during consultations with the relevant domain experts.

A second application of expert systems developed by the ICRF aims to build an intelligent interface to the Fund's existing database of worldwide cancer statistics. Also written in PROLOG, this intelligent interface has a knowledge base containing logical relationships and rules about factors such as types of cancer disease, incidence of the disease amongst different ethnic groups and so forth. The intelligent interface, in attempting to answer a specific question in the absence of relevant information, will instead supply information on topics most closely related to the original question.

According to our interviewee, Dr John Fox, organisations wishing to progress in the field of expert systems should expect to invest substantially, and to make a long-term commitment to PROLOG, in order to reap the benefits. (A practical expert system takes at least one year to develop.) And the expertise of the development staff should cover four areas of skill: artificial intelligence, domain expertise, computer systems development and management.

Mars Group Services

Mars Group Services is a wholly owned subsidiary of the Mars Group, providing specialist systems support for the confectionery-manufacturing parent company.

Following a preliminary evaluation of expert systems, staff at Mars Group Services decided that the best way to further advance their understanding of the subject was to conduct an experimental, in-house trial. The main objectives of the trial could be achieved, it was thought, by experimenting with the least expensive system on the market, MicroExpert.

For simplicity, a mainframe version of this system was purchased.

The expert system developed was a diagnostic system, designed to identify the cause of faults in the communications network used to interlink the company's distributed computing system. This particular domain was selected primarily for an opportunistic reason: the system developer, Sal Pinto, also possessed the domain expertise. The combination of skills in one person simplified the task of developing the expert system.

According to Sal Pinto, developing the system took only five days, and was as easy as it would have been if the system had been written in Basic. By mid-1983 the system was being used successfully by the computer operations staff to locate faults on the communications network. An important advantage of the expert system (as opposed to a conventional system) has proved to be the way it has obliged the developer to structure the knowledge in a form that makes it both comprehensive and accessible to others.

EXPERT SYSTEMS IN FRANCE: TWO CASE STUDIES

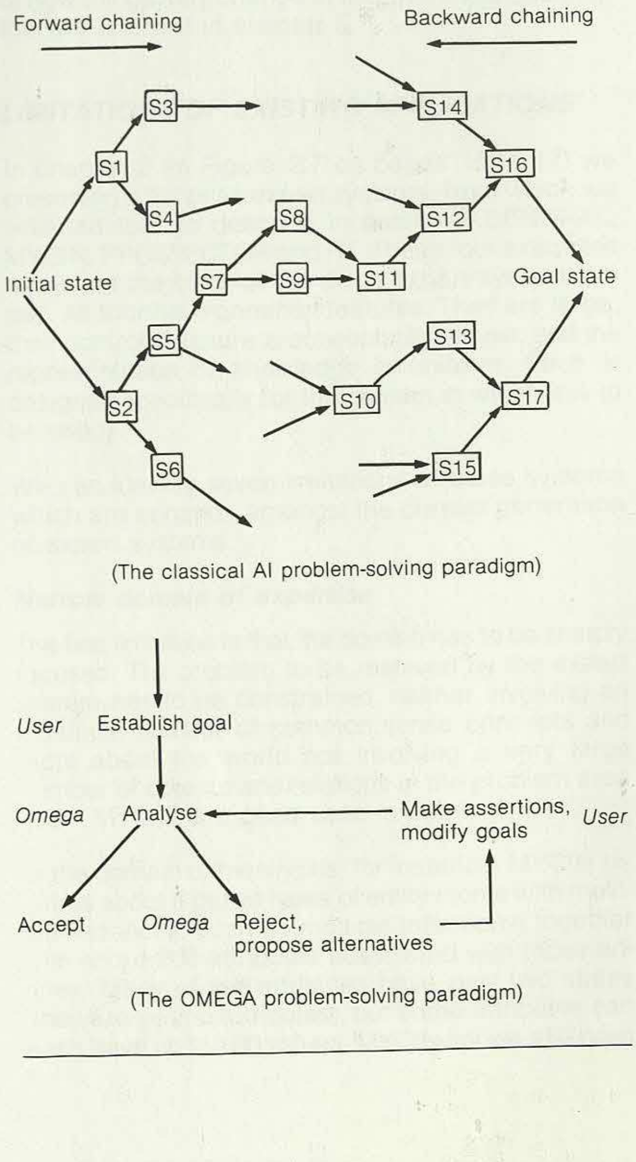
In France as in the United Kingdom, experience with expert systems is somewhat piecemeal, and on a scale that is far smaller than in the United States. In this section we describe briefly two expert systems, the Kayak project and the Reseda project.

The Kayak project

The Kayak project, which is funded by the French government, is being undertaken at INRIA, the French National Institute for Research into Information and Automation. Kayak's domain is office automation, and its purpose is to support work in the office. Kayak draws on and extends ideas from artificial intelligence research. The work has stressed the nature of office work as group activity, rather than merely a collection of individual tasks. Kayak attempts to capture not only the expertise embodied in individual tasks, but also expertise about how the tasks interact and how they contribute in aggregate to the solution of organisational problems.

As well as helping to explore the nature of office work, Kayak is aimed at researching a highly specialised multifunction office workstation, which will itself be able to support the Kayak system. Dr Gerald Barber, who works on the project at INRIA's headquarters in Rocquencourt, said that a demonstration version of the prototype pilot workstation should be running by the summer of 1983. The eventual aim is to produce commercial products.

Figure 3.3 Comparison of the classical artificial intelligence paradigm with the OMEGA paradigm



A special knowledge-representation language is being used to develop the system. The language, called OMEGA, is written in LISP. OMEGA provides a uniform framework within which to implement tools to support an office worker's problem-solving activities. The different tools can co-operate to achieve the goals of a particular office task. Knowledge is embedded in the form of descriptions about objects and their relationships, and it provides the basis for OMEGA's reasoning mechanism. This reasoning mechanism (paradigm) of OMEGA is different from the classical problem-solving paradigm of artificial intelligence. In the problem-solving support paradigm of OMEGA, the office worker establishes a goal, for example to send a message or to complete a step in an office procedure. Based on what OMEGA knows about a goal, it tries either to establish the goal or to refute the goal. If it is not possible for OMEGA to establish the goal, the system notifies the office worker that the goal cannot be established, or that contradictory information has been discovered during the attempt to establish the goal.

At this point the office worker can either modify the goal or make further assertions, possibly supplying information necessary to establish the goal. OMEGA then attempts to establish the goal. This cycle continues until the goal is established. The problem-solving support paradigm of OMEGA is illustrated in Figure 3.3.

The Reseda project

The Reseda project is being developed at the Centre National de la Recherche Scientifique in Paris, where a team of three to five artificial intelligence researchers led by Professor Zarri have been developing techniques for information retrieval.

Reseda itself is an expert system equipped with 'deep level' reasoning abilities in the field of complex biographical data management. The specific domain of Reseda is French history between 1350 AD and 1450 AD.

Through its inference mechanism, Reseda is able to deal with incompleteness in the information with which it is supplied at either of two levels. At the first level, the system carries out what is termed 'transformation', or paraphrasing of the original query. It generates a 'search model' in order to increase the chances of matching the required data. In this case, the information is presumed to exist. The advantage of this approach lies in the greatly increased efficiency of the database, which may continue to be organised in a traditional way (which is relatively simple and logical).

At the second level of incompleteness of information, Reseda attempts to reconstruct missing information. The system is capable of finding plausible explanations for certain known facts in the database by establishing 'casual' links with other known facts. The casual linking is carried out by a class of inference procedure called the 'hypothesis'.

To illustrate this we give the following example: the system is asked to explain why a person x was given an influential post by the administration. The system will check to see what coincided with this appointment. It establishes that a new person y was appointed to head the administration immediately preceding x's appointment. The system also establishes that x used to work for y in a previous administration. The plausible conclusion is that y asked x to work for him in the new administration.

The difference between Reseda and more commonplace forms of expert system lies in the database. The biographical facts which the system is able to process and store are both permanent and extensive. This contrasts with many other expert systems, where the information to be interpreted is introduced at the time of processing.

Reseda is written in APL (later versions may use an INTERLISP-D environment), and developed on large Amdahl and NEC mainframes. Professor Zarri told our researcher that Reseda is now being used as the basis of two further prototypes for demonstrating other potential applications. The domain of the first prototype is the invasion of Czechoslovakia by Germany in 1939. The domain of the second prototype is the Falkland Islands conflict of 1982. These developments have both been undertaken for a private company. Professor Zarri himself has made plans to adapt Reseda to manage medical files.

CHAPTER 4

POTENTIAL APPLICATIONS OF PRESENT-DAY EXPERT SYSTEMS

In the preceding chapters of this report we have explained what expert systems are and what they are used for, and described briefly the experience of some of the pioneering users. In this chapter, we consider the question of what present-day expert systems are best suited for. We begin by summarising the limitations of existing applications. That leads us to review the characteristics of potential applications, and then to identify the resources — staff, hardware and software — needed by an organisation intending to develop an expert system for its own internal use. The whole of the chapter is concerned with what is available and what can be done today. The subject of how things may change in the future is a question that we address in chapter 5.

LIMITATIONS OF EXISTING APPLICATIONS

In chapter 2 (in Figure 2.7 on pages 15 to 17) we presented a list of 41 expert systems, from which we selected four to describe in detail — DENDRAL, MYCIN, PROSPECTOR and R1. These four examples represent the state of the art in expert systems today. All four have common features. They are large, their control structure is conceptually simple, and the representation of knowledge is uniform. Each is designed specifically for the domain in which it is to be used.

We can identify seven limitations of these systems which are common amongst the current generation of expert systems.

Narrow domain of expertise

The first limitation is that the domain has to be sharply focused. The problem to be resolved by the expert system has to be constrained, neither involving an indefinite number of common sense concepts and facts about the world nor involving a very large number of objects and relations in the problem area itself. MYCIN is a good case in point.

In the domain of meningitis, for instance, MYCIN requires about a dozen types of entity (some with multiple instances, such as multiple infections) together with about 200 attributes associated with those entities. Many of the attributes have only two states (they are yes/no attributes), but some attributes can each have up to 100 values. MYCIN knows 450 rules that relate sets of object-attribute-value triplets, and another 500 to 1,000 individual facts stored as definitions (such as "E Coli is gram — negative"); as lists (such as a list of normally sterile sites); and as relations (such as "the prescribed drug for streptococcal infections is usually penicillin"). The domain of expertise cannot grow too large, because efficient means for building and maintaining large knowledge bases are not currently available.

Developed and tuned over several years

The second common limitation of expert systems is the time that is needed to develop them. Most expert systems have been constructed laboriously by a team of specialists over several years. To help reduce the problem, some research groups have explored ways of automating the construction of knowledge bases. Others have tried to write routines that conduct a dialogue with an expert, for the purpose of extracting knowledge without the help of a knowledge engineer. So far, however, these activities have proven successful only when they have been able to build on an existing framework.

Large and expensive to build

Most successful expert systems have a knowledge base containing several hundred rules that has taken a team of systems builders and domain experts years to construct and organise. As a result, most expert systems have cost more than a million dollars to build. That excludes the cost of the very considerable effort needed to maintain and refine the system thereafter.

Few domains explored

Because of the expense and the associated risk involved in developing an expert system, the domains that have been explored so far have tended to be related either to the industry sectors (such as oil and chemicals) able to justify the investment in terms of the value of results, or to government-sponsored sectors (such as medicine and mineral exploration). The selection of domains has been influenced more by the availability of finance than by applicability.

Scarce and expensive expertise

Most of the best-known expert systems have attempted to capture expertise in domains where human experts are themselves both expensive and scarce, and where training takes many years. At the same time, the knowledge of these domains has been widely available, having been published over the years in text books and reference books. These characteristics apply to domains such as medicine, geology and chemistry.

Stylised explanations

The sixth common limitation of expert systems is the explanation facility (or human window). Despite the acclaim that has sometimes surrounded it, the explanation facility actually offers little insight into an expert system's way of reasoning. MYCIN, for instance, lists all the rules needed to be "fired" in order to reach a goal, to explain why it needs a piece of information. It does so in the same way for every user. In response to a request from a user to explain how a conclusion has been reached, MYCIN merely provides an execution trace.

Scant built-in knowledge of scope and limitations

Neither the utility programs for constructing knowledge bases, nor the reasoning programs themselves, contain much knowledge about their own assumptions and limitations. This explains why most present-day expert systems can be used only by experts who already have gained sufficient appreciation of the systems' knowledge and limitations to enable them to interpret its conclusions sensibly.

POTENTIAL APPLICATIONS

During our discussions with Richard Duda, he offered two recommendations to organisations considering developing their own expert systems. First, he advised against choosing an expert system to solve a problem if a traditional computer system could be used instead. Second, in choosing a problem for solution by an expert system approach, he advised in favour of selecting a problem best suited to this approach, rather than one most desired by the organisation.

Duda's comments, together with the limitations we have already discussed in this chapter, enable us to identify the following five characteristics of potential applications of expert systems:

- —Knowledge and data about the domain will largely be available already in the form of published literature, internal reports and files. This information will be documented in a factual rather than procedural way.
- —Some of the information associated with the domain will be imprecise, based on experience gathered over the years by experts in the field.
- —The domain itself will be associated with high-cost operations, and the benefits to be gained from solving the problems will be sufficient to justify the risk inherent in developing an expert system.
- —The problem will be divisible logically into stages, presenting an opportunity for stage-by-stage working in order to reduce investment and risks.
- —The knowledge associated with the domain will be substantial, yet the available experts will be expensive and limited in number.

This leads us to suggest three broad fields as potentially appropriate for applying expert systems: training, advice, and intelligent interfaces.

Training in special fields

Knowledge-based systems have an obvious contribution to make to computer-aided training in areas where the knowledge is highly specialised and difficult to acquire. The GUIDON system (see reference 12) developed by Clancy at Stanford University exploits the MYCIN knowledge base to teach both facts and problem-solving strategies to medical students. MYCIN's diagnostic rules are augmented by the addition of methods for guiding the dialogue with the student, presenting diagnostic strategies and responding to the students' initiative. We believe that many expert systems which were originally designed to provide advice will evolve to become successful training tools in specialised areas.

Advice

Providing advice is a very broad field. One example of using expert systems to provide advice is in fault diagnosis where, after a question-answer session, the system advises on the cause of faults and proposes remedies. Another example is in insurance broking, where the system can be fed with client details and advice on alternative policies. Tax advice is another field, where the expert system is fed with details about individuals and, for instance, advice on how to fill in tax returns. Yet another potential application area within the general field of advice is counselling citizens on their rights with regard to such things as social benefits, tax rebates and so forth. Decision support systems could probably grow in stature and usefulness following the application of expert system techniques.

Intelligent interfaces to existing databases and software

Using complex software is a very knowledge-intensive activity. Manuals are complex, filled with detail and easily forgotten. Often they are poorly written, containing merely the facts about the system and not the rules-of-thumb necessary for its use. Expert systems knowledgeable about particularly complex software systems could radically improve matters.

They would have an extensive knowledge base, able to interpret and fulfil user requests. The knowledge base would be provided by the system developers, and augmented by the user community itself. In this situation, the manuals of the past would become active and influential in the service of user needs. The market for expert systems of this sort would be as broad as the market for software.

Other special fields

The new (expert system) software techniques will be used to try and solve problems for which traditional techniques have proved inadequate. Voice recognition and natural language processing are two such areas. Also, Feigenbaum (see reference 5) believes that one of the important application areas will be computer games.

STAFF RESOURCES NEEDED

Having identified the application areas for which expert systems are best suited, we now discuss the question of resourcing, beginning with staff resources.

Most researchers in artificial intelligence and expert systems have emphasised the difficulty of acquiring the information necessary to structure the knowledge base. A key component is the need to make available domain experts who are able and willing to spend time developing and debugging the knowledge base. A problem that is frequently encountered is that the expert, once available, finds difficulty in expressing knowledge in a form that is acceptable for direct input to a knowledge base.

A common way of extracting knowledge from an expert is by careful and painstaking analysis, undertaken by a second, trained person called a knowledge engineer. Knowledge engineers are computer scientists skilled in knowledge-representation and inference techniques, and able to converse comfortably with the domain expert.

Feigenbaum (see reference 5) has stated that "knowledge acquisition is the critical bottleneck problem in artificial intelligence." Severe difficulties are often experienced in acquiring the relevant knowledge. The most difficult aspect of this process is to help the expert initially to structure the domain knowledge. The knowledge engineer takes an active role in the knowledge-acquisition process — interpreting and integrating the expert's answers to questions, drawing analogies, posing counter examples and raising conceptual difficulties.

Duda has confirmed this view (see reference 14). He has stated that, to construct a successful expert system, the following prerequisites have to be met:

—There should be at least one human expert involved, acknowledged as being well able to perform the task.

—The primary source of the expert's exceptional performance must be special knowledge, judgement and experience.

—The expert must be able to explain how to use the special knowledge and experience, as well as the methods used to apply expertise to particular problems.

Apart from the domain expert and the knowledge engineer, a third high-level resource is required to construct an expert system — a highly skilled computer systems designer. This follows from the emphasis that is universally placed on the design of the user interface. During his interview with our researcher, Read Smith of Schlumberger estimated that as much as 80 per cent of the programming effort in an expert system may be devoted to developing the user interface.

The view of John Fox of the Imperial Cancer Research Fund is that an expert system development team should consist not of three but of four people: the domain expert, the knowledge engineer, the senior system designer, and finally a skilled administrator. According to Dr Fox, the administrator is essential for co-ordinating and controlling the development effort, so that the task is properly managed.

Before we leave the subject of staff resources, it is important to emphasise the effort needed to maintain an expert system following its development. Maintaining a large knowledge base is every bit as difficult as constructing it in the first place. Because it is concerned with problems having no closed solutions, the knowledge base of an expert system will change as experts accumulate more experience and develop new techniques. In medicine, for instance, new measuring devices make it possible to detect new states and to quantify known parameters more precisely. New micro-biological agents are discovered, as well as new drugs to treat them.

Maintenance may mean actively searching through the knowledge base for problems that need attention. There may be gaps in the knowledge base where some of the many possible combinations of conditions are not covered. There may be overlapping items in the knowledge base, leading to inconsistency or redundant conclusions. Items may become outdated. An intelligent maintenance system should have the syntax and semantic knowledge needed to assign blame to the items in the knowledge base that appear to be responsible for poor performance. (This is because verifying the solutions using manual methods as in traditional systems is often impractical.) An intelligent maintenance system should also be able to suggest modifications.

The problems of maintaining a knowledge base become more difficult when two or more domain experts contribute to it. Although several physicians contributed to MYCIN, only one physician made changes at any one time. Recommendations for change were routed through an administrator, whose task it was to maintain consistency.

HARDWARE AND SOFTWARE RESOURCES NEEDED

Because both constructing and experimenting with an expert system is very expensive, there is already a discernible trend towards dedicated symbol-processing workstations and software design tools. We have already mentioned two of these tools in chapter 3 — LOOPS from Xerox PARC and KEE from IntelliGenetics.

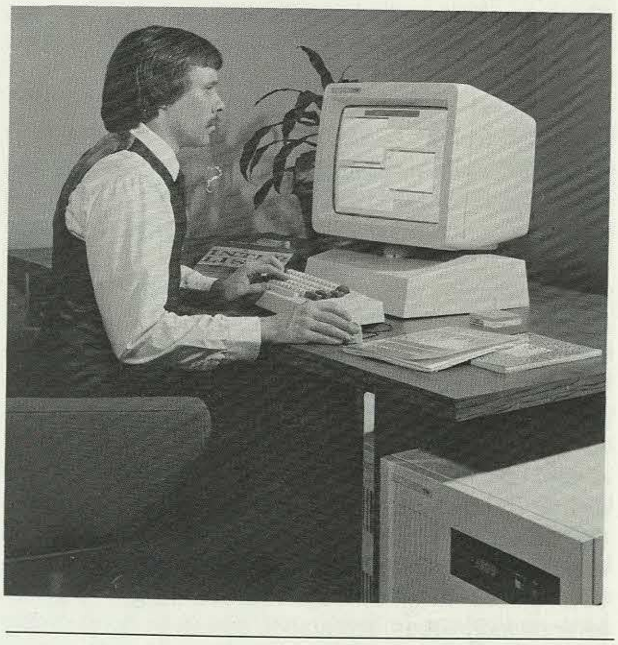
Other similar tools are OPS5, developed at Carnegie-Mellon University in association with DEC; AGE developed at Stanford University (see Reference 15); and XPRT, developed at MIT (see Reference 16). These are all tools that help someone to design and build an expert system within an existing framework, and to provide an efficient human interface. The Xerox 1100 series of personal computers, which provide advanced interactive graphics, can also serve as effective workstations for developing expert systems when equipped with software such as INTERLISP-D. The Xerox 1100 workstations enable users to display text in multiple fonts, manipulate raster images, display multiple windows, provide menu-driven selection and offer a wide range of graphic utilities — all controllable through a mouse or the keyboard.

Other examples of such LISP-based workstations are the Symbolics L-3600 and the LMI LISP machine (both derivations of the MIT LISP Machine development). Specialised workstations for expert systems are, however, very expensive. Prices for the Xerox 1100 workstation, for instance, range from $45,000 to $180,000. A Xerox 1100 workstation running INTERLISP-D is illustrated in Figure 4.1.

Taking into account both the specialist human resources and specialist hardware and software needed to develop an expert system, our opinion is that the total cost is unlikely to be less than half-a-million dollars. One million dollars and upwards is likely to be a more realistic estimate in most cases.

Figure 4.1 Xerox 1100 workstation

The Xerox 1100 Scientific Information Processor is a personal computer able to run the INTERLISP-D language. The workstation features a main memory of up to 1.5M bytes, a 43 cm diagonal CRT display with high resolution (1024 x 808 pixels), 64 button keyboard and 3 button mouse.



CHAPTER 5

EXPERT SYSTEMS IN THE FUTURE

In the preceding chapter we were concerned with the characteristics of present-day expert systems and applications, and the current availability (or unavailability) of skilled resources. Now we turn to the future, to question how both the techniques of expert systems and the skills of human resources are going to change, and what the impact of the changes might be.

THE NEED FOR IMPROVEMENTS

At present, expert systems are narrowly focused to suit the specific needs of particular kinds of user in highly specialised and narrow domains. Before expert systems can become more generally useful and cost-effective, some significant improvements have first to be made. We now consider the nature of these improvements.

Improved acquisition of knowledge and better understanding of natural languages

The first improvement that is needed concerns the acquisition of knowledge and the understanding of natural languages. We have linked these two topics together because most researchers in artificial intelligence agree that the problem of knowledge acquisition will not be fully resolved before we have gained a better understanding of how to process natural languages. At present, acquiring knowledge from an expert is a lengthy, laborious and costly process. What is more, the result is an interpretation of the expert's knowledge by a knowledge-engineering intermediary, and it may not be totally accurate.

Knowledge acquisition is a never-ending process. As we have mentioned, the knowledge base has continually to be refined in order to improve system performance. To illustrate this point, the knowledge base of R1 grew in a period of only two years from 500 to 2,500 rules. To help with the problem, some expert systems have been extended by the construction of their own specialised knowledge-acquisition systems — expert systems in their own right. Two examples of knowledge-acquisition expert systems are META DENDRAL for the DENDRAL expert system, and the TEIRESIAS knowledge-acquisition system for the MYCIN expert system.

Richard Duda has estimated the effort required to assemble the knowledge base for PROSPECTOR as 100 man-years. Even though most of the information is available in published form, a large effort is still required.

The best long-term solution would be for the expert system to create the rules it needs from raw, unformatted data fed in directly by an expert (thereby short-circuiting the knowledge-engineering function). For example, a doctor could type his knowledge into a MYCIN-type system, which would then proceed to create from the raw material the structured rules required in the knowledge base.

Better understanding of how to represent knowledge

The second improvement needed before expert systems can be used more widely is a better understanding of how to represent knowledge. Research into artificial intelligence has yielded a number of knowledge-representing techniques (see page 10). The production-rule technique is the most common, not because it is the best but because it suited best the chosen domains at the time — namely medicine and geology. An important reason why the applications of expert systems have been constrained to narrow domains is the limited understanding of how to represent knowledge. With narrow domains, it is sufficient merely to make use of narrowly based knowledge applied with a specific technique.

Better understanding of how to deal with uncertainty

Dealing with uncertainty is one of the factors that distinguishes an expert system from an intelligent question-and-answer system. The techniques that have been developed to deal with uncertainty and ambiguity however, are still immature. This is despite continuing research into inexact reasoning, and despite the relatively recent development of techniques such as the method of certainty factors used by MYCIN (explained in chapter 2 on page 17). The fact is that today's techniques of inexact reasoning are not theoretically sound. That is why they tend to be used on the basis of trial and error. If one technique yields results that seem reasonably satisfactory for the problem in question, it is accepted. If the technique does not yield satisfactory results, either it is adjusted or an alternative is adopted.

Need to understand the process of human judgement, reasoning and perception

At the present time, researchers in artificial intelligence have little understanding of human judgement, reasoning and perception. That precludes any possibility of building practical expert systems in domains where human understanding, reasoning and perception are of prime importance. Today's generation of expert systems manipulate facts and rules-of-thumb in mechanical ways. That explains why the so-called explanation facility (or human window) is merely a trace (or execution list) of the rules that have been used in the process of searching for a solution.

Better ways of identifying domains

The techniques of present expert systems — types of knowledge, reasoning and inferencing — are not generalised techniques. Rather, they have been developed with very specific problems in mind. Identifying further domains appropriate to today's limited techniques is a difficult practical problem. Put another way, expert systems are looking for users; techniques are looking for applications. The problem underlines the risk of developing an expert system, when it is hard to tell at an early stage whether or not the system eventually will work.

More trained and skilled people

In the main centres of expert system development both in North America and Europe there is general agreement over the need for more highly trained people. Most sought-after are people who combine a background in artificial intelligence or computing science (preferably with PhD qualifications) with a deep understanding of methods of acquiring and representing knowledge. Too few qualified people, rather than too little money, now seems to be the problem — both in Europe and in the United States. In the United Kingdom, for instance, the money that has been made available under the Alvey programme for accelerating research and training in knowledge-based systems is unlikely to be fully allocated. At Carnegie-Mellon University, Professor McDermott warned our researcher of his concern that commercial organisations will attempt to develop expert systems before suitably qualified staff are available, with the inevitable result of unsatisfactory systems and a spate of job-switching by inadequate staff.

Lower-cost hardware

All the most significant current-generation expert systems have been developed on costly hardware — either large mainframes, powerful minicomputers (such as the DEC VAX), or on dedicated workstations able to run LISP or its derivatives. LISP workstations are favoured in the United States for developing the newest commercial expert systems because, users claim, they improve productivity. But workstations of this type cost between $45,000 and $180,000, enough to make even the most richly funded European organisation think twice.

REALISING IMPROVEMENTS

We have described briefly seven areas in which significant improvements have to be made before expert systems can become more cost-effective and more generally usable. Of the seven, only the last improvement (lower-cost hardware) seems virtually certain to be attained. New microchip technology will enable suppliers to design machines for special applications at a fraction of the cost of the present general-purpose workstations.

As for the remaining six improvement areas, we asked Richard Duda for his views on whether and when they might be achieved. "The proper goal is to be able to type text manuscript direct into the terminal, then have a program process the text, understand it and create the rules from it". It was an optimistic view, thought Duda, that envisaged this goal being reached within the next 20 years. Success would depend on the progress of researchers into artificial intelligence. If they were able to advance as rapidly as they hoped in new fields of research such as induction and perception, then today's relatively limited expert systems truly mark the start of a revolution. On the other hand, if the researchers fail to make the hoped-for progress, then expert systems may turn out to be little more than a passing fad.

Buchanan (see reference 1) points out how early we are in the development of expert systems. He explains:

"Artificial intelligence (and expert systems) is still very much in the so-called 'natural-history' stages of scientific activity in which specimens are collected, examined, described and shelved. At some later time, a theory will be suggested that unifies many of the phenomena noticed previously and will provide a framework for asking questions. We do not now have a useful theory. The vocabulary that we use to describe existing systems is, however, more uniform and useful than it was a decade ago. And the questions that we pose in the context of one programme are sometimes answered in another.

"Expert systems will provide many more data points for us over the coming years. But it is up to everyone in the field to do controlled experiments, analyse them, and attempt to develop a scientific framework in which we can generalise from examples. At the moment we ourselves lack the vocabulary for successful codification of our own data."

Despite the limitations and the rather primitive state of expert systems at the time of writing this report, it is quite clear that some organisations could benefit from this new technique. These organisations must be prepared to invest significant amounts of time, scarce resources, and money in a long term development effort. Only then will they benefit from the application of this specialised technique in areas such as training, advice and interfacing with existing databases and software (see chapter 4).

The companies supplying expert systems will concentrate their effort in the next few years on improving the cost-effectiveness of their software products. One product that in our opinion is already pointing the way is the general-purpose LOOPS system under development at Xerox PARC, which we described on page 23. A second product in this category is the specialised office workstation that is evolving from the Kayak project in France, which we described on page 26.

The problem of identifying suitable domains is probably the most significant cause of the high risk of expert systems development. It is only after substantial development work that one can tell whether the specific technique chosen is appropriate; whether all the necessary knowledge can be acquired; what the final size of the knowledge base is likely to be; and whether it can be organised efficiently. Many developments of expert systems have (and will have) to be stopped at a fairly late stage, when it is realised that the technical problems cannot be overcome.

During the next five years we expect to see some progress towards alleviating the shortage of skilled staff. As recognition of the importance of artificial intelligence grows, and as it receives more public funding, so the study of artificial intelligence, expert systems and knowledge-based systems will attract a growing body of students at universities and other centres. At the same time, companies specialising in knowledge engineering will offer more training courses aimed at helping business-people to understand the subject, and how best to exploit it.

As we have made clear in this report, most work on expert systems, in terms both of basic research and commercial application, is being undertaken in the United States. The body of knowledge existing in the United States, as well as the breadth and depth of resources there, is several years ahead of what is available in Europe, particularly in the field of commercial applications. In our opinion, the gap will widen rather than narrow. If the advances that are needed in the field are going to emerge, it seems most likely that they will come from the United States.

CHAPTER 6

MANAGEMENT GUIDELINES

For the next two or three years, the limitations described in chapter 4 are going to dictate the types of expert system that will be developed. One type of system will be for very large applications in a narrow and specialised field. These systems will usually be tailored for individual companies and will cost more than $1 million. Very few companies will be able to afford a high-risk investment on such a scale. Another type of system will consist of very small experimental applications of mainly educational value. These systems will be developed by organisations wishing to get first-hand experience of the new techniques. The systems typically will be unsophisticated prototypes and will cost more than $30,000. Most of these developments will be driven by curiosity rather than by genuine need.

Although many organisations will try a small-scale experiment before attempting full-scale development, a substantial investment is usually needed to achieve significant benefits. Companies should therefore bear in mind the size of commitment necessary to develop a full expert system application.

Looking further ahead, more sophisticated software tools and cheaper hardware will reduce software development costs substantially. Accumulated experience will reduce the risk factor associated with such developments. More organisations will then undertake large-scale developments, mainly of advice and fault-diagnosis type systems.

The following guidelines, together with the potential application areas identified in chapter 4, will help Foundation members wishing to explore the new expert system techniques. Applications should be considered in the following circumstances:

- Where knowledge is already available in some written form. (This knowledge will usually be held in a non-procedural form.)
- Where the application area calls for continuous update of logic rules.
- Where a system can be developed in a modular way, thus reducing substantially the initial outlay, and therefore the risk.
- Where there is a clear incentive for a user to use the system.
- Where a user is going to be able to maintain and continually improve the knowledge base.

Expert system developments that fall outside these guidelines are likely to be very high-risk ventures for the foreseeable future.

CONCLUSION

The present dominance of the United States in artificial intelligence research and development will continue for at least the next decade. Nevertheless, we expect some Japanese and European suppliers to develop leading expert systems products in specific fields.

Expert systems will not revolutionise data processing during the next five years. At the time of writing this report, expert systems can be regarded only as a new software technique, still at a primitive state of development. The critical shortage of skilled resources and the knowledge acquisition problem will retard progress.

Some pioneering organisations will undertake the costly, high-risk development of commerical expert systems, and some of these organisations will realise substantial benefits in very specific, key application areas.

BIBLIOGRAPHY

1. Buchanan, B. G., New research on expert systems, *Machine Intelligence 10.* Edited by Ellis Horwood, 1982.

2 Bramer, M. A., A survey and critical review of expert systems research, BCS 81, *Information technology for the '80s.* Proceedings of the conference, London, 1-3 July, 1981. Heyden, 1981.

3 Jones, K. S., Intelligent knowledge-based systems: papers for the Alvey committee, University of Cambridge computer laboratory.

4 Pergamon Infotech's State of the Art Conference, Japan and the fifth generation.

5 Feigenbaum, E. A., *Knowledge Engineering for the 1980s,* Edward A. Feigenbaum, Professor of Computer Science, Stanford University, Stanford, California, USA.

6 Michie, D., Expert Systems, *The Computer Journal,* Volume 23, Number 4.

7 Stefik, M., et al. *The Organisation of Expert Systems: A Prescriptive Tutorial,* Xerox, Palo Alto Research Center.

8 Newell, A. & Simon, H. A., Computer Science as empirical enquiry: symbols and search, *Communications of the ACM,* March, 1976.

9 Clark, K. L. & McCabe, F. G., Prolog: a language for implementing expert systems, *Machine Intelligence 10,* edited by Ellis Horwood, 1982.

10 Feigenbaum, E. A., Themes and case studies of knowledge engineering, *ESMA,* pp. 3-25, 1979.

11 Heuristic Programming Project, Stanford University Heuristic Programming Project. (A 'prospectus' of the various systems developed by the Stanford H.P.P.), 1980.

12 McDermott, J., R1: An Expert in the computer systems domain, *Proceedings of the First Annual National Conference on Artificial Intelligence,* 1980.

13 Glancey, W. J., Shortliffe, E. H. & Buchanan, B. G., Intelligent computer-aided instruction for medical diagnosis, *Proceedings of the 3rd symposium on computer application in medical care,* 1979.

14 Duda, R. O. & Gusching J. G., *Knowledge-based expert systems come of age,* BYTE Publications Inc., Vol. 6, No. 9, September, 1981.

15 Nii, H. P. & Aiello, N., AGE (Attempt to Generalize). A knowledge-based program for building knowledge-based programs, *Proceedings of the sixth international joint conference on artificial intelligence,* Tokyo, Japan, 1979.

16. Winston, P. H., *Artificial intelligence,* Addison Wesley Publishing Company.

17. Quinlan, J. R., *Fundamentals in the knowledge engineering problem.* Introductory readings in expert systems, D. Michie (editor), Gordon Breach Science Publications, 1982.

18. Newell, A., *Intellectual issues in the history of artificial intelligence,* Department of Computer Science, Carnegie-Mellon University.

GLOSSARY OF TERMS

Artificial intelligence	A branch of computer science concerned with enabling computers to mimic the characteristics that make people seem intelligent.
Backward/forward chaining	Alternative control strategies used by an inference engine to derive a solution to a problem.
Bayesian probability	A probability theory exploiting the elementary theorem known as Bayes rule. This rule establishes a numerical relationship between an hypothesis and observed evidence.
Browser	A schematic representation (logic tree) — displayed on a screen — of the logical relations between rules in the knowledge base.
Certainty factors	A method for handling inexact information (experts' assessments), developed as part of the MYCIN system project.
Control strategy	The method the system uses to solve the problem presented to it.
Decision support system	A system which provides information to assist the manager in his decision making process, and also evaluates the consequences of a chosen decision.
Deterministic system	A system which is not dealing with uncertainty — all the logical conditions and relations are either true or false.
Domain	Application area.
Fuzzy logic	A method for handling inexact information by attempting to quantify non-numeric (value) judgements.
Horn clause	Horn clause subset of predicate logic is used to express symbolic information in a way that can be used to solve problems. It forms the basis for the PROLOG programming language.
Inexact reasoning	The art of good guessing. Reasoning with rules of expertise, good practice and knowledge of the field.
Inference engine	The problem solving algorithm (or rule interpreter) and its method of applying to the problem the domain knowledge in the knowledge base.
Intelligent knowledge based system	A system which uses intelligent inference procedures to apply knowledge to perform a task. Expert systems are a subclass of knowledge based systems.
Heuristics	Rules of thumb. Rules of expertise, good practice and knowledge of the field.
Hierarchy	A term used to describe the structuring of knowledge in the knowledge base.
Knowledge base	A database of knowledge in which both facts and heuristics are represented as individual elements of knowledge about a particular domain.
Knowledge engineering	The process of building a specific expert system including design of the system and knowledge acquisition.
LISP	A computer language based on the manipulation of symbols and symbol structures. LISP is the main language used in artificial intelligence research and commercial applications.
Logic programming	Programming by expressing facts, relationships and rules in logical statements. It is used in applications that require intelligent symbol manipulations.
Paradigm	An algorithm, used as a term for the reasoning mechanism of the inference engine.
Pattern matching	The process of testing the logical statements in the knowledge base against the data provided to find whether they are 'true', 'false' or require additional information.
Predicate calculus	A widely studied formal language of symbol structures. Some of its concepts are relevant to symbolic computing.
Probabilistic system	A system which uses uncertain rules and information as part of its knowledge base.
PROLOG	A logic programming language. It is a high level language capable of manipulating symbols and symbol structures, while providing extended facilities for expressing knowledge and using this knowledge in a reasoning process.
Rule based system	An expert system using 'IF-THEN' statements as a method for representing the domain knowledge in the knowledge base.
Symbol	A string of characters, which may be numeric or non-numeric, such as Apple, Table, Five, 3.14159, etc.
Symbol structure	A type of data structure containing symbols (also known as a list structure).
Symbolic processing	Manipulation of symbols and symbol structures. Also referred to as list processing.

Butler Cox & Partners Limited
Butler Cox House, 12 Bloomsbury Square,
London WC1A 2LL, England
☎ + 44 1 831 0101, Telex 8813717 BUTCOX G

Belgium & The Netherlands
SA Butler Cox NV
Avenue Louise – 479 – Louizalaan,
Bte – 47 – Bus.
Bruxelles 1050 Brussel
☎ (02) 647 15 53, Telex 61963 BUTCOX

France
Butler Cox SARL
Tour Akzo, 164 Rue Ambroise Croizat,
93204 St Denis-Cedex 1, France
☎ (1)820.61.64, Telex 630336 AKZOPLA

United States of America
Butler Cox & Partners Limited
P.O. Box 590, Morristown, New Jersey 07960, USA
☎ (201) 285 1500

Switzerland and Germany
Butler Cox & Partners Limited
Butler Cox House, 12 Bloomsbury Square, London WC1A 2LL
☎ (London) 831 0101

Italy
Sisdo BDASrl
20123 Milano – Via Caradosso 7 – Italy
☎ 86.53.55/87.62.27, Telex 311250 PPF MI

The Nordic Region
Statskonsult AB
Stortarget 9, 5-21122 Malmo, Sweden
☎ 46-401 03 040, Telex 127 54 SINTAB