The Application of Expert Systems in Libraries and Information Centres

The Application of Expert Systems in Libraries and Information Centres

Editor
Anne Morris

Bowker-Saur
London • Melbourne • Munich • New York

British Library Cataloguing in Publication Data
Application of expert systems in libraries and information centres.
I. Morris, Anne, *1950-*
025.00285633

ISBN 0 86291 276 8

Library of Congress Cataloguing-in-Publication Data
Application of expert systems in libraries and information centres / edited by Anne Morris.
p. cm.
Includes index.
ISBN 0-86291-276-8
1. Expert systems (Computer science) — Library applications. 2. Information technology. 3. Libraries — Automation. I. Morris, Anne, 1950- .
Z678.93.E93A66 1992
025.3'0285 — dc20

91-38751
CIP

Bowker-Saur is part of the Professional Division of Reed International Books

Cover design by Calverts Press
Typeset by SunSetters
Printed on acid free paper
Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire

About the contributors

Ralph Alberico

Ralph Alberico is currently head of the Undergraduate Library at the University of Texas at Austin. In addition to numerous articles, he is the author of *Microcomputers for the online searcher* (Meckler, 1987) and co-author of *Expert systems for reference and information retrieval* (Meckler, 1990). He is a contributing editor of the journal *Computers in Libraries* and serves on the editorial board of the electronic journal *PACS Review* (Public Access Computer Systems Review). Previous positions include serving as Head of Reference at James Madison University in Virginia and at Loyola University in New Orleans. He was educated at the State University of New York at Buffalo and at the University of Alabama.

Bill Black

Bill Black is Brother International Lecturer in Machine Translation at the Centre for Computational Linguistics, UMIST, Manchester. He studied Philosophy and Sociology, then Linguistics, at Leeds University, and before taking up his present post, he taught and researched in Computer Science at Teeside Polytechnic and UMIST. His current research includes work on robust human-machine dialogue in natural language and on automatic abstracting, supported by the CEC and the British Library.

Roy Davies

Roy Davies is the subject librarian for mathematics, computer science, operational research, and earth sciences at the University of Exeter. Previously he held professional posts at the Ministry of Defence and the Polytechnic of Wales. He has a BSc in chemistry and a postgraduate diploma in librarianship, both from the University of Strathclyde, and a postgraduate diploma in computer science from the Polytechnic of Wales. His publications include papers on the implications of AI for librarianship, qualitative data analysis, and the inference of novel hypotheses from the results of information retrieval.

Forbes Gibb

Forbes Gibb is currently Head of Department of Information Science at Strathclyde University. He is founding Editor of *Expert Systems for Information Management* and European Editor of *Online Review*. His research interests are concentrated on advanced techniques for text analysis, storage and retrieval, and the impact of telecommunications on business.

Anne Morris

Anne Morris is a lecturer in Information Processing in the Department of Information and Library Studies at Loughborough University. She has postgraduate degrees in ergonomics, information technology and mathematics and has written numerous articles on expert systems. Anne Morris is also co-author of *Human aspects of library automation* and co-editor of the *Directory of expert system tools.*

Alastair G. Smith

Alastair G. Smith is a lecturer in the Department of Library and Information Studies, Victoria University of Wellington, New Zealand where he specializes in information retrieval and in scientific and technical information work. He has a BSc in physics from the University of Auckland, gained a Diploma of the New Zealand Library School, and is an Associate member of the New Zealand Library Association. He has experience in patent information work, and held positions as a reference librarian and special librarian before helping establish two New Zealand online information services.

Gwyneth Tseng

Gwyneth Tseng is a Lecturer in the Department of Information and Library Studies at Loughborough University. After graduating in Physics she moved into academic library work and obtained a masters degree in Information Studies in 1976. Now her main teaching specialism is information retrieval. She has been professionally active and has published on a range of topics within the discipline of information and library studies, including a number of articles on online business database selection and use of expert systems.

Preface

The last decade has seen expert systems develop into one of the most successful fields of artificial intelligence. Commercial organiaztions on both sides of the Atlantic, have recognized he enormous potential of such systems and have forced the frontier of knowledge forward at a rapid pace. Expert systems, no longer confined to the research laboratory although still in an early evolutionary stage, have added a new dimension to information processing. In addition to executing complex computation, computers are now able to offer users advice and solve problems that would normally require human expertise.

Expert systems have been used very successfully in both industry and commerce, reputedly saving some companies millions of dollars a year. Against this background, it is not surprising that the library and information services (LIS) sector, traditionally at the forefront of computer technologies, has been researching, assessing and debating the likely impact of expert systems on the information professions in recent years.

The purpose of this book is to review the progress made so far in applying expert systems technology to library and information work. It is aimed at students, researchers and practitioners in the information or computing field who are keen to explore the potential of using expert systems in this area. No previous knowledge is assumed; a glossary of terms is provided for readers unfamiliar with expert systems jargon. Chapter 1 provides an overview of expert systems technology covering historical aspects and the link to its parent discipline – Artificial Intelligence, the characteristics and application of expert systems, and detailed guidance on the anatomy and development of such systems. Chapter 2 examines the use of expert systems technology to simplify online information retrieval. In particular, it looks at the functionality of intermediary systems, software which mediates between the searcher and remote online information retrieval systems, and gives selected examples which illustrate how expert systems technology has been used in their development. Chapter 3 focuses on the use of expert systems in reference work, describing models of the reference process and research that has been undertaken in this area. Chapter 4 looks at knowledge-based indexing and the need for new approaches to information storage and retrieval. The next Chapter examines the links between rule-based systems, natural language processing and abstracting. It is concerned with the linguistic aspects of the process of accessing sources of information, and with how rule-based techniques, such as those used in expert systems, can be used to facilitate the process. Chapter 6 reviews the progress made in applying expert systems technology to cataloguing. The final Chapter attempts to predict the impact of expert systems and AI on libraries over the next ten years. Five areas are considered: knowledge media, knowledge industries, knowledge institutions, modes of discourse and implications.

The editor and the publisher are pleased to receive suggestions and observations regarding this book's contents and usage.

Anne Morris

Contents

Chapter 1

Overview of expert systems

Anne Morris

1 Introduction

Expert systems are computer-based systems that use knowledge and reasoning techniques to solve problems that would normally require human expertise. Knowledge obtained from experts and from other sources such as textbooks, journal articles, manuals and databases is entered into the system in a coded form, which is then used by the system's inferencing and reasoning processes to offer advice on request. Expert systems belong to the broader discipline of artificial intelligence (AI) which has been defined by Barr and Feigenbaum (1981) as: 'the part of computer science that is concerned with designing intelligent computer systems, that is, systems that exhibit the characteristics we associate with intelligence in human behaviour – understanding language, learning, reasoning, solving problems, and so on'.

AI, as a separate discipline, started in the 1950s when it was recognized that computers were not just giant calculators, dealing with numbers, but logic machines that could process symbols, expressed as numbers, letters of the alphabet, or words in a language (Borko, 1985). Since then AI has grown rapidly and today encompasses not only expert systems but many different areas of research including natural language understanding, machine vision, robotics, automatic programming and intelligent computer-aided instruction.

The first expert systems, described in more detail later, were developed by researchers from the Heuristic Programming Project at Stanford University, California, led by Professor Feigenbaum. The success of these systems combined with the increase in computer processing power gave encouragement to other researchers and created a steady interest in the commercial sector. Most of this early work occurred in the USA. The Lighthill report of 1969, which was sceptical about the value of AI, effectively stopped all UK and government funding for work in this field during this period (Barrett and Beerel, 1988).

The turning point came in 1981, when the Japanese announced their plan to build a so-called 'fifth generation' computer having intelligence approaching that of a human being. It can be seen from Table 1.1 that such a computer would be a huge step forward, having natural language capabilities, and processing and reasoning capabilities far greater than machines available either then or now. To meet these objectives, the Japanese outlined an ambitious programme involving four main areas of investigation: man–machine interfaces; software engineering; very large-scale integrated circuits; and knowledge-based systems.

Overnight, AI and expert systems became of national importance. The US and European governments reacted defensively to prevent the Japanese dominating the information industries of the 1990s and beyond. Research programmes aimed at stimulating collaboration between industry and academic organizations and forcing technological pace were set up with great haste. The next year, 1982, saw the start of the Alvey programme administered by the UK Department of Industry, the ESPRIT programme (European Strategic Program for Research in Information Technology) funded by the European Community, and a major programme of research coordinated by the

Generation	*Dates*	*Component*	*Description*
1st	1945 – mid 1950s	Vacuum tubes	Big, slow unreliable computers
2nd	mid 1950s – 1965	Transistors	More reliable but still slow; used machine-level instructions
3rd	1965 – early 1970s	Integrated circuits	Quicker, smaller, more reliable; used 'high-level' languages
4th	early 1970s – present	Large/very large scale integrated circuits (LSI/VLSI)	Systems used today, speed, reliability & 'high-level' language facility much improved.
5th	?	Even larger scale integrated circuits? New materials?	Systems having natural language intelligent interfaces, based on new parallel architectures (known as non-Van Newman architectures) new memory organizations, new programming languages and new

Table 1.1 *Computer generations*

Microelectronics and Computer Technology Corporation (MCC) in the United States.

In the rush to gain advantage, many corporations invested huge sums of money in research and development projects. Expert system tools quickly became available and the number of courses and journals in expert systems mushroomed. Unfortunately the rapid growth fuelled exaggerated claims about the capabilities of expert systems. Not surprisingly, many of the projects in the early 1980s fell short of expectations. This led to a period of disillusionment and a sharp fall in the popularity of expert systems in about 1986, particularly in the USA. Fortunately, today a more realistic view of expert systems prevails. Most people now accept that an expert system cannot completely replace a human expert and that it is not a panacea for an organization's loss of human expertise or lack of investment in training.

1.1 Characteristic features of expert systems

Expert systems are different from conventional programs in many respects, for example:

1. Expert systems contain practical knowledge (facts and heuristics) obtained from at least one human expert and should perform at an expert's level of competence within a specialized area. Conventional programs do not try to emulate human experts.

2. The knowledge is coded and kept separate from the rest of the program in a part called the *knowledge base*. This permits easy refinement of the knowledge without recompilation of the control part of the program, which is often known as the inference engine. This arrangement also enables expert systems to be more easily updated, and thus improved, at a later date. It also means that the control and interface mechanisms of some systems can be used with different knowledge bases. Systems of this type are called expert system shells. With conventional programs, knowledge about the problem and control information would be intermixed, making improvement and later development more complicated.

3. Knowledge is represented with the use of symbols using techniques known as production rules, frames, semantic nets, logic, and more. This natural form of representation means that the knowledge base is easy to examine and modify. Conventional programs can only manipulate numerical or alphabetical (string) data, not symbols.

4. Expert systems attempt to generate the 'best' possible answer by exploring many solution paths. They do this using heuristic searching techniques which will be discussed later. Conventional programs are executed according to a predefined algorithm and have only one solution path.
5. Expert systems are able to offer explanations or justifications on demand. Since expert systems are typically interactive, they are capable of explaining how or why information is needed and how particular conclusions are reached. This can be provided in the middle or at the end of consultations. Information of this type is provided to boost the user's confidence in the system and is not generally provided with conventional programs.
6. Expert systems can occasionally make mistakes. This is not surprising, because the systems have to rely on human expertise and are designed to behave like human experts. However, they do have an advantage over conventional systems in that program code can be more easily changed when mistakes occur, and some expert systems have the ability to 'learn' from their errors.
7. Expert systems are able to handle incomplete information. When an expert system fails to find a fact from the knowledge base that is needed to derive a conclusion, it first asks the user for the information. If the information cannot be supplied then the system will try another line of reasoning. Obviously if too much information is missing, the system will be unable to solve the problem. Conventional programs would crash immediately if the data needed were unavailable.
8. Some expert systems are also able to handle uncertain information. Expert systems offering this facility require certainty factors, confidence factors, or probabilities to be associated with information. These are used to indicate the extent to which the expert believes the information is true. They are used during the inferencing process to express a degree of confidence in the conclusion reached. This type of approach is rarely, if ever, used in conventional programming.

1.2 Why are expert systems important?

The motivation for building expert systems must lie in the benefits obtained. This is particularly true in the commercial and industrial sectors, where the return from an expert system development would be expected to far exceed the costs incurred. What then are the possible benefits? These would depend on individual situations but the more general advantages are listed below:

1. Experts can be freed from routine tasks and made available for more exciting, creative and demanding work.

2. Expertise can be pooled when more than one expert contributes to the system development. The pooling exercise can assist in the refinement of procedures and help to make them more consistent.

3. Knowledge can be safeguarded, developed and distributed. Enormous sums of money are spent on training individuals, yet all their knowledge and expertise is lost when they die or leave the company. Expert systems offer a way of capturing this expertise and knowledge whilst at the same time making it available to other people.

4. Expertise can be available 24 hours a day. Since expert systems provide explanations for advice given, they can be used without the presence of the expert.

5. Expert systems can be used for training purposes. The problem-solving and explanation capabilities of expert systems are particularly useful in training situations. Training can also be distributed throughout a company and done on an individual basis at times suited to the employee.

6. Expert systems can provide a standardized approach to problem solving.

7. The development of an expert system offers the expert an opportunity to critically assess and improve his problem-solving behaviour.

8. The performance of non-experts can be improved over a period of time and may eventually even reach expert status.

9. In many situations, expert systems can provide solutions to problems far more quickly than a human expert.

10. Expert systems have the potential for saving companies a vast amount of money, thus increasing profits.

1.3 Applications of expert systems

The huge potential of expert systems has not gone unrecognized in academia, industry or commerce. The early systems, developed in the research environment, prepared the ground well for much bigger and better systems. Today the technology can boast a wide range of application areas, several of which are discussed below.

1.3.1 EARLY SYSTEMS

The first expert system to be developed was DENDRAL at Stanford University in the late 1960s (Lindsay *et al.*, 1980). DENDRAL is capable of determining the chemical structure of unknown compounds by analysing mass spectrometry data. DENDRAL has been successfully used by many chemists and has even resulted in the discovery of new chemical structures.

Following the success of DENDRAL, the same research team produced MYCIN. MYCIN is an expert system designed to deal with problems in the diagnosis and treatment of infectious blood diseases (Shortliffe, 1976). Work on the system continued until the 1980s, when tests showed that its performance compared favourably with that of physicians (Lenat and Brown, 1984). Several projects related to MYCIN were also completed at Stanford; these included a knowledge acquisition component called THEIRESIUS (Davis, 1982), NEOMYCIN and GUIDON tutorial type versions of MYCIN (Clancey, 1981; Bramer, 1982), PUFF (an aid to diagnosing pulmonary disease) and EMYCIN, an expert system shell.

Other notable early systems in the medical field include PIP (Pauker *et al.*, 1976) used to record the medical history of patients with oedema; INTERNIST-1/CADUCEUS (Pople, 1982) that attempts to diagnose internal diseases; and CASNET (Weiss *et al.*, 1978) developed to assist in the diagnosis and treatment of glaucoma.

Early commercial expert system projects include PROSPECTOR, (Gaschnig, 1982) a system that assists geologists in the discovery of mineral deposits, and RI, now enhanced and called X/CON, that is used by the Digital Equipment Corporation to configure VAX computers to customer specifications.

1.3.2 SOCIAL AND PUBLIC SECTOR APPLICATIONS

In recent years a number of researchers have applied expert systems technology to social applications. The systems produced have attempted to 'improve both the quality and quantity of advice and expertise available to the man in the street' (Smith, 1988). Examples include systems designed to:

- provide advice for expectant mothers about maternity rights
- offer advice on an employee's rights regarding dismissal

- provide information and advice about the maze of local authority housing grants and planning procedures
- provide guidance on the legislation and practice relating to social security benefits
- assist with car maintenance
- provide assistance with travel planning

A few of these types of expert systems are already available to the general public via local authority viewdata systems. One can imagine that in the near future such programs might also be available in public libraries, citizens advice bureaux and the like.

1.3.3 FINANCIAL APPLICATIONS

Many financial organizations, such as banks, insurance companies and finance houses, are now using expert systems to try to give them a competitive edge. Expert systems have been used for a wide variety of applications in this field including systems designed to:

- assess customer credit risk
- assess insurance premiums and risks involved
- give advice on investment, stock exchange regulations, tax and mortgages
- assess business insolvency
- assess insurance claims

Big stakes are involved in such systems. For example, it is estimated by the developers of the UNDERWRITING ADVISOR that the use of this system could save insurance companies whose annual commercial premiums average $250 million, a total of $35 million over a 5-year period (Wolfgram, Dear and Galbraith, 1987, p26).

1.3.4 INDUSTRIAL APPLICATIONS

A survey in 1988 showed that most large industrial and manufacturing companies in the UK had either introduced expert systems into their daily operations or were experimenting with the technology (O'Neill and Morris, 1989). Applications in this area include:

- fault diagnosis (e.g. from computer circuits to whole plants)
- control (trouble-shooting, air traffic control, production control etc.)
- design (machines, plants, circuits, etc.)
- military operations
- quality assurance

- design/construction planning
- software design
- planning of complex administrative procedures

Numerous applications can also be found in education. For a fuller account of applications the reader is referred to the numerous texts available on the topic (e.g. Waterman, 1986; Lindsey, 1988; Wolfgram, Dear and Galbraith, 1987; and Feigenbaum, McCorduck and Nii, 1988).

2 Components of an expert system

Conceptually expert systems have four basic components (Figure 1.1): the knowledge base, the interface, the inference mechanism or inference engine, and the global database.

2.1 Knowledge base

A knowledge base is the part of the program that contains the knowledge associated with a specific domain. It includes facts about objects (physical or conceptual entities), together with information about the relationships between them and a set of rules for solving problems in a given domain. The latter is derived from the heuristics, which com-

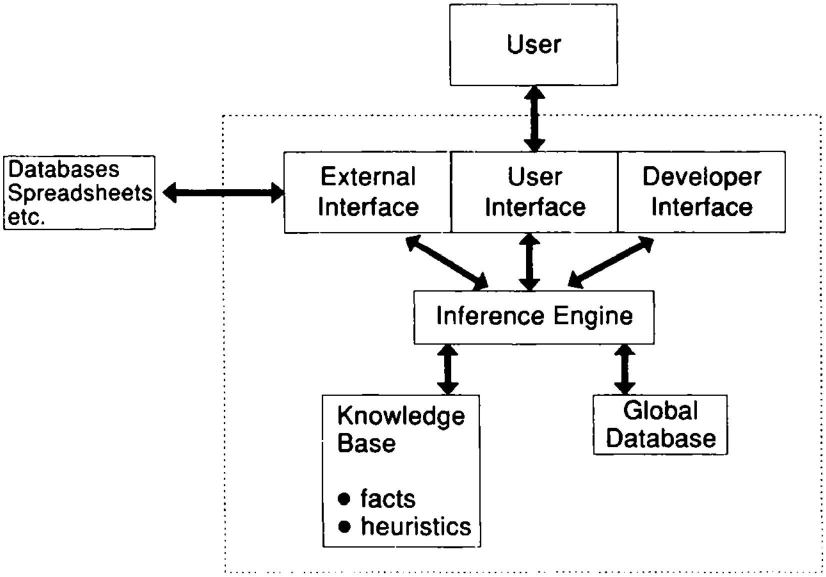

Figure 1.1 *Architecture of an idealized expert system*

prise judgements, intuition and experience, obtained from the expert(s). Sometimes the knowledge base also contains metarules, which are rules about rules, and other types of knowledge such as definitions, explanations, constraints and descriptions. Precisely what is incorporated, and the way the knowledge is represented will depend on the nature of the expert system. The techniques used are described in a later section.

2.2 Interface

The interface can be considered as having three main parts: the user interface, the developer's interface and an external interface.

The user interface is the section of the program which enables the user to communicate with the expert system. Most expert systems are interactive; they need users to input information about a particular situation before they can offer advice. The exceptions are where expert systems are used in closed-loop process control applications. In these cases the input and output of the expert system is via other machines. Such systems will not be considered in this book. Most of the existing user interfaces of expert systems are menu-driven, accepting single words or short phrases from the human user. A few have limited natural language capabilities, but much work still remains to be done in this area. A good user interface to an expert system will allow the user:

- to ask questions, such as why advice has been given, how a conclusion has been reached or why certain information is needed
- to volunteer information before being asked
- to change a previous answer
- to ask for context-sensitive help on demand
- to examine the state of reasoning at any time
- to save a session to disk for later perusal
- to resume a session previously abandoned mid-way

In addition to these characteristics, expert systems need to be easy to learn and use, and involve a minimal amount of typing by the user.

Most of today's integrated expert system tools will provide a developer's interface. This enables a knowledge engineer (the name for the developer), to build the knowledge base, test it and make modifications. Since this process is iterative and can involve many cycles, it is essential that the program offers user-friendly editing facilities and good diagnostic capabilities. Easy access to the user's interface for testing the system is also important. Morris (1987) discusses both the developer's and the user interface requirements in some detail.

The external interface is concerned with the exchange of data from sources other than the user, for example spreadsheet and database packages, data files, special programs, CD-ROM products or even online hosts. Early expert systems had very poor or non-existent external interfaces. The situation is now changing, however, as the demand for integrated computer systems becomes ever more important (O'Neill and Morris, 1989).

2.3 Inference mechanism

The inference mechanism is responsible for actually solving the problem posed by the user. It does this by using a set of algorithms or decision-making strategies to generate inferences from the facts and heuristics held in the knowledge base and/or information obtained from the user (Lachman, 1989). The algorithms also control the sequence in which inferences are generated, add newly inferred facts to the global database and, in some cases, process confidence levels when dealing with incomplete or uncertain data.

The main purpose of using algorithms is to find a solution to the problem posed as efficiently as possible. Problem-solving algorithms used in expert systems can be divided into three layers (Wolfgram, Dear and Galbraith, 1987):

1. General methods which are regarded as the building blocks of problem-solving techniques.
2. Control strategies which guide the direction and execution of the search.
3. Additional reasoning techniques which assist with modelling and searching for the solution path.

General search methods can be divided into two categories: blind searches and heuristic searches. Blind searches do not employ intelligent decision making in the search; the paths chosen are arbitrary. Examples of blind search techniques include *exhaustive*, where every possible path through a decision tree or network is analysed; *breadth-first*, where all the paths at the top of the hierarchy are examined before going on to the next level; and *depth-first*, where the search continues down through the levels along one path until either a solution has been found or it meets a dead end, in which case it has to backtrack to find the next possible path. Heuristic searches are more efficient than blind searches because they attempt to identify the pathways which will most likely lead to a solution. Examples of heuristic searches include hill-climbing, best-first, branch-and-bound, A^* algorithm and generate-and-test, details of which can be found in most books on AI (see, for example, Rich, 1983).

Control strategies, or reasoning strategies as they are sometimes called, are used to decide what operators to apply at each stage of the search. The most common control strategies used in expert systems are forward chaining, backward chaining and bi-directional. Forward chaining strategies start with the data and work forward to find a solution. In rule-based systems, the facts are matched with the antecedent, or the 'IF', part of the rules. If a match occurs the rule is fired and the consequent, or the 'THEN', part of the rules becomes the new fact. Chaining continues with user interaction, where necessary, until the solution is found. Backward chaining works in the opposite direction. In this case the process starts by identifying possible solutions. It then searches the knowledge base for relevant facts or requests information from users to either verify or disprove them in turn. In rule-based systems using backward chaining, facts are matched with the consequent part of the rules. Forward chaining and backward chaining strategies are also known as data-driven and goal-directed searching techniques, respectively, for obvious reasons. Bi-directional strategies use a combination of both forward and backward chaining to try to arrive at a solution more quickly.

Additional reasoning techniques are often incorporated into the inference engine to deal with uncertainty and anomalies between the facts and relationships in the knowledge base. The commonly used techniques are: Bayesian probabilities, the use of certainty factors, degrees of belief and measures based on fuzzy logic. All attempt to give the user some idea of the confidence he can place on the advice given. Another technique, which is becoming more popular, is blackboarding. This is often used when the knowledge required to solve a problem is segmented into several independent knowledge bases and/or databases. The blackboard acts as a global knowledge base, receiving and storing problem-solving knowledge from any of the independent sources. Further information on these and other control strategies can be found in Hayes-Roth (1984); Keller (1987); Graham (1989); and Harmon and Sawyer (1990).

2.4 Global database

The global database is the section of the program that keeps track of the problem by storing data such as the user's answers to questions, facts obtained from external sources, intermediate results of reasoning and any conclusions reached so far (Barrett and Beerel, 1988). It is really just a working store and is wiped clean after each session.

3 Knowledge acquisition and representation

One of the most difficult tasks facing expert system developers is 'knowledge acquisition' (Sowizral, 1985). Knowledge acquisition can be defined as the process which 'involves eliciting, analysing and interpreting the knowledge which a human expert uses when solving a particular problem, and then transforming this knowledge into a suitable machine representation' (Kidd, 1987). Knowledge acquisition can be extremely slow and costly as well as difficult, and justly earns the reputation of being the main bottleneck in the development of an expert system. To reduce the tedium and improve the effectiveness of knowledge acquisition, a variety of techniques has been developed. This section outlines these techniques and the common knowledge representation formalisms.

3.1 Knowledge acquisition techniques

Before using any of the techniques described below it is essential that knowledge engineers have thoroughly familiarized themselves with the problem or domain area. Grover (1983) suggests that knowledge engineers would be advised to produce something like a Domain Definition Handbook which might contain:

- a general description of the problem
- a bibliography of principal references
- a glossary of terminology
- identification of experts
- characterization of users
- definition of suitable measures of performance
- description of example reasoning scenarios

Armed with this background knowledge the developer can start the process of acquiring the expertise or private knowledge of the domain expert that stems from the accumulation of years, and sometimes decades, of practical experience. This includes:

- knowledge of concepts in the domain and the relationships between them
- the relative importance and validity of the concepts and relationships
- knowledge about routine procedures
- strategies for dealing with unexpected cases
- facts and heuristics (little known rules-of-thumb) used to make educated guesses when necessary and to deal with inconsistent or incomplete data
- classificatory knowledge which allows the expert to make fine distinctions among a number of similar items

Obviously this process can be omitted if the domain expert is also the knowledge engineer. There are pros and cons, however, to the expert being the knowledge engineer (O'Neill and Morris, 1989). The majority of books about knowledge acquisition warn against being 'one's own expert' because systems produced in this way can be provincial in effect and can contain idiosyncrasies.

The main techniques used in knowledge acquisition are: interviewing, protocol analysis, observation, and multidimensional techniques. These are discussed briefly below. For more detailed information, readers are referred to Hart (1986); Kidd (1988) Neale (1988); Diaper (1989); Neale and Morris (1989); and Boose and Gaines (1990).

3.1.1 INTERVIEWING

Interviewing is by far the most common method of knowledge acquisition (O'Neill and Morris, 1989). Interviews are particularly useful for acquiring basic knowledge about the problem domain such as concepts, general rules and control strategies. Apart from the first meeting with the expert, which is likely to be unstructured since the primary objective is to establish rapport, interviews should be focused with specific aims and objectives in mind. In focused interviews the knowledge engineer controls the direction of the interview by asking questions about selected topics. To help this process a number of questioning strategies have been developed. These include:

1. Distinction of goals. Experts are asked what evidence is necessary to distinguish between one goal (conclusion) and another.

2. Reclassification. Experts are asked to work backwards from goals and sub-goals by elaborating on the actions or decisions on which they are supported.

3. Dividing the domain. After dividing the domain into manageable chunks, the expert is given a set of facts (e.g. symptoms) and forward chains through successive sub-goals to reach the final goal (solution).

4. Systematic symptom-to-fault links. Here a list of all possible faults in a system and all possible symptoms are presented to the expert, who is asked which faults would produce which symptoms.

5. Critical incident. This involves the expert being asked to recall particularly memorable cases.

6. Forward scenario simulation. In this the expert describes in detail how he would solve hypothetical problems posed by the interviewer.

3.1.2 PROTOCOL ANALYSIS

Protocol analysis is a technique which attempts to record and analyse an expert's step-by step information processing and decision-making behaviour. It basically involves asking the expert to think aloud while solving a problem. All the verbalizations, which are tape recorded, are then transcribed into protocols and analysed for meaningful relationships. In some cases, where video tape has been used, a skilful knowledge engineer can also take into account body language and eye movement when analysing the importance of such relationships.

Protocol analysis has been successfully used in a number of domains but it does have a few shortcomings. Its major drawback is that it is extremely time consuming – this is particularly true of the transcription phase. Experts can also think faster than they talk, therefore any analysis will only be partial. For these reasons, protocol analysis is best followed up with other techniques such as interviewing.

3.1.3 OBSERVATION

Observation is similar to protocol analysis except that experts are not required to think aloud. Recordings consist of natural dialogue and, if video images have been taken, the expert in action. Some researchers have found it more effective than protocol analysis in the field of medical diagnosis (Cookson, Holman and Thompson, 1985; Fenn *et al.*, 1986), but it still has the same drawbacks: lengthy, time-consuming transcriptions containing repetitions, contradictions and inaudible mutterings. Observing an expert at work, however, can be a useful familiarization exercise at the beginning of a project. Rarely, if ever, can the technique be used alone.

3.1.4 MULTIDIMENSIONAL TECHNIQUES

The purpose of these techniques is to elicit structural criteria which are used by the expert to organize his concepts, and thus to form a representational 'map' of the domain, which is often difficult to put into words (Gammack, 1987). The most common technique used, particularly by academics, is card sorting. With card sorting, experts are asked to sort cards, each bearing the name of one concept, into groups according to any criteria they choose. This is repeated until the expert runs out of criteria. When analysed, the knowledge engineer should be able to formulate a conceptual map of the domain. This technique was successfully used by researchers when identifying how librarians chose between different sources of online information (Morris, Tseng and Newham, 1988). Two other techniques, multidimensional scaling and repertory grid, are similar in that they involve the experts comparing concepts to identify any differences between them. For discussion of these techniques, see, for example, Neale (1988).

3.2 Knowledge representation

Knowledge representation is concerned with how knowledge is organized and represented in the knowledge base. There are several methodologies available in AI but the five most common methods used in expert systems are as follows:

- production rules
- semantic networks
- frames
- predicate calculus
- hybrid of the above

By far the most popular method is production rules. This is particularly true in the case of microcomputer systems where, up until recently, lack of power has prevented the use of more complex and demanding representation techniques. The dependence on production rules is likely to change, however, as microcomputers become more powerful.

3.2.1 PRODUCTION RULES

Production rules are used to represent relationships in terms of English-like conditional statements. The basic conditional statement is of the form If-Then:

IF (condition)

THEN (action or conclusion)

which reads 'IF the condition is true THEN either the action should be taken or a conclusion has been reached'.

Production rules can be much more complicated, incorporating the operators 'and', 'or' and 'not' for example. To illustrate this, examine the rule below, which might feature in an expert system to advise library staff on whether to fine a member of staff for an overdue book.

IF user is staff

AND overdue letters>2

AND excuse is not plausible

AND staff member is not on library finance committee

AND staff member is not the librarian's spouse

THEN fine = days_overdue x 25p

AND advice is 'make them pay!'

The condition part of the rule, (before the THEN part), is also referred to as the antecedent, premise or left-hand side (LHS). Similarly the action part of the rule, (the THEN part) is also referred to as the consequent, conclusion, or right-hand side (RHS). Uncertainties can also be expressed in rules by attaching certainty factors to either the antecedent or the consequent part of the rule. Take for example the following simple rule:

IF distance in miles>2

AND weather is rainy CONFIDENCE 75

OR weather is windy CONFIDENCE 90

THEN transportation is car

In this case, if the user has to go more than two miles and he is at least 75% confident that it is raining, or at least 90% sure that it is windy, he is recommended to travel by car.

There are several advantages to rule-based systems:

1. Rules are easy to express and to understand.
2. The system is modular in design, in that rules can be added, deleted or changed without affecting the others.
3. Rules can represent procedural as well as descriptive knowledge.
4. Small rule-based systems are generally quick to develop.

The two main disadvantages of rule-based systems are:

1. They impose a very rigid structure, which makes it difficult to follow the flow of reasoning and to identify hierarchical levels within the problem area.
2. They are generally inefficient in execution because they are unable to make use of the more sophisticated reasoning strategies detailed in an earlier section.

3.2.2 SEMANTIC NETS

A semantic net, or semantic network, is a general structure used for representing descriptive knowledge. It is a graphical representation of the concepts and relationships existing in a particular domain. Concepts (or objects or events) are represented by nodes, and the relationships between them are represented by the links which span the nodes. The links are more commonly referred to as arcs, and have an arrow at one end to show the direction of the relationship (Figure 1.2).