Loughborough University

This item was submitted to Loughborough's Research Repository by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

The place of expert systems in business now and over the next decade

PLEASE CITE THE PUBLISHED VERSION

PUBLISHER

© Julie, Su-Jung Chen

PUBLISHER STATEMENT

This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Chen, Su-Jung. 2019. “The Place of Expert Systems in Business Now and over the Next Decade”. figshare. https://hdl.handle.net/2134/33054.

LOUGHBOROUGH
UNIVERSITY OF TECHNOLOGY
LIBRARY

AUTHOR/FILING TITLE

CHEN, J S J

ACCESSION/COPY NO.

03218202

VOL. NO. | CLASS MARK

-2 MAR 1992
date due:-
13 NOV 1992
LOAN 3 WKS. + 3
UNLESS RECALLED
SALFORD
date due:-
16 MAR 1993
LOAN 3 WKS. + 3
UNLESS RECALLED
EDINBURGH
UNIV.

LOAN COPY
date due:-
11 JAN 1994
LOAN 3 WKS. + 3
UNLESS RECALLED
date due:-
22 JUL 1994
LOAN 3 WKS. + 3
UNLESS RECALLED
LIMIST
MANCHESTER
LIBRARY USE
ONLY

30 JUN 1995
16 JAN 1995
16 NOV 1995
14 APR 2000
10 AUG 2000

003 2182 02

THIS BOOK WAS BOUND BY
BADMINTON PRESS
18 THE HALFCROFT
SYSTON
LEICESTER LE7 2UD

THE PLACE OF EXPERT SYSTEMS IN BUSINESS NOW AND OVER THE NEXT DECADE

by

Julie, Su-Jung Chen

A Master's Thesis

submitted in partial fulfillment of the requirements

for the award of

Master of Philosophy - M. Phil.

of the Loughborough University of Technology

May 1988

© Julie, Su-Jung Chen 1988

y9304405

Loughborough University of Technology Library
Date Jly 89
Class
Acc No 0321820 2

DEDICATION

To my mother Mrs. S. T. Chen and deceased father Mr. J. D. Chen

ACKNOWLEDGEMENTS

The work presented in this thesis could not have been accomplished without the assistance of many people. Therefore, the author wishes to express her gratitude to those people who have provided their assistance and help.

Firstly the author wants to thank her Research Supervisor and Director of Research, Dr. P. N. Finlay and Dr. M. R. Hill. Their understanding, patience and supervision helped the author sustain her research through the most difficult stages.

The colleagues of the author and the staff of the department are also to be thanked for their assistance. Mr. M. Forghani is the first one to be personally thanked. During the period of this research, he acted as a trainer, friend and even a brother of the author. Also, Mr. Peter Stratfold and Mrs. Joyce Savage: without their assistance, the author would not be able to finish her typing and printing of this thesis.

Ovum Ltd. is also to be thanked for their kindly lending the author their published book - Commercial Expert Systems in Europe. This book provided the author with the updated status of expert systems applications in business.

Those companies which allowed the author to pay them visits to see how they developed and used expert systems are also to be thanked for their cooperation and assistance.

Finally, the author would like to thank her strongest supporters --her family. This research would not have been accomplished without their assistance. This is especially true of her husband who is still engaged on his Ph.D research in this university.

The Place of Expert Systems

in Business Now and over the Next Decade

Table of Contents

	Page No.
List of Tables	
List of Figures	
Abstract	
1. Introducing Expert Systems	1
1.0 Introduction	1
1.1 Reasons for the Study	1
1.2 Types of I.T. Systems	8
1.2.1 Data, Information and Information in Business	8
1.2.2 Business Needs for I.T.	9
1.2.3 Data Processing Systems	12
1.2.4 Management Information Systems	14
1.2.5 Decision Support Systems	15
1.3 Expert Systems	17
1.3.1 What Are Expert Systems?	17
1.3.1.1 Characteristics of Expert Systems	19
1.3.1.2 Differences Between Expert Systems and Conventional Systems	22
1.3.2 Evolution of Expert Systems	25
1.3.3 Applications of Artificial Intelligence	28
1.4 Conclusion	32

2. General Issues on Expert Systems 33

2.1 How Do Expert Systems Work? 33

2.2 Classification of Expert Systems 36

2.3 Proposal of Creating Expert Systems Applications 39

2.3.1 What Is Knowledge? 39

2.3.2 Guidelines for Selecting an Expert Systems Application and for Identifying the Problem Domain 41

2.3.3 Differences between Acquiring Knowledge for Expert Systems and Conventional Systems 45

2.3.4 Knowledge Acquisition 46

2.3.5 Knowledge Representation 54

2.3.6 Reasoning 58

2.3.7 Explanation 65

2.3.8 Tools 66

2.3.9 Validation 74

3. A Review of Experimental Research Carried out by the Author 76

3.0 Introduction 76

3.1 The Visits 77

3.1.1 Reasons for the Visits 77

3.1.2 Hurdles Encountered When Choosing Companies to Visit 79

3.1.3 The Choice of Company 79

3.2 Looking at the Business Use 82
3.3 Analyses, Reviews and Results for the Visits 85
3.3.1 Overall Review on the Visits 85
3.3.2 Problems Discussion onthe Visits 91
4. Production of an Expert System for the Selection of Financial Planning Software 98
4.0 Introduction 98
4.1 The Literature 99
4.2 Approaches for Development 101
4.2.1 Why Chose a Shell to Carry out the Project? 101
4.2.2 A Survey of Expert Systems Shells 101
4.2.3 Why Xi Plus Was Chosen for Developing the Expert System? 122
4.3 Structuring the Problem 124
4.3.1 The Expert System Domain 124
4.3.2 The System Rules 133
4.4 Review of the Approaches Applied in the Project 133
4.5 Further Work 135
4.6 Conclusion 143
5. Features of Current Expert Systems and Future Trends of Expert Systems Development 145
5.1 Summary of the Features of Current Expert Systems 145
5.1.1 Features of Current Expert Systems 145
5.1.2 Limitations of Current Expert Systems 147
5.2 Current Status of Expert Systems in UK Business 148
5.2.1 The General Scene 148
5.2.2 Technical Overview of UK Business Application 151

5.3 Future Trends of Expert Systems Development 156

5.4 Research Directions of Expert Systems in the Next 5 - 10 Years 170

6. Conclusion and Summary 172

References 178

Appendices

A. Case Studies 190

B. A list of European Available Shells 232

C. Knowledge Base for Selecting a Financial Planning Package 235

D. A Form of Questionnaire for Financial Planning Packages 246

List of Tables

	Page No.
1.1 Numbers of micro computers installed in 8 countries and one geographical area (Western Europe)	3
1.2 Historical growth of computer systems installed in Taiwan	6
1.3 Percentage distribution of areas of computer application in Taiwan (June 1986)	7
1.4 Summarised differences between expert systems and conventional systems	23-24
1.5 Application areas of Artificial Intelligence	31
2.1 Comparative advantages/disadvantages of Lisp and Prolog	68
3.1 Analyses and results for the visits	85 86
5.1 List of commercial versions of Prolog in Europe	154-155
5.2 Overview of knowledge problems common to most companies or professionals	159-160

List of Figures

	Page No.
1.1 Process of decision-making through Data Processing	14
1.2 Interaction of expert systems components	19
1.3 Evolution of AI researches on expert systems	27
2.1 AND/OR tree for goals established in inference engine	34
2.2 The success factors in the creation of expert systems applications	44
2.3 Investigation of candidates - using repertory grid	52
2.4 Relationships of factors constitute a cold	59
2.5 AND/OR tree for hypothesis H	64
4.1 Name tree of Expert Edge using example of 'holiday'	112
4.2 Structure of the expert system for selecting a financial planning package	129
4.3 States of variables decide the Data Model Structure	131
4.4 States of variables decide the System Complexity	132
4.5 Example screen of SuperExpert for selecting a financial planning package	138
4.6 Rules induced by SuperExpert for selecting a financial planning package	139
5.1 The forecasted knowledge engineer market in the next five years	169

ABSTRACT

Information technology has entered a new generation. In recent years, considerable interest has been focussed on the commercialisation of expert systems, which represent an important application of Artificial Intelligence in the field of Information Technology.

Expert systems are now in a crucial stage of development because, although in business computerised systems are not new, expert systems still need time for their applicability and usefulness to be proved. The market for expert systems will not develop if such systems are unable to cope with the demanding applications of business; for example with top management problem-solving and decision-making. This thesis is principally concerned with determining the position of expert systems in business by looking at these major business related issues.

This thesis is aimed to examine the place/position of expert systems in business in order to give pointers as to how the development of expert systems should/would take place now and over the next decade. Two major aspects are discussed in detail:

1. What can expert systems do?
2. What are the trends in the development of expert systems over the next 5 - 10 years?

The first question has been discussed by a large number of authors, whilst there has been relatively little discussion of the second. This research focuses on the investigation of both the current and the future position of expert systems in business and therefore sets out to answer the above two questions.

For synthesizing the theoretical discussions from the literature and practical applications in business, four major methodologies are used for this research: (i) a literature review of the features and characteristics of expert systems, (ii) visits to the six companies that are developing and implementing expert systems, (iii) a survey of eight well- known expert systems shells and explanation of the production of an expert system through a shell, and (iv) an overview of the current status and forecasted future trends of expert systems in business.

In order to compile data on the usefulness and applicability of expert systems in business, six companies were visited during the period of the research. The features of these practical business applications of expert systems were compared with the theoretical approaches discussed in the literature.

The author was required to produce an expert system by using a shell. A survey of expert system shells has been carried out, and the results are reported.

With all the experience and knowledge from the above, an overview of the current status and forecasted future trends of expert systems development, is derived.

Based upon the above methodologies, the author analyses the factors affecting the future trends in expert systems development, such as the support of top management and human factors.

The forecast future for expert systems is different from the bright future that most of the literature anticipate. Future research directions of expert systems are also discussed in this thesis.

The major conclusions from the study can be drawn as follows:

a). expert systems will not be able to perform tasks as well as humans in the next 5 - 10 years. This is because of the human factors and the difficulties of eliciting, capturing and representing unstructured knowledge from experts.

b). the shells will become more and more popular in the development of expert systems because of their low costs.

c). the support of computing professional is necessary, otherwise the development and production of expert systems will remain small scale, with narrow problem domains, and of little use in business operations.

d). the encouragement and support of high-level management are vital to the development of expert systems.

CHAPTER 1 INTRODUCING EXPERT SYSTEMS

1.0 Introduction

This chapter starts by briefly introducing the reasons why the author has chosen to study for a Master of Philosophy degree and outlines the importance of the introduction of Information Technology into the business world. An introduction which has resulted in the evolution of expert systems.

In the latter part of this chapter, the typical types of IT Systems which include DP, MIS, DSS and Expert Systems, as a whole, are introduced. Furthermore, the characteristics of expert systems are described in detail and the differences between expert systems and conventional systems are summarised in tabular form in this chapter.

1.1 Reasons for the Study

Coming from Taiwan, the author wanted to review the current status of expert systems in order to be able to develop such systems there.

Taiwan, a small island situated in the Far East, is facing a transitional phase from being a developing country to becoming a developed country. The fact that labour costs in Taiwan are increasing, means that labour intensive industry is no longer sustainable in the face of the lower-costs in other developing

countries. Therefore, Taiwan is concentrating on developing its industry along technological lines; for example, in the development of precision industries, and in the production of IT components in the development of Information Technology itself.

The history of the development of Information Technology in Taiwan is much more recent than that of western countries, and yet the current status of IT and its application in Taiwan is considerable. Up to mid-1986, around three thousand computer systems have been installed in both industrial and educational organisations which represents a 29% increase over the figure of 2,298 systems recorded in June 1985 (1). The figure of IT usage in Taiwan may only represent a minor percentage if compared to a similar figure of usage for the United Kingdom. The author did try to ascertain a comparable official figure for general use of Information Technology in the UK but, unfortunately, no such figure could be ascertained. Therefore, in order to help the readers to make a comparison of general usage of IT in the U.K. and Taiwan, an illustration of the popularity of micro computers installed in eight countries and one geographical area (Western Europe) is given in table 1.1.

	1984			1985			Growth (%)	
COUNTRIES	AVG ANNUL EARNINGS (USD)	CUMU-LATED NO.(K)	POPU-LARITY (NO/K PERSON)	AVG ANNUL EARNINGS (USD)	CUMU-LATED NO.(K)	POPU-LARITY (NO/K PERSON)	NO. INS TALLED	AVG ANNUL EARNING
U.S.A	15,479	17,210	72.7	16,718	27,780	116.3	61	8.0
Canada	12,930	1,312	52.3	13,000	2,090	82.6	59	1.3
Japan	10,247	4,824	40.2	11,053	7,660	63.3	59	7.9
W.Europe	8,015	9,156	33.6	8,132	14,246	52.2	56	1.5
U. K.	7,544	3,048	54.0	7,808	4,348	76.9	43	3.5
W.Germany	9,989	894	14.6	10,228	1,574	25.7	76	3.0
France	8,920	882	16.1	9,035	1,462	26.6	66	1.3
Taiwan	3,046	136	7.2	3,142(X)	194	10.1	43	3.2
S. Korea	1,999	166	4.1	2,002	260	6.3	57	0.2

Table 1.1 Numbers of micro computers installed in eight countries and one geographical area (Western Europe)

Source from: Taiwan Institute for Information Industry

Note(X): The updated figure for 1986 is 4,900 (2)

From table 1.1, it can be seen that there are considerably fewer micro computers in Taiwan in comparison to the developed countries. One of the major reasons for this discrepancy seems to be familiarity with the English Language. Similar situation can be found from those countries where English is not used as the native or second language (Japan, W. Germany and France, for example) reveal a great reduction in the popularity of micro computers in comparison to those countries where English is spoken more readily, although these countries have a higher average annual earning than the U.K.. Moreover, the difficulty of putting Chinese characters into a computer is still the main obstacle to the development of computerisation in Taiwanese business.

The historical growth of computer system installations and the percentage distribution of computer applications in Taiwan are illustrated in table 1.2 and 1.3. From table 1.3, it can be seen that the areas of computer application in Taiwan are generally concerned with processing data regarding personnel, wages, inventories, accounts and bills. While wishing to advance in terms of technological innovation, Taiwan is confronted with the problem of being deficient in professional personnel for research and development(1). Proposed research directions for developing information technology in Taiwan are aimed at a broad coverage, for example, artificial intelligence (AI) and expert systems are two of the most important research areas. The academic and

research institutions in Taiwan have begun investigating AI and expert systems intensively. However, the professional researchers with a background in IT cannot meet the current pressing demand and the author believes that there will be more and more manpower devoted to the area of AI and expert systems development and applications in the next two years. Therefore, it is opportune to undertake a study at degree level on expert systems.

GROWTH OF SYSTEMS INSTALLATIONS (JUNE OF EACH YEAR)

YEAR	PRIVATE ENTERPRISES		INFORMATION COMPANIES		GOVERNMENT AGENCIES		STATE-RUN ENTERPRISES		ACADEMIC CIRCLES		TOTAL	
	NO.	GROWTH	NO.	GROWTH	NO.	GROWTH	NO.	GROWTH	NO.	GROWTH	NO.	GROWTH
1971	2	-	6	-	6	-	8	-	12	-	34	-
1972	2	0	7	17	8	33	8	0	17	42	42	24
1973	8	300	7	0	12	50	7	-12	20	18	54	29
1974	13	62	9	29	13	8	10	43	24	20	69	28
1975	18	38	17	89	16	23	17	70	32	33	100	45
1976	53	194	23	35	23	44	21	24	47	47	167	67
1977	85	60	26	67	26	13	25	19	55	17	217	30
1978	129	52	39	50	34	31	37	48	71	29	310	43
1979	208	61	60	54	47	38	56	51	92	30	463	49
1980	301	45	92	53	58	23	79	41	116	26	646	40
1981	440	47	99	8	94	62	183	132	172	48	988	53
1982	544	24	152	54	136	45	220	20	246	43	1298	31
1983	808	49	178	17	161	18	226	3	283	15	1656	28
1984	934	16	163	-8	246	53	307	36	361	28	2011	21
1985	1031	10	186	14	316	28	353	15	412	14	2298	14
1986	1385	34	229	23	405	28	458	24	497	21	2954	29

Source from: Directorate-General of Budget, Accounting and Statistics, Taiwan

Table 1.2 Historical growth of computer systems installed in Taiwan

APPLICATION AREAS	TOTAL	INSTITUTION: PRIVATE ENTERPRISES	INFORMATION COMPANIES(X)	GOVERNMENT AGENCIES	STATE-RUN ENTERPRISES	ACADEMIC CIRCLES
PERSONNEL/WAGES	14	15	11	18	10	16
BILLINGS	11	12	10	4	12	5
INVENTORY CNTL	12	14	12	4	13	5
TAX ADMINIST.	4	4	5	9	1	1
ACCOUNTING	12	14	14	8	9	6
PRODUCTION CNTL	6	8	7	2	4	1
FINANCIAL MANAG	9	9	9	6	11	6
STATISTICAL ANAL.	10	9	7	20	10	12
ENGINEERING APPL.	2	1	4	4	3	7
SCIENTIFIC APPL.	2	-	2	5	2	7
CUSTOMERS SVC.	7	6	10	8	14	1
TRANSPORTATION ADM.	2	2	2	-	4	-
TEACHING/TRAINING	3	1	5	1	1	21
OTHERS	6	5	2	11	6	12
TOTAL	100	100	100	100	100	100

Source from: Directorate-General of Budget, Accounting and Statistic, Taiwan

Table 1.3: Percentage distribution of areas of computer application in Taiwan

Note(X): The information companies are software houses and hardware firms.

1.2 Types of IT Systems

Before discussing the subject of Information Technology, it is necessary to discuss what is meant by 'information' and information in the 'business' context in order to gain a better insight into information technology itself.

1.2.1 Data, Information and Information in Business

'Data' are generally accepted as the subject which represents people, objects, events or concepts that can be given by conversation, mathematics or other symbolic surrogates. The term 'information' is the result of refining, formatting, filtering or converting data. Therefore, information is produced from data, i.e. data are the raw materials from which information is produced (3).

People use information for everyday living, for example, they use information to find out traffic conditions, the times of trains and TV Programmes. Likewise, organisations use information for their operations, for example, the investment climate, stock market status, etc.

In the practical world, the word 'information' always depends on the person receiving the information and the context in which he/she finds him/herself. Finlay and Forghani (4) give a definition of information as "data that are seen as directly

relevant to a person or an organisation". The main application of IT is to use computers to process data.

In business, information arises from the processes undertaken by departments or communications with outside bodies. Businesses must have the capability to manage their information which they need in order to operate effectively.

Criteria of useful information are given by Moss (5) as follow:

- sufficiently accurate,
- available in the right place,
- available at the right time,
- available in a form which can be read by those needing to make use of it,
- sufficient in quantity and of suitable quality.

Furthermore, Moss indicates that information "should not be duplicated unnecessarily nor kept beyond its period of usefulness and it should be carefully selected for its relevance to the objectives of the organisation" (5).

1.2.2 Business Needs for IT

Having stated the working definitions of both data and information, an introduction to the concept of information technology itself is necessary before describing business needs for IT. Richardson (6) defined IT as "the collection, storage, processing, dissemination and use of information". Also, "it is

not confined to hardware and software but acknowledges the importance of man and the goals he sets for his technology, the values employed in making these choices, the assessment criteria used to decide whether he is controlling the technology and is being enriched by it ".

Information technology is widely used in many areas just like the wide range of activities where information is used, such as office automation, telecommunication, education, etc.

Business needs for IT can be gauged by considering the match of IT capabilities and business activities. Burns (7) gives his views on the contributions of IT and these can be summarised as follows:

a) automation of clerical activities, such as payroll, invoicing and stock control within large business and government organisations.

b) centralisation of administrative functions, thereby shifting the decision-making activity up the organisational hierarchy.

c) introduction of an expensive, well-staffed technical unit so that the power and influence of the manager of that unit is substantially increased.

d) a shift in organisational function brought about by one part of the system being automated or supported.

Followings are the narrative explanations for Burn's views of IT contributions:

Automation of clerical activities

This was the initial requirement in the early age of computer-based systems. Many firms use computers to perform repetitive tasks so as to reduce costs.

Centralization of administrative functions

The benefits of automation when applied to the role of low/middle management were required by the senior management of companies. Better decision-making generally results from better access to information, the higher up the organisational structure the decisions are taken, the higher the value of resources involved.

Increased power and influence of manager

This is not only caused by the introduction of IT, it is also a consequence of the development of any other new technology. IT can provide the necessary information for low/middle-level managers' performing effective, multifunctional decision-making tasks. In the real-world, managers are not usually involved in developing IT systems themselves because of the time availability and their programming capability. Nevertheless, more and more managers do realise that some IT systems are virtually impossible

for other people to develop directly as they are unable to specify the information needed to support the semi-structured or unstructured decision-making tasks. Also, managers sometimes use IT as a means of increasing influence over their subordinates as well as making more effective decisions through those systems developed by someone else(8).

Shift in organisational function

This consequence is less obvious than those above. However, side-effects sometimes occur when information is provided for some particular purposes. For example: the combination of departments for more functional tasks as a result of work simplification - the procurement department may be combined into the finance department after the automation of internal control procedures.

During the decades of IT development, there have been a number of IT support systems developed by researchers. Four types of system which can be deemed as representative of the stages in the development of IT are discussed:

-Data processing systems (DP)

-Management information systems (MIS)

-Decision support systems (DSS)

-Expert systems

1.2.3 Data Processing Systems

Stimuli which are relevant and to be input into computers are called 'data'. Many events take place in a single working day in business organisations. When the facts about those events are worth recording, they become 'data'. Thus, data are the raw facts concerning occurrences or happenings in a business. A manager cannot spend most of his time wading through voluminous data to reach items which are important to his action taking or decision making. In order to overcome this problem, a system which is able to transform raw data into meaningful information to meet the needs of management is required, which is called a 'data processing system'. Martin (9) defined data processing as 'the conversion of raw facts into useful information'. Data processing is usually conducted by a computer system and so the above definition should be properly expanded into the conversion of raw facts into useful information 'under the control of a program stored inside the computer'(10). Figure 1.1 illustrates the method of data processing. The decision maker makes decision according to the meaningful information processed by DP systems. A good decision is more likely to be made because of the increase of accuracy and speed, this leads to the higher possibility of goal achievement.

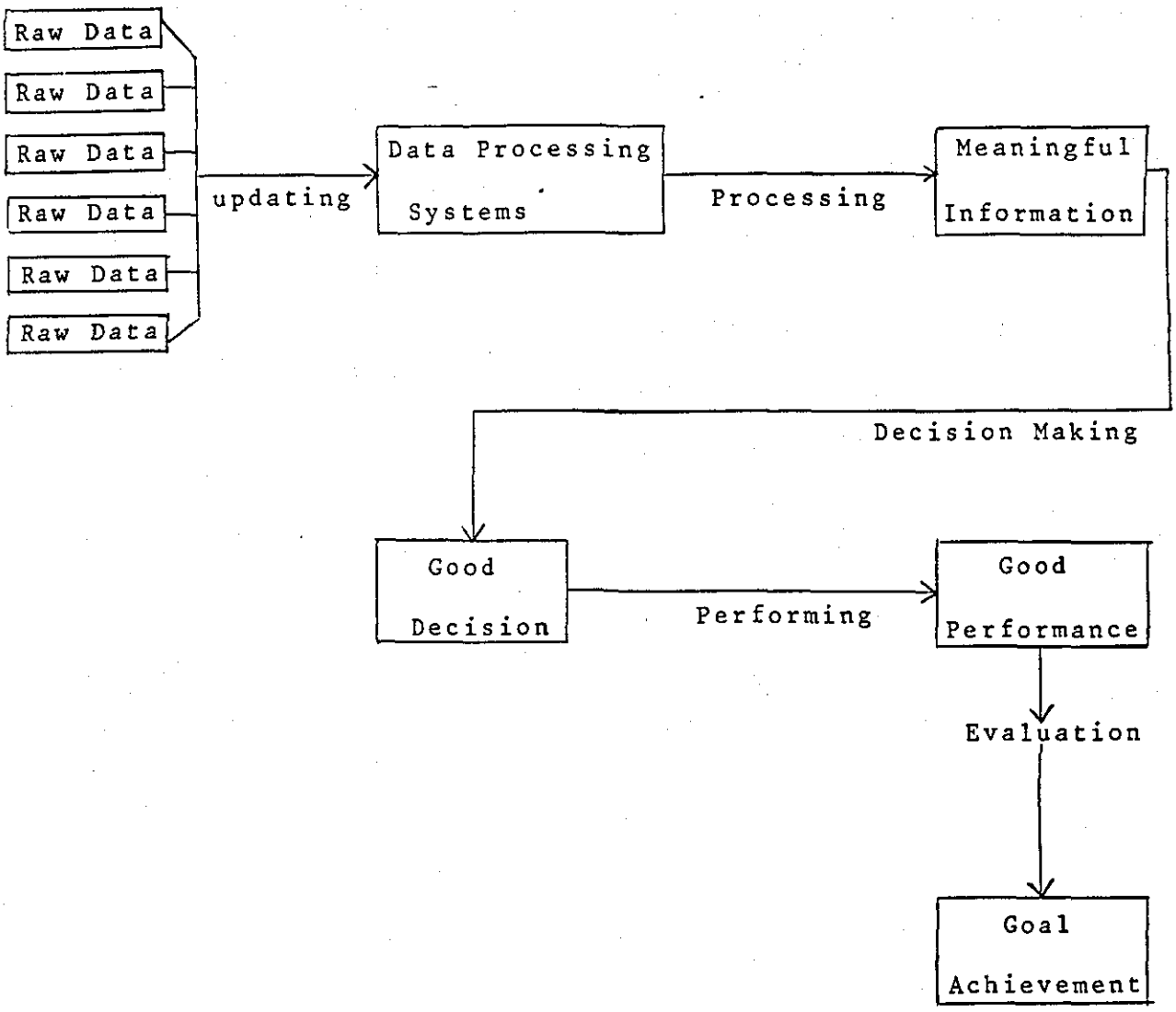


Figure 1.1 Process of decision-making through data processing

1.2.4 Management Information Systems

Data processing systems were developed primarily for record keeping and the automation of routine clerical tasks, such as payroll and billing which are processed by the input of already known data and the output of predictable results. At the data processing stage, accuracy and speed of transactions are the requisite criteria.

Management Information Systems (MIS) were developed in order to 'provide the information necessary to support the decision making process within the organisation' (11). Information is the source on which the manager needs to make a decision. Without information, the manager is unable to perform his function in the organisation.

Since MIS has been an outgrowth of DP, it is not easy to differentiate between these two systems. Kroeber and Watson (12) attribute DP and MIS to 'transaction processing', a function that is generally recognised as necessary to both DP and MIS. Possibly, the major difference between DP and MIS is the outputs, DP produces detailed reports and transaction data, whilst MIS produces summaries and report extracts which can be useful to a manager's routine decision-making and also produce replies to management queries. Detailed comparison of these two systems is given in table 1.4.

1.2.5 Decision Support Systems

There are many definitions of decision support systems, and there has not yet been a universally accepted one. Freyenfeld (13), realising the terminological confusion in this field and tried to produce a definition of DSS by offering a version 'generally accepted as valid and useful by representatives of some 30 suppliers, users, and academic organisations in the U.K.':

A decision support system is an interactive data processing and display system which is used to assist in a concurrent decision-making process, and which also conforms to the following characteristics:

(i) it is sufficiently user-friendly to be used by the decision maker(s) in person.

(ii) it displays its information in a format and terminology which is familiar to its user(s).

(iii) it is selective in its provision of information and avoids exposing its user(s) to an information overload.

From the above, it can be seen that emphasis is placed upon the DSS as being used by the decision maker(s) directly (i.e. in person). This highlights the problem that there are many decision-makers, especially at the top management level, who do not use computers directly for decision making themselves but treat the computer as a form of assistance of secondary importance and use them through their subordinates.

The evolution and constitution of DSS is well documented elsewhere. Finlay and Forghani (4) have detailed these aspects in their paper, for example. Leaving aside the question of the evolution and constitution of DSS, it is useful to focus attention upon the flexibility of the DSS to deal with strategies, ad hoc situations presenting a structureless approach and long-term decision methodologies which differentiate the DSS from DP and MIS which process data derived from structured

situations only, for example, sales of the previous month, actual overheads and expenditure of the previous month.

1.3 Expert Systems

1.3.1 What Are Expert Systems?

Like decision support systems, there are a number of definitions for expert systems, most of them place emphasis upon the academic viewpoint and, as such, they constitute intelligent advice, but the real modelling of the human brain's activities is not actually detailed explicitly. Such a definition, not infrequently, causes ambiguity for the inexperienced reader. A lengthy definition from the BCS (British Computer Society) is detailed below for clarification:

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.

Another definition of expert systems defined by Michie (16) is: A machine system which embodies useful human knowledge in machine memory in such a way that it can give intelligent advice and also can offer explanations and justification of its decisions on demand.

Among the plethora of definitions, Zorkoczy (17) gives a concise definition of expert systems :

Expert systems are software package (computer programs) aimed at providing expert 'consultancy' advice and assistance with problem-solving in limited specialist fields of science, engineering, mathematics, medicine, education, etc.

To summarise, expert systems are a set of computer programs which is capable of knowledge representation and reasoning for the purpose of providing expert advice and of problem-solving in specific areas.

Also, an expert system consists of a number of essential components: a knowledge base, a driver program, a natural language front-end translator program, an explanation capability, and a program to enable an expert to update the knowledge base

(18) . Figure 1.2 shows the interaction of expert systems components.

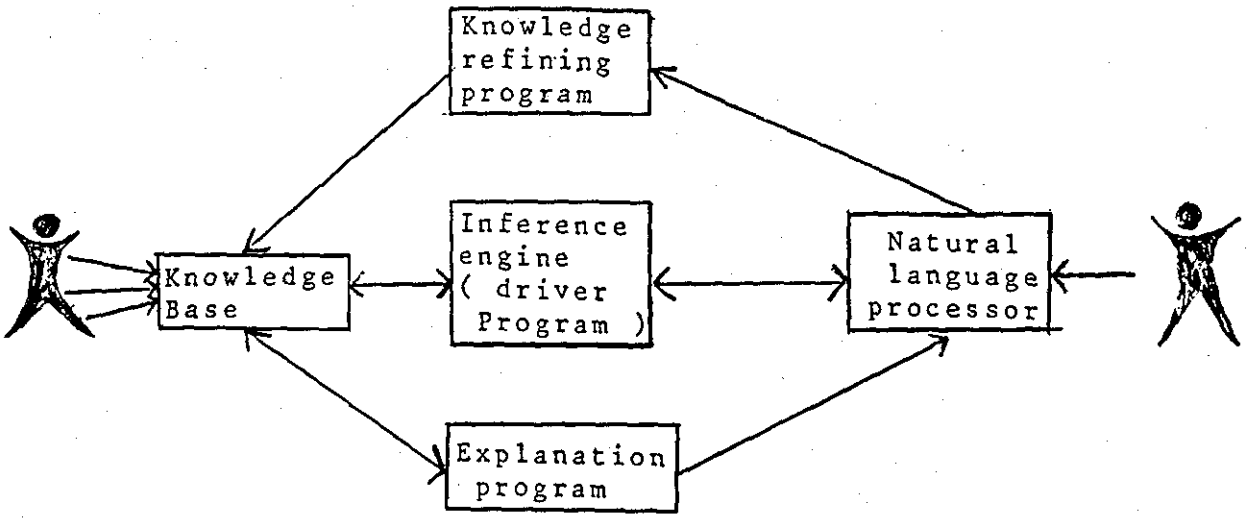


Figure 1.2 Interaction of expert systems components (18)

It can be seen from figure 1.2 that an expert system is actually a set of computer programs which incorporates an expert's knowledge into its knowledge base through knowledge engineering and performs knowledge inference through its inference engine so as to provide satisfactory answers to users' queries. It also provides explanations of the reasoning process in a language easily understood by the user to answer users' questions.

1.3.1.1 Characteristics of Expert Systems

Although several definitions of expert systems were discussed in the previous section, it is still necessary to detail the characteristics which a practical expert system possesses. This

section discusses the following characteristics of expert systems based on figure 1.3:

- knowledge base
- inference engine
- knowledge refining program
- explanation program
- natural language processor

Knowledge Base

The major part of an expert system is its knowledge base rather than its inference engine. During the system implementation, knowledge is accumulated. The knowledge representation is used to describe clearly and organise the knowledge in order to simplify the decision-making process. In the seven methods of knowledge representation given by Winfield (18), the production system is the most common method. A production system consists of a number of rules, each rule is set by IF....THEN....type. Thus, the storage and the codification of knowledge are the most important approaches in the implementation of expert systems.

Inference Engine

The inference engine is the program driving the system by using the given variables that have values and the rules to generate conclusion which matches the relation between the rules and given

variables. The way by which the inference engine reasons its rules will be introduced in Chapter 2.

Knowledge Refining Program

The most useful characteristics of expert systems are that they contain human expert's expertise and techniques which provide directions for problem-solving and represent the best point of view of the domain expert. In order to keep these knowledge, the knowledge refining program is necessary, i.e. by deleting, amending or inserting the old, existing or new information through the program.

Explanation Program

An expert system which expects its users to accept all the conclusions without having the opportunity to obtain an explanation of how those conclusions were reached should not be considered as a good system (unless this is developed under a specific purpose). Therefore, an expert system is expected to be capable of answering its user' request of 'how', 'why' or 'what if'.

Natural Language Processor

An expert system is able to provide this facility because it contains the necessary knowledge and facility to explain its reasoning through the communication in natural language and in words which are understandable to the user. Incorporated into the software of an expert system is the user interface which is

designed to facilitate communication between the user and the system.

1.3.1.2 Differences between Expert Systems and Conventional Systems

There are many differences between expert systems and conventional systems, the characteristics of expert systems constitute part of these differences. Alty and Coombs (19) and Quinlan (20) have documented such differences and these are summarised in table 1.4. Here the conventional programs are confined to traditional data processing systems and management information systems which are programmed in traditional computer languages, i.e. BASIC, FORTRAN, COBOL, etc.

	Conventional Systems (DP and MIS)	Expert Systems
Knowledge representation	By the appropriately defined representation, usually stored in fixed length and binary code.	Using natural form, without machine limitation, for example, manages (Peter, John) means Peter manages John.
Classifications & relations	Using 'dummy' variables for classification and symbols for relationships, for example $ in BASIC at the end of the variable should be treated as a string of characters not a number.	Using predicate calculus notation, for example, reports-to (John, Peter) IF manages (Peter, Jack) OR (manages (Peter, Jack) AND reports-to (John, Jack)) means "John reports to Peter is TRUE if either Peter manages Jack is TRUE or Peter manages Jack and John reports to Jack are both TRUE.
Control structures and Procedures	Have a large sequential element, punctuated by iterative procedures, such as Do..While For...Do, and GOTO.	Use rule of thumb in IF....THEN.... type.

Inference of Data	Using existing data to infer new data, this causes an increase on response time.	Using an inference engine which is separated from its knowledge base thus can perform the inference process quickly.
Inexact Reasoning	Overwhelmingly deals only with 'true' or 'false'. by well predefined variables which are provided under full certainty.	Can deal with uncertainty by using Bayesian probability, certainty factors or fuzzy logic.
Explanation for Reasoning	No facility	Facility
Modification	Difficult	Has separate knowledge base which can be amended easily.

Table 1.4 Summarised differences between expert systems and conventional systems (19, 20)

1.3.2 Evolution of Expert Systems

Expert systems constitute a major application of artificial intelligence (AI). AI brings together all kinds of professionals such as engineers, psychologists and linguists working with computer scientists in various areas of potential application.

In 1957, researchers developed the general-purpose program for solving general problems in certain areas (i.e. theorems and puzzles) (21) - this was the GPS (General Problem Solver) created by Newell, Shaw and Simon (22). However, it was eventually found that the development extremely difficult and inefficient. At best, researchers could only develop one specific program to deal with a particular sort of problem.

In the late 1960s, AI researchers concentrated on making computer programs intelligent, i.e. to describe problems in a usable form which can facilitate the problem-solving process. The first success with real possibilities was the DENDRAL system which used computer language for identifying molecular structures in unknown organic compounds and for planning a sequence of reactions to synthesise organic chemical compounds (21).

In 1970s, AI researchers realised that 'there were advantages attached to the strategy of representing human knowledge

explicitly in pattern-directed modules instead of encoding it into an algorithm that could be implemented using more conventional programming techniques (23). This important realisation resulted in the separation of the software concerned with the expert's knowledge – the knowledge base – from the section concerned with the problem-solving (inference). MYCIN, written in Lisp , is a good example of rule-base medical diagnostic expert system which provides consultative advice on diagnosis and treatment for infectious diseases (24). In MYCIN, medical knowledge is stored as a set of rules augmented by certainty factors. The factors are used to express the degree of belief in the conclusion of a rule. The development of intelligent programs by relating high technical and specific knowledge to a particular problem domain for a particular purpose represents the initial stage in the evolution of expert systems. Figure 1.2 describes the evolution of expert systems (25).

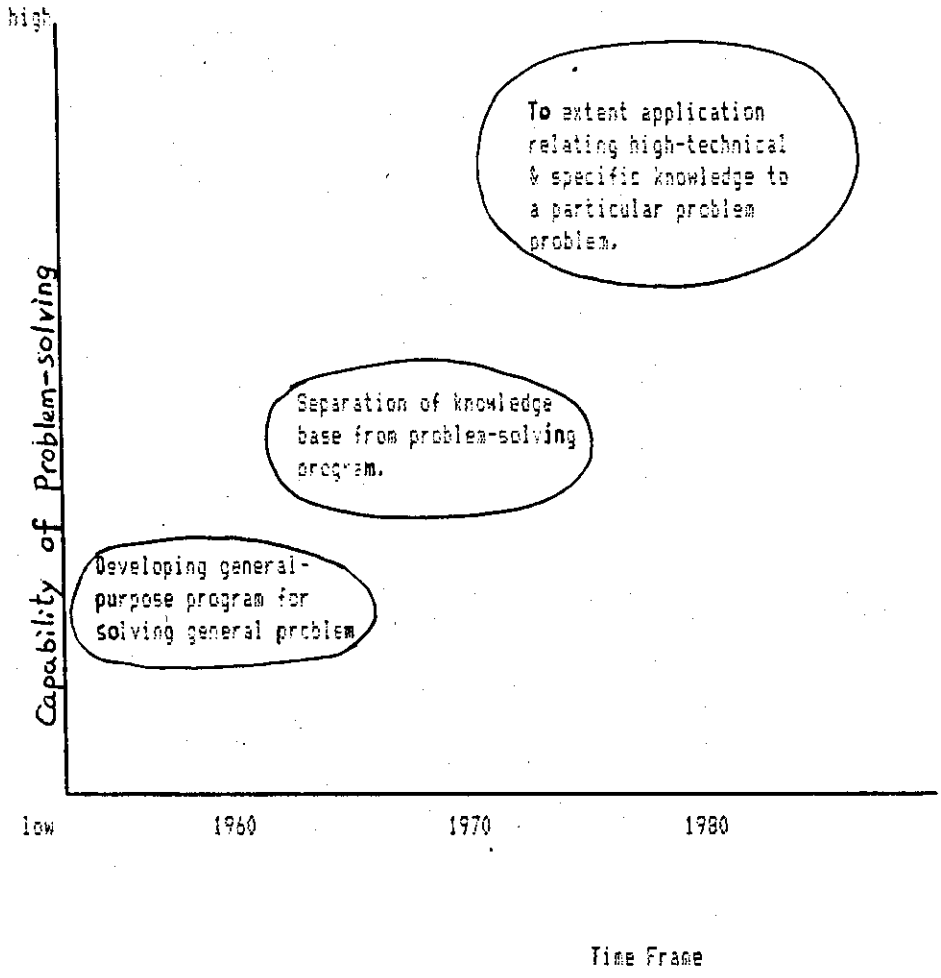


Figure 1.3 Evolution of AI research on expert systems

1.3.3 Applications of Artificial Intelligence

The research and application areas of Artificial Intelligence have been expanded during recent years. There are different application areas of AI defined by different researchers, the followings are representative examples:

a. Feigenbaum (26)

- Problem Solving
- Logical Reasoning
- Language
- Programming
- Learning
- Expertise
- Robotics and Vision
- Systems and Languages

b. Graham (27)

- Problem Solving
- Natural Language Processing
- Perception and Pattern Recognition
- Information Storage and Retrieval
- Control of Robotics
- Game Playing
- Automatic Programming
- Computational Logic
- Expert Systems

c. Nilsson (28)

- Natural Language Processing
- Intelligent Retrieval from Database
- Expert Consulting Systems
- Theorem Proving
- Robotics
- Automatic Programming
- Combinatorial and Scheduling Problems
- Perception Problems

d. Gevarter (29)

. Natural Language Processing (NLP)

. Computer Vision

. Expert Systems

. Problem Solving and Planning

Apart from the above, Harmon and King (30) consider artificial intelligence in three research areas:

. Natural Language Processing

. Robotics

. Expert Systems

To summarise the above five definitions into table 1.5, it can be seen that both the natural language processing and expert systems are the two major areas of application of AI agreed by all these researchers:

Application	Feigenbaum	Graham	Nilsson	Gevarter	Harmon & King
Natural Language	x	x	x	x	x
Expert Systems	x	x	x	x	x
Programming	x	x	x		
Robotics	x	x	x		x
Problem Solving	x	x		x	
Perception			x		
Information Storage/ Access		x	x		
Game Playing		x			
Computational Logic/ Vision		x		x	
Reasoning/Learning	x				
Theorem Proving			x		
Sorting and Scheduli			x		

Table 1.5 Application areas of Artificial Intelligence

1.4 Conclusion

From the four types of IT system (DP, MIS, DSS and Expert Systems) discussed, it can be said that research and development into expert systems are the result of a continual desire for the computerisation of human expertise in problem-solving. A recognition of the characteristic differences between conventional programs and expert systems has led to the latter becoming beneficial to business, mainly because of the following factors:

i) The separation of knowledge (Knowledge base) from the control structure (Inference).

ii) The user-friendly programming.

iii) The facility to cope with the condition of uncertainty.

iv) Efficiency in performing inference/search processes.

CHAPTER 2 - GENERAL ISSUES ON EXPERT SYSTEMS

As stated in Chapter 1, expert systems represent an application of Artificial Intelligence Technology. This chapter is concerned with the discussion of the way in which expert systems work and the general requirements for creating an expert system application.

2.1 How Do Expert Systems Work?

The word 'knowledge' in expert systems represents the combination of facts and rules, for example:

Fact: Mary is injured by a car accident.

Rule: If Mary is injured by a car accident, then hospitalisation is necessary.

In expert systems, a large number of rules are heuristic - i.e. different from precise mathematical analysis, they induce acceptable solutions, not exact answers. The knowledge represented in rules and facts that are needed by expert systems to make induction is called the knowledge base.

The program which provides expert systems with effective thinking power is called the inference engine. In the previous example, the result 'hospitalisation is necessary' is induced by using the fact 'Mary is injured by a car accident'.

In practical applications, expert systems are unlikely to be able to solve problems just by one step (i.e. only by one rule) and therefore many small parts of problems are established with their own sub-goals. Sub-goals are established by writing appropriate rules about them into the knowledge base. Using these rules in conjunction with the facts already known about the problem, the inference engine will proceed to find the appropriate rules for reaching the goal. This process is repeated until a solution goal is found.

The way the inference engine proceeds through sub-goals is often from the AND/OR tree. Figure 2.1 shows the AND/OR tree, where the initial facts(I) are shown at the bottom whilst the final goal(G) is shown at the top. To reach the final goal, the inference engine has to work through a sequence of inductions.

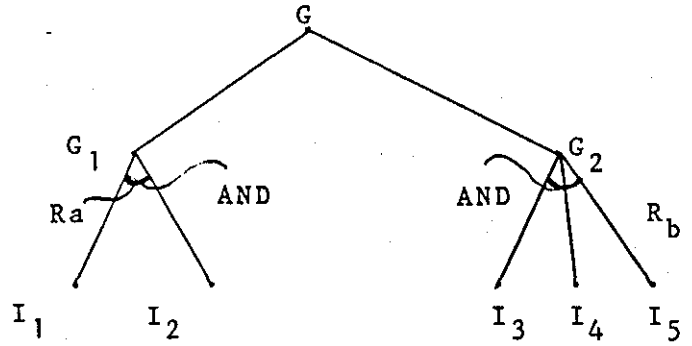


Figure 2.1 AND/OR tree for goals established in inference engine

In this figure, the initial facts (I) can be considered as the 'IF' part of a rule (R). For example, facts I_1 and I_2 are the conditions of rule R_a. In a forward chaining direction, the sub-goal(G_1) is reached when the 2 facts of I_1 and I_2 are satisfied. Similarly, sub-goal(G_2) is reached when the facts I_3, I_4 and I_5 are satisfied through Rule R_b. The inference engine will then reach the final goal G when the two sub-goals G_1 and G_2 are satisfied. The working condrol of an inference engine may include backward chaining, forward chaining and/or a bi-directional control strategy. These control strategies are introduced in next section.

For the purpose of making expert systems acceptable to the user, friendly communication between the two must be exist. This means that the communication must be in natural language free from inappropriate jargon and the text must be understandable to the user. Moreover, the expert system should have the capability to adapt the type of questions asked and the amount and type of information requested, to accommodate the user's needs.

Whether the user is naive or expert, he/she would like to know how the system reaches a conclusion. A user would not be satisfied by a conclusion reached by an expert system without any explanation. Expert systems have the means to explain the way a conclusion is reached. As stated in Chapter 1, although this capability is not necessarily required, an expert system should

be able to answer its users' request for an explanation by 'how', 'why' or 'what if'.

2.2 Classification of Expert systems

There are several methods of categorising expert systems. Addis (31) (highlighted by Simons (32)) tries to show the relationship and differences between expert systems and traditional data processing systems, and suggests that there are three major categories of expert systems. The first class is for those systems that can only provide answers that have been clearly recorded beforehand. The second class of expert systems allows simple fact interpolation where rules are used to generate new facts that can be used. The third class of expert systems is able to extrapolate facts where new rules of discovery can be generated. In fact, it is questionable whether there are any expert systems that can cope with the third category.

Another categorisation of expert systems is classified by Stefik et al.(33) (outlined by Alty and Coombs(19)) according to how well they can cope with problems that are not 'well structured'. Here the 'well-structured' problem can be regarded as a problem of small search space with reliable domain knowledge and data provided. This approach to the classification of expert systems influenced Alty and Coombs who use it to organise their ordering of chapters dealing with examples of expert systems in their book - Expert Systems: Concepts and Examples. This ordering was:

- rule-based diagnostic systems for reasoning from uncertain data and knowledge (MYCIN AND PROSPECTOR),
- associative and causal approaches to diagnosis (INTERNIST and CASNET),
- reducing large search spaces through factoring (DENDRAL),
- handling large search spaces through the use of abstraction (RI and MOLGEN)

Sell (22) provides two ways of classifying expert systems. The first is by area of application:

(1). Medicine (MYCIN, PUFF, etc.), here Sell indicates two reasons why the area is rich in products - the acute need of expertise for tackling complex systems since the human body consists of a complexity underlying systems. The second reason is that a detailed heuristic knowledge about how the system works is required.

(2). Chemistry and geology (DENDRAL, PROSPECTOR, etc.)

(3). Computer engineering (R1)

(4). Electronics (EL)

(5). Structural engineering (SACON)

The second way Sell indicates to classify expert systems is by the task that expert systems are called upon to perform:

(1). Analysing data and interpreting meaning (DENDRAL)

(2). Diagnosing the reasons for or sources of disparity between expected and actual states or operations of a system (MYCIN)

(3). Prognosticating the next state or action of a system, specifically give warning of impending malfunction (VM)

(4). Teaching and Training new experts (GUIDON)

Apart from the above ways of classifying expert systems, a practical way of categorising systems by means of the reasoning control strategy performed by the inference engine is introduced by the Butler Cox Report (34):

(1). Goal Driven Control Strategy (backward chaining):

This strategy is initiated by a goal rule and the system attempts to determine if the goal rule is correct. It goes back to the 'IF' sections of the rule and tries to determine if they are correct. The system proceeds to consider other rules that would satisfy the conditions and meet the goal. MYCIN, and most existing expert systems, use a backward chaining strategy(30).

(2). Data Driven Control Strategy (forward chaining):

The strategy begins with a set of conditions which satisfy the 'IF' clauses, then the system checks to determine what additional rules might be true and asks the user for input of data until the system reaches a goal.

(3). Bi-directional Control Strategy:

The strategy is a combination of the previous two strategies. It applies these two strategies simultaneously.

Since the thesis is aimed at the discussion of expert systems in business for practical purposes, the Butler Cox's method of classifying expert systems is thereby to be adopted for a survey of commercial expert systems shells in later discussion and Sell's second method of classifying expert systems is to be used for the business application case studies.

2.3 Proposal for Creating Expert Systems Applications

2.3.1 What is Knowledge?

Prior to discussing the proposal, this section discusses the nature of knowledge. Dretske (35) gives a definition that knowledge is "a form of justified true belief", the word 'justified' is noteworthy. However, in the practical world, it is not possible to give justification to all knowledge, especially to knowledge based on commonsense. Therefore, Dretske proposes to replace knowledge with information and belief. He identifies knowledge as 'information-produced belief', the information given to a person is perceived on the basis that he or she knows about the possibilities of that information from the source. Sell (22) indicated three sources of knowledge - literature, experts and examples, and three different bases of knowledge - scientific laws, experience and models. The purpose for using knowledge is to provide any information that helps to solve problems in the domain. The most helpful information is any clearly expressed regularity of information, that allows people to forecast what will happen or to explain how and why something has happened (22). Of those three different bases of

knowledge, Sell believes that the scientific law is the strongest form of regularity. Accordingly, acquiring knowledge in a scientific subject is much easier than in any other field. General problems of expert knowledge acquisition will be discussed in section 2.3.4.

One of the valuable contributions made by the development of expert systems is to business and industry. However, to most of the expert systems' users, the question " Can expert systems solve my problem?" is the most common query before they decide to use expert systems.

The major consideration, therefore, before selecting an expert system is to determine whether the development of the proposed application is possible and appropriate.

2.3.2 Guidelines for Selecting an Expert System Application and for Identifying the Problem Domain

Prerau (36) uses a 'checklist' of attributes, which describe the desirable features of setting an expert system domain for the identification of applications. These attributes cover both technical factors - such as the characteristics of the problem - and non-technical, organisational factors - such as the personality of human experts.

Attributes of desirable domain (Prerau,(36)):

1. Basic Requirements
   a. the completed system is expected to have a significant payoff for the corporation
2. Type of Problem
   a. the task primarily requires symbolic reasoning
   b. the task does not require knowledge from a large number of areas
3. The Expert
   a. the expert has built up expertise over a long period of task performance
4. Problem Bounds
   a. the task is neither too easy (taking a human expert less than a few minutes) nor too difficult (requiring more than a few hours for an expert)
   b. the number of important concepts (e.g. rules) required is limited to several hundreds

5. Domain Area Personnel
   a. the system can be introduced with minimal disturbance of the current practice
   b. the project is strongly supported by a senior manager, for protection and follow-up.
6. Other Desirable Features
   a. the task is decomposable, allowing relatively rapid prototyping for a closed small subset of the complete task, and then slow expansion to the complete task
   b. the user interface will not require extensive effort
   c. the task is similar to that of an existing expert system
   d. any requirement for real-time response will not involve extensive effort.

More concise guidelines for identifying a problem domain for a specific task are given by Harmon and King (30). The task:

. focuses on a narrow specialty.
. does not depend heavily on background knowledge or common sense.
. is neither too easy nor too difficult for a human expert.
. is defined as clearly as possible.
. has outcomes that can be evaluated.

The first of Harmon and King's guidelines of choosing a domain focussing on a narrow speciality is not easy for the purpose of accommodating the domain expert and the knowledge engineer. There should also be included a few exceptional situations which are uncommon under normal conditions and which rarely happen but

must nevertheless be considered. Compromises between the domain expert and the knowledge engineer are necessary. The second guideline emphasises the purpose of an 'expert' system which should represent the human expert's expertise rather than the commonsense that is general to most people. The last guideline is utilitarian, to provide a practical means to evaluate the costs and benefits of using expert systems.

Figure 2.2 summarises the success factors in the creation of expert systems applications with explanation for these factors. The most important requirement is the existence of a real expert who possesses the high-level expertise to tackle the domain problem from a more technical viewpoint than other people. Also, considering the problem of a mathematical model in real world, people would like to solve such problem by a traditional computing system rather than an expert system, because mathematical problem is difficult to be expressed into rules like IF....THEN.... type, and the answers which the mathematical problems required should be the solution sticking to the requirement of precision and accuracy, whilst precision is not strongly required by an expert system(4).

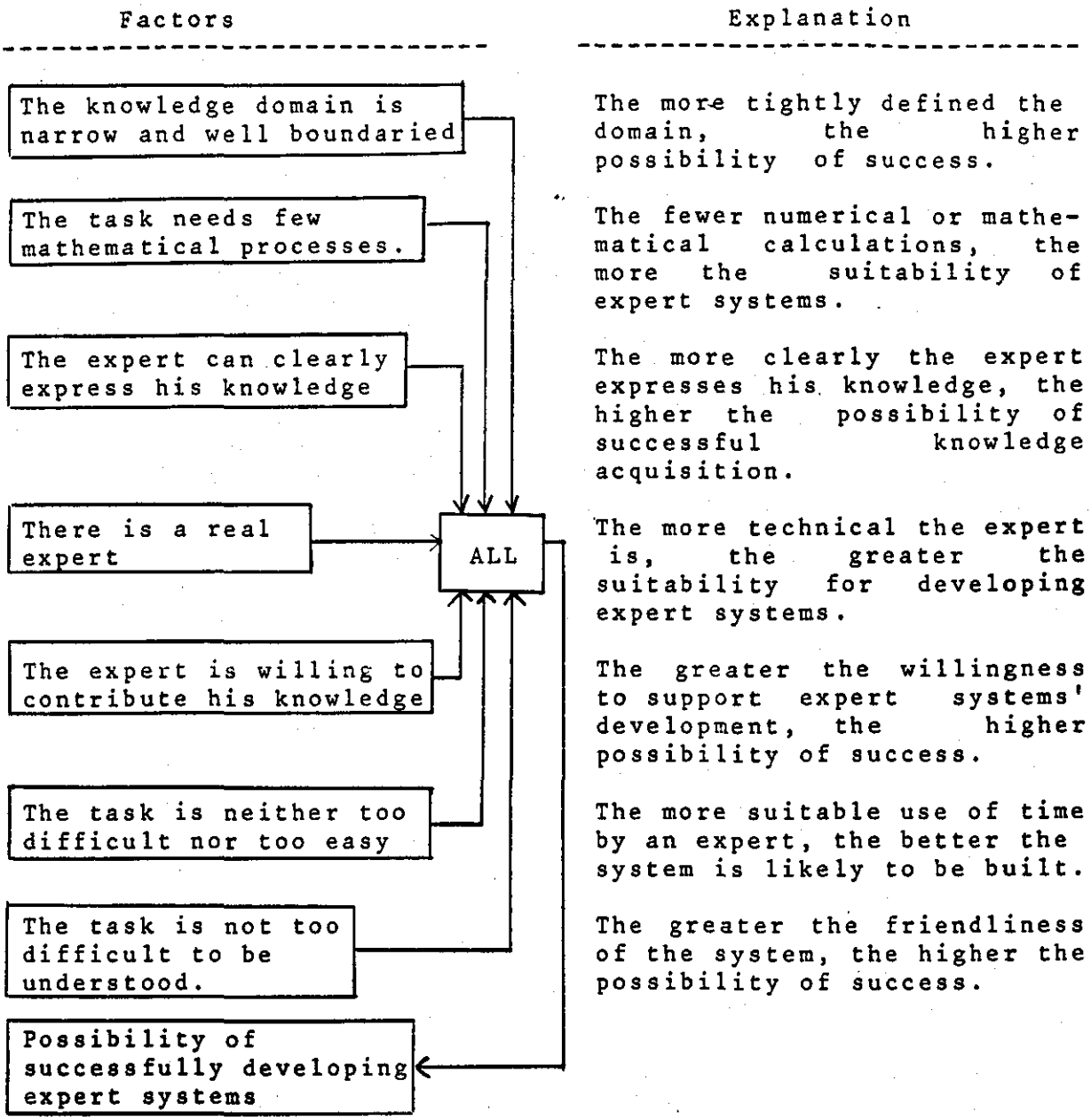


Figure 2.2 The success factors in the creation of expert systems applications

Once a problem domain has been decided, there are still hurdles in the process of building an expert system, they are:

- Knowledge acquisition (elicitation)
- Knowledge representation
- reasoning
- explanation
- tool
- validation

2.3.3 Differences between Acquiring/Representing Knowledge for Expert Systems and Conventional Systems

The process of acquiring knowledge for an expert system is called 'knowledge engineering' whilst the process of acquiring the process of routine transaction for a conventional system is called 'systems analysis'.

The common form of knowledge acquisition, as Blanning (37) states, is 'the elicitation of protocols: experts are asked to state their thoughts as they apply their expertise to real-world problems'. Thus it emerges that the knowledge engineer interviews the domain experts to collect useful information and to represent the information through system structures which usually consist of if....then models. The experts can describe their processing activities in their own terminology. During the process, the knowledge engineer reviews the system logically by himself or asks the experts to review their information as it is represented in model form and inform the knowledge engineer of necessary

corrections to the knowledge base. Verification of the system should be therefore efficient in expert systems because the experts review their information within their professional or technical expertise and in natural language or terminology familiar to them.

In contrast, a conventional system does not provide a flexible approach for process representation. The clerks or the technicians (similar to the 'experts' in expert systems) have to provide a detailed flow of the process by which they handle the data, also documents for necessary input and output are demonstrated to system programmer. These documents are finally interpreted in a computer language and displayed in different ways or formats by the computer. Any error in the system program is difficult to be tested by the clerks or technicians because their 'data' has been re-formatted into a computer program which is now out of their area of expertise and the terminology is totally unfamiliar to them.

2.3.4 Knowledge Acquisition

Knowledge acquisition is the process of gleaning knowledge from human experts prior to implementing the system rules - the knowledge base. However, Welbank (38) asserts that the main difficulty in the production of an expert system is almost universally acknowledged to be knowledge elicitation. Also, Grover (39) indicates that there is no recognised methodology for the process of knowledge acquisition. Success depends on the

complexity of the knowledge structure and the problem domain being tackled. Kidd (40) points out four reasons that make knowledge elicitation difficult and time consuming:

. Human knowledge is complex and messy and often ill-formulated.

. Humans find it difficult to articulate what knowledge they have and how they use that knowledge to solve problems.

. The more expert someone becomes at a task, the more 'unconscious' his or her knowledge becomes.

. The data one gets from using knowledge elicitation techniques are in the form of an expert's verbal comments or his actions. These need careful, even skilled, interpretation as to what underlying knowledge they imply.

Indeed, people regard their knowledge as expertise which is often learned from practical experience. They know 'how to solve it' in a skillful way but most of them do not know the way to express their knowledge logically. Hart (41) recommends four methods of knowledge elicitation in her paper of 'Knowledge Elicitation: Issues and Methods':

- Interview
- Protocol analysis
- Induction
- Repertory grid technique

Interview

Interviewing experts for the elicitation of knowledge is the most common method of fact-finding, the trouble is the knowledge

engineer may not have a list of detailed questions to ask whilst the expert may not know how to describe his knowledge logically. Furthermore, the expert may frequently use technical terms and the knowledge engineer must make sure that he has the same understanding of the terminology which the expert is using.

The blackboard technique of having a group of experts meet together on a model of the expertise is one of the variational techniques introduced to overcome the difficulties of interviews. The experts argue between themselves until they are all satisfied with the details. The knowledge engineer can understand the reasons behind their discussions.

Another alternative is to allow the knowledge engineer to analyse the knowledge away from the expert and present his findings to the expert. This provides the opportunity for the expert to comment on something close to the final version of the knowledge to be represented.

However, Hart comments that the main problem with these methods is that they lack overall structure. The consultations are likely to be lengthy, and until a recognised methodology is developed the output will be difficult to analyse.

Protocol Analysis

This is based on a transcripted interview, but with attempts to structure the process and produce more meaningful results. The

interview is conducted along the lines of comments on specific examples or documented cases, such as laboratory reports or case histories. The knowledge engineer should find it easier to detect general patterns because the expert may emphasis one particular feature of each example and it is easier to structure the expertise into concepts.

Alvey, Myers Greaves (42) report that protocol analysis is difficult to make comprehensive in diagnostic systems because the 'harder' problems have a completely different structure from the common ones and they are also easily omitted from the discussions of documented cases.

Induction

Induction is used because of the same problem encountered by both interviews and protocol analysis, the expert feels it easier to refer to specific examples than to describe his processes. Therefore, the induced rules apply to the examples, however, one can not be sure that the results are correct. The quality of results depends on the attributes chosen and the particular examples used.

Some attributes may not appear in the induced rules because they are less important or because of their high correlations with other attributes which are represented in the induced rules. All the situations should be discussed with the expert. Hart describes the usefulness of induction because 'it identifies

questions and provokes discussion with the expert about gaps, contradictions and redundancy'. Discussion of rule induction through an expert system shell will be detailed later in this Chapter 4.

Repertory Grid Technique

This method is based on the psychological problem that 'much of the expert's expertise lies in the way in which he views problems, i.e. his perception or insight' (43). Every one has his own method of devising his own personal construct to analyse problems. The repertory grid is a method of investigating such a model. The model consists of elements and constructs. The constructs correspond to the attributes of rules, except that they must be bipolar, e.g. black/white, true/false. Constructs are the way in which pairs of elements can be described, e.g. A is strong but B is weak; C and D are both true. Elements are analogous to examples in induction, they are chosen by the user on the condition that they are most relevant to the problem.

One of the essential things in the repertory grid method is to define a particular problem and ask the expert to think about it. He then produces elements and constructs which he thinks are relevant to this particular problem. The grid is the structure of cross-references between constructs and elements for that problem. The expert is forced to investigate his opinion of the problem and the success of this method relies on the user's happiness with the result.

An example of 'recruitment' using the repertory grid technique is given in figure 2.3:

Investigation of Candidates

Constructions		Elements						Constructions	
		E_1	E_2	E_3	E_4	E_5	E_6		
C_1	Not good educational background	4	3	2	5	1	1	C_1	Good educational background
C_2	Non-working experience	3	2	3	3	4	1	C_2	Years working experience
C_3	Bad leadership	5	2	1	4	4	3	C_3	Good leadership
C_4	No potential	3	3	4	3	2	1	C_4	Good potential
C_5	Low salary demanded	3	2	4	5	1	2	C_5	High salary demanded
C_6	Bad communication skill	1	3	2	3	4	5	C_6	Good communication skill

Steve (Good)
Richard (Bad)
David (OK)
Neil (Very Good)
Rody (OK)
William (Poor)

Figure 2.3 Investigation of candidates - using a repertory grid

In this example, the expert (Manager) was asked to evaluate the candidates' qualifications to fit a particular position. Before the consultation, he was unable to clearly define the qualification that a successful candidate must possess. Therefore, he was asked to list the names of candidates and the characteristics of each candidate. Then he was also asked to rate their suitability for each characteristic. From figure 2.3, he gave six candidates' name as the elements of the recruitment, also he listed the requirements of qualification, such as education background, working experience, leadership, salary demanded and communication skill, etc. These requirements are considered as the constructions of the recruitment. With the bipolar constructions, the manager could give score to each candicate from 1 to 6, 1 is the worst score whilst 6 is the best. By calculating the total score given to each candidate, the manager could then find that Neil should be the best candidate who satisfied the requirement the most; Steve was the second one to be accepted, whilst William was the one who would not be considered as a qualified candidate as he scored the least. From this figure, the manager should be satisfied that he is able to describe the main qualification for recruiting a right person, also, he should be satisfied that he has found the right person for the right position. In a similar mannar, the knowledge engineer is able to identify the important issues for building up an expert system to help recruit qualified staff.

This method can be used when the problem is relatively poorly defined because the expert would be very unclear about the relationship between the constructs whilst induction would provide little help in this situation.

2.3.5 Knowledge Representation

Like knowledge acquisition, there is no unique representational formalism for knowledge representation (44). Three common methods of knowledge representation are introduced in this section:

- Rule-Based Representation
- Semantic Network Representation
- Frame-Based Representation

Rule-Based Representation

A rule-based representation is the most common form of knowledge representation. Rules are represented in 'IF abc THEN xyz' statements.

In a rule-based expert system, domain knowledge is represented by a set of rules which are responsible for determining whether those given facts or information match the current condition. When the antecedent of a rule - IF part is satisfied, the consequent of the rule - THEN part is executed.

The match between the antecedent and the given facts produces the inference chain. In rule-based systems, three common inference chains are forward chaining, backward chaining and bi-directional chaining. These three chains have been introduced in section 2.2.

Semantic Network Representation

Semantic Net is used to describe knowledge based on net construction. A semantic net includes nodes and arcs. The nodes represent objects, facts or concepts, the arcs usually represent hierarchies which include ISA (is a) and ISPART (is part). For example:

Considering the case of defining the facilities of a ship, the first one, Queen Mary, is a mailboat; the second one, M.V. Philip is an oiler. Both engine and boiler are components of a ship. In this example, they are stated only once in the lower level which describes a ship. The relationships can be described as follows:

Queen Mary ISA mailboat
M.V. Philip ISA oiler

Mailboat ISA ship
Oiler ISA ship

Swimming pool ISPART of the mailboat
Boiler ISPART of the engine
Engine ISPART of the ship

The relationship between ISA and ISPART constructs the inheritance hierarchy in the semantic net. This means the lower level of the hierarchy in the net inherits the higher level of the hierarchy in the net. In this example, the net can deduce the truth of 'Queen Mary has boiler' by using the knowledge represented by the arcs. This approach can save considerable space when dealing with hundreds of ships, because every ship has a boiler and engine.

Frame-Based Representation

Marvin Minsky, the author of the 'frame' concept, suggests that a knowledge base can be broken into modular chunks ('frames') (32).

The idea is to organise some objects or events by certain standard properties or relations to form a prototype for problem-solving. Therefore, a frame is a collection of certain features which store the values of attributes and describe the nature of an object. In frame-based systems, the features are called 'slots'. Features can be default values or a set of rules. In the ship example, the Queen Mary mailboat is the object and there are features for its properties, also there are some properties which allow for default values. The default values in this example are that the mailboat has at least one engine and one boiler. Default values are useful in frame-based knowledge representations when exceptions are rare in a particular domain. The following example shows the frame of knowledge for the Queen Mary mailboat:

Queen Mary Mailboat

Slots (Features)	Entries (Values)
Name	Queen Mary
Condition	Rusted in outward appearance
Function	Mail transport
Number of engine	Default: 2
Boilers	Default: Yes
Swimming pool	Default: Yes
Useful life	If needed, check the condition of engine or ask help from ship surveyor

In this example, useful life is the slot of procedural instruction for determining other results and is undefaulted but may be needed by the user.

In the author's view, frame-based representation is more or less similar to the concept of a database retrieval system. In the above example, those slots (features) can be input into a database system. However, the main difference between frame-base and database is the default values of those slots in frame-base. These defaults represent those common values that posessed by an object. Unlike frame-base, these defaults must be input into database system item by item for each object.

2.3.6 Reasoning

Rules are the core of expert systems. A rule consists of two parts: the antecedent and the consequent. If the antecedent of a rule is true, the consequent is concluded. In fact, rules do nothing by themselves, they must be associated with the inference engine of expert systems in the consideration of specific problems. The major methods of rules applied by the inference engine have been discussed in the section 2.2, they are forward chaining, backward chaining and bi-directional chaining.

It is necessary that an expert system should be able to reason with uncertain or incomplete information. Human experts would use weighting factors plus judgment to reach a conclusion and make their decision.

There are several approaches for reasoning with uncertainty in expert system building. Three of the most common are to be investigated in this section:

- Fuzzy Logic
- Certainty Factors
- Bayes's Theorem

Fuzzy Logic

This approach was developed by Zadeh (45) for the accommodation of commonsense knowledge which is difficult to adapt by means of conventional logic. In fuzzy logic uncertainty can be tackled on the assumption that the relevant statements are not finite, that is, using a fuzzy set of fractional values between zero and one to indicate the degree of truth of an antecedent. Using an example for diagnosis of catching cold, the factors may be any of a sore throat, a high temperature or a running nose. Relationships among these factors can be expressed by figure 2.4.

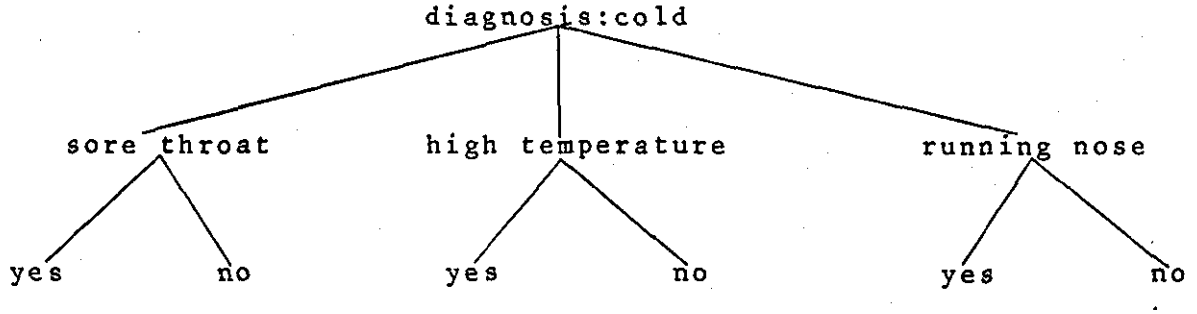


Figure 2.4 Relationships of factors which constitute a cold

When asking the user if the patient has caught a cold, the response could be

Yes (0.8)

This means 0.8 is the degree of certainty which comes from the user that the patient has caught a cold, and that there is 0.2 uncertainty that he is not ill. Applying this concept to the rule:

If the throat is sore then the diagnosis is cold

this rule concludes that a patient is ill if he has a sore throat, however the patient is not completely ill, it might because that he spoke too much. Therefore, the user can only give 0.8 certainty that a sore throat is caused by a cold.

So suppose one has the following:

If the throat is sore and the temperature is high then the diagnosis is cold

With the certainty value:

diagnosis {throat (0.8), temperature (0.6)}

This problem is more complicated. For a more complex problem with more than one antecedent variable, the values are combined in a particular way as follows:

$$P(A \text{ And } B) = \text{Min}\{P(A), (B)\}$$

$$P(A \text{ Or } B) = \text{Max}\{P(A), (B)\}$$

Here, P(A) is the probability that event A will occur, etc.

Applying this method to the first rule:

$$P(\text{throat and temperature}) = \text{Min}(P(\text{throat}), P(\text{temperature}))$$
$$= \text{Min}(0.8, 0.6)$$
$$= 0.6$$

Therefore, fuzzy logic can translate the non-numeric information into figures that incorporate an element of commonsense knowledge to enable manipulation in the absence of complete knowledge. But if applying the above method to the following rules, the situation will cause fuzzy logic to a conflicting result:

If throat is sore and temperature is high then diagnosis is cold.

If throat is sore and nose is running then diagnosis is cold

Provide the certainty value of 'nose is running' is 0.5, the possibility that the patient has caught a cold under the second rule is 0.5 (i.e.Min (P(throat), (nose)) =Min((0.8, (0.5))= 0.5).

Therefore, the conclusion of the diagnosis is cold has two conflicts, i.e. 0.6 and 0.5.

Fuzzy logic does not indicate how these conflicting values are reconciled, the resulting certainty value could be the maximum, minimum, average or others depending on the system designer.

Certainty Factors (CF)

This approach was developed by Shortliffe(46) in the MYCIN project. It uses two components for measuring factors of opposite sides - a belief factor (MB) and a disbelief factor (MD), both within the range from zero to one - to indicate the degree of certainty. The certainty factor is calculated by taking the difference between the two components, i.e. Certainty factor (CF) = belief factor (MB) - disbelief factor (MD). This formula incorporates the resultant certainty factor range from -1 to +1. -1 represents the degree of certainty that a statement is totally false; +1 represents that a statement is totally true. The values between -1 and +1 represent the degree of belief/disbelief, whilst zero represents unknown.

Generally, rules are written in the following format:

If X then Y with certainty factor CF

The certainty factor is useful to overcome the lack or incompleteness of information in Fuzzy Logic by using the following formula:

$$MB[h:e_1, e_2] = MB[h:e_1] + MB[h:e_2] * (1 - MB[h:e_1])$$

Here, h represents the hypothesis given, such as 'if throat is sore and temperature is high then diagnosis is cold'. The measure of belief is updated by giving evidence 1 (E_1) and evidence 2 (E_2). Applying to the 'cold' example, the rules are repeated as follow:

Rule 1:

IF 1). throat is sore, and

2). temperature is high

THEN there is suggestive evidence (0.6) that diagnosis is cold

Rule 2:

IF 1). throat is sore, and

2). noise is running

THEN there is suggestive evidence (0.5) that diagnosis is cold

Using the values used before for those three factors as follows:

Cold:

throat is sore (0.8)

temperature is high (0.6)

Nose is running (0.5)

Using an AND/OR tree which discussed in previous section, the hypothesis that 'IF throat is sore and temperature is high and nose is running THEN diagnosis is cold' with its three factors (events) can be expressed as follows:

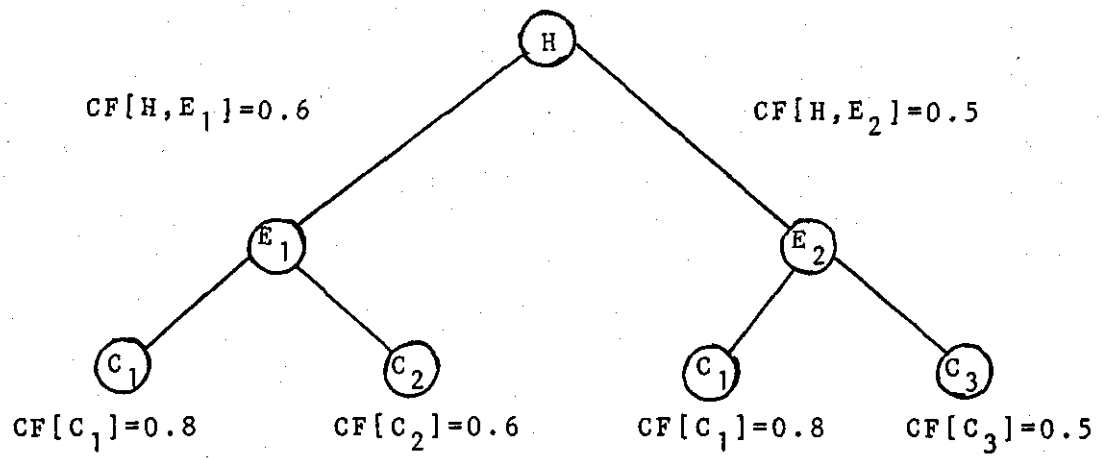


Figure 2.5: AND/OR tree for hypothesis H

Certainty factor (CF) for hypothesis H can be computed as follows:

$CF[E_1]$=Min (0.8, 0.6)= 0.6, $CF[E_2]$= Min (0.8, 0.5)= 0.5

Therefore, applying the formula

$$MB[H:E_1,E_2] = MB[H,E_1] + MB[H,E_2]* (1-(MB[H,E_1]))$$

In fuzzy logic, two conflicting values are 0.6 and 0.5. Using the certainty factor, the resulting measure is:

$$MB\ [Cold: 0.6, 0.5] = MB\ [Cold: 0.6] + MB\ [Cold: 0.5] * (1 - MB\ (Cold: 0.6))$$
$$= 0.6 + 0.5 * 0.4$$
$$= 0.80$$

The value is higher than each single value, i.e. 0.6 and 0.5. This method overcomes the lack or incompleteness of information in Fuzzy Logic.

Bayesian Theorem

In the certainty factor, the values of belief/disbelief factors are probabilities. The Bayesian method represents domain knowledge as probabilities, including prior probabilities of outcomes and conditional probabilities of problem features given each possible outcome (47). The previous two methods both have the same shortcoming, the lack of a reasonable theoretical basis. Although the Bayesian theorem has the required theoretical base, the approach is not implemented to as large an extent as the previous two, because of the difficulty of assigning values for prior probability and the large amount of data necessary to be input for conditional probabilities, whilst these data are not often available(48).

2.3.7 Explanation

Explanation is an important factor in evaluating the friendliness of an expert system to a user. It helps people understand expert systems, and perhaps the user would rather consult an expert

system than an expert. If the expert system provides an 'unusual' or 'difficult-to-understand' piece of advice, the user is able to obtain an explanation from the system. However, most of the explanation facilities of existing expert systems, including shells, provide their explanations by simply copying the reasoning rules. As Kidd (40) indicates, they consist of a trace of the rules used in the reasoning process with some degree of syntactic doctoring depending on the program. Although such explanations do provide proof of the correctness of the conclusion reached, they do not really provide the appropriate method of explanation wanted by the naive users.

Therefore, Kidd suggests four important considerations for providing acceptable explanations:

. Knowledge about the user, in order to communicate an adequate understanding of the problem solving process to the user.
. Knowledge about the underlying causal mechanisms in a domain, in order to justify the relevance and utility of the performance level rules.
. Knowledge about the decision-making method employed by the system which is currently implicit within the system and cannot be used in explanations.
. Knowledge about how good explanations are constructed.

2.3.8 Tools

There are three major kinds of tools for building an expert system - programming language, toolkit and shell.

Programming Language

Traditional programming language such as COBOL, PASCAL, etc, have not proved to be well suited to AI applications, because these languages can not represent the real-world knowledge ideally. Lisp and Prolog are two newly developed languages mostly used for AI-related systems.

It is interesting to note that the Lisp language is very popular in the US whilst Prolog is favoured in the U.K., Europe and Japan (32). For practical purposes, these two languages provide English-like programming ability for knowledge representation and integration of other facilities into the system.

There is much debate concerning the advantages and disadvantages of using Lisp or Prolog, however, the respective advantages or disadvantages of the two languages are widely agreed. Table 2.1 summarises Simons (32) and Johnson's (49) viewpoints on the advantages/disadvantage of Lisp and Prolog:

Advantages	Disadvantages
Lisp	
. Suited to large systems	Suited to expensive workstations or supermini computers
. Easier to provide the necessary procedural information in efficient manner than Prolog.	Programs are often written with need for implementation-oriented concepts.
Prolog	
. Easier for a novice.	Difficult to provide necessary procedural information in an efficient manner.
	Difficult to maintain and comprehend for large programs.
. Suited to small systems.	
. Provides the programmer with generalised record structure manipulation facility.	
. Programs can be written without the need for implementation-oriented concepts.	

Table 2.1 Comparative advantages/disadvantages of Lisp and Prolog

Toolkit

The toolkits are usually implemented on Lisp machines and have a good user interface. They provide the system developer with flexible development tools which include a variety of methods of knowledge representation. In the European toolkits market, the American products Kee, Art and Knowledge Craft are the market leaders (50). Because Lisp is mainly used in the States, the European response to these toolkit products is not so widespread as that of the shells.

However, a number of European toolkits are being developed, in U.K. a major project for building an expert system toolkit is under the Alvey programme for developing a 'flexible toolkit for building expert systems'. Three parties have participated in the project, they are GEC Research, GEC Avionics and Edinburgh University's Department of Artificial Intelligence. This project has been underway since the beginning of 1985 and is to last for three years with a budget of £1.2 million (information from: Commercial Expert Systems in Europe, Ovum Ltd. PP. 59-70 , (50)).

Although the major feature of toolkits is that they are very flexible, so that the system developer is allowed to choose the method for knowledge structure, this flexibility also brings the disadvantage that a developer might not be able to select the right one. Also, prices of toolkits are much greater than that of the shells. For example, all of the American toolkits sell at more than $50,000.

Shells

For developing an expert system by self-programming, selection of a programming language as an application tool is necessary. There are not many customers for an expert system who have the time or expertise available to develop a system from the very start, also, the idea that 'a couple of problem domains can be tackled by the same inference engine' is the main advantage of an expert system shell. Therefore, McLening (51) points out that the quickest and cheapest way of acquiring an expert system is by buying a shell - a package empty of information but with the rule structure, or inference engine, already in place. Here an expert system shell can be defined as 'a computer program with logical reasoning ability, explanation facility, but empty knowledge base'.

Shells are regarded as appropriate for familiarisation and experimentation with expert system techniques and building of smaller expert systems (50,52). There are several factors affecting the choice of a shell to be used. The elements for selecting a shell are discussed below:

Factors for Choosing an Expert System Shell

Koppen of Philips C ISA/CAD Centre in Netherlands (53) conducted a detailed review of four shells - EXTRAN-7, SAVOIR, S1 and Rulemaster by using the following factors:

(a). Knowledge representation: Rule-based representation by describing knowledge into IF....THEN is the popular method. However, there are still many shells where the knowledge is represented in frame-based form.

(b). Inference: This is the way the knowledge is driven for reaching conclusions. The control strategy of backward or forward chaining strategy for diagnostic systems or procedural strategy for teaching systems are required. Also, the capability of dealing with conflicts when more than one rule is applicable to a certain situation is sometimes needed.

(c). Interfaces: Explanation facility is the first consideration for interfacing with the user. Also, natural language for messages to communicate between the machine and the user, and the possibility of linking the shell with other computer systems are important.

(d). Hardware configuration: The hardware on which a shell can be run is important; not all shells can be fitted to a specific machine.

(e). Costs: There are more than 40 shells available on the European market, the price range is from hundreds to thousands of pounds. PC versions of shells are usually much cheaper.

Using an Expert System Tool for Business Applications

There are three important roles in using tools for applications - the knowledge engineer, the expert and the user. It is possible

that these three roles may be acted by any of the following types:

- The expert is not only the knowledge engineer, but also the user.
- The expert is the user whilst the knowledge engineer builds the system.
- The expert is the knowledge engineer who builds the system for the user.
- The expert, knowledge engineer and user are totally different people.

There are no acknowledged guidelines for which approach is the best. However, Hemus (54) suggests that if the problem domain has only one major domain expert, it is best to train the domain expert to build his own knowledge base. This suggestion may not always be workable, it depends on the size of domain, the expert's qualification, enthusiasm and time available, etc.

Although an expert system shell is convenient for business applications, there are still limitations in the area of knowledge representation. The knowledge engineer is therefore not able to be completely free to represent the knowledge. Kidd (40) warns that one should avoid buying a commercially-available shell because of the inflexibility. However expert systems shells have their advantages and disadvantages: using a shell or not for developing expert systems depends on the judgment of those who actually involved in developing an expert system application in business.

In conclusion, the toolkit is the middle option of those three tools - programming languages, toolkits and shells. Generally, programming languages offer the highest flexibility in development whilst the shells offer the lowest. To the developers, the crucial point for the development of an expert system application is choosing the most appropriate tool, however, the developers may just choose the language, toolkit or shell which they are familiar. The choice depends on both the type of problem that the expert system is expected to solve and the role the expert system acts within the organisation. In principle if the knowledge can be represented within the framework of a shell, there is no need to use a toolkit, or if developing an expert system is only for the purpose of experimentation or familiarity with expert system technology, then a shell is enough. Whilst if there are more ambitious goals such as developing a more sophisticated system in the future or expecting more substantial experimentation, then a toolkit or language may be more suitable.

The author has developed a workable expert system by using a shell. For accomplishing the development, a number of shells were surveyed. A detailed survey of four shells (Xi Plus, Crystal, Expert Edge and SuperExpert) and a brief survey of another four shells (ES/P Advisor, Easy Expert, Micro Expert and Micro SYNICS) are outlined in Chapter 4.

2.3.9 Validation

Validation is the process of testing the agreement between the process conducted by the system and the real expert. Unlike conventional systems, expert systems deal with unstructured data, the need for validating them is even greater especially for those diagnostic systems. Sell (22) recommends five requirements for validation:

(a). Consistency - The system should produce a similar answer to a similar question.

(b). Completeness - The knowledge base is sufficiently wide in its coverage to allow the system to tackle successfully any problem within its domain.

(c). Soundness - The system comes to the right conclusions which is in agreement with the expert's judgment.

(d). Precision - An extension of the requirement for soundness. The system makes correct probabilistic or qualified judgment.

(e). Usability - Similar to user-friendly, this requirement is that the interaction between the user and the system should proceed as intended by the designer.

The above requirements for validation overlap somewhat with the features of expert systems. Finlay et al. (55) comment on Sell's recommendation for validating expert systems, they consider that Sell's approach can be useful for discussing the validation of mathematical models which are considered inclusive of logic and data models. Also they recommend that defining the variables

used so as to remove ambiguity in relations is necessary. Therefore they suggest using accuracy to replace soundness. Moreover, they consider both accuracy and precision as two elements of exactitude. Therefore, there are three requirements to be considered in validating a logical model: consistency, completeness and exactitude (includes precision, accuracy and definition). On the other hand, only precision and accuracy are required for a data model system.

In Finlay et al.'s paper, two main methods for validation of an expert system are discussed:

(a). Analytical validation - by checking each part of the model, in which each part is checked individually and in association with other, interacting parts.

(b). Synoptic validation - by checking that an acceptable output is achieved for each of a set of inputs.

The above two methods are usually used in combination. In practical terms, system validation should be performed by both the knowledge engineer and the expert. The former tests the knowledge base from logical parts without the participation of the expert, meanwhile the latter examines the system against some selected representative examples to make sure that the system tackles problems in the way he/she normally does.

CHAPTER 3 - A REVIEW OF THE EXPERIMENTAL RESEARCH CARRIED OUT BY THE AUTHOR

3.0 Introduction

The theoretical approach to expert systems, including definitions and a proposal for creating an expert system, has been discussed in the first two chapters. The next two chapters will discuss the author's practical experience of expert systems applications using the following approaches:

a). Visiting: In order to compare the results of research with practical business applications of expert systems, the author needed to visit commercial organisations. Six representative companies in which expert systems have been or are being implemented were selected and interviews were carried out by the author. Each visit was written into a case study report, these six case studies are attached as Appendix A of this thesis.

b). A survey of expert systems shells: There are around 50 shells available on the European market (50), of which around 20 are of British origin. The list of shells available on the European market can be found in Appendix B of this thesis. With the purpose of producing a working expert system by using a shell, the author examined eight shells. The survey and the reasons for selecting a specific shell for developing the system are discussed in Chapter 4.

c). Development of a working expert system: Apart from the visits to the six companies, the author worked with her supervisor to produce a working system. The intention of this development was to improve the author's understanding of expert systems and for the author to gain familiarity with expert systems shells. Furthermore, this development was aimed at helping organisations to select an appropriate financial planning software package. The process of development is discussed in Chapter 4. The knowledge base of the system is detailed in Appendix C.

3.1 The Visits

3.1.1 Reasons for the Visits

As for the development of other technologies that began in the laboratory and then progressed to business applications, expert systems development is taking a similar direction. The objective of this research is primarily concerned with finding out the position of expert systems in business now and the near future in terms of the practical applications. Visits to companies therefore become a necessary methodology in this research for the following reasons:

a). Similar to the development of traditional computing systems that at the initial stage of computing applications, few companies would adopt the new technology as a necessity to business, the author takes for granted that the use of expert systems for business operation is not so popular as that of

the traditional DP systems currently in business. Therefore, any opinion gathered from the visits given by companies on the use of expert systems and for business applications are important factors in ascertaining the current and future position of expert system technology.

b). There are various approaches to developing an expert system: the adoption or indeed rejection of development techniques from business would be helpful to evaluate the currently available approaches and recommend future research directions for expert system development.

c). For the purpose of backing up the theoretical views on expert systems discussed by those literature, a combination of in-house research and practical interview with expert systems users is necessary.

d). As for the reason for not using questionnaire. It can be said that a questionnaire is the tool to use when all the questions are well structured with simple or definite answers. In this research the purpose is to find out about the companies' processes of development of expert systems and their future views of expert systems usage. These answers are difficult to elicit using a questionnaire.

e). As stated, each visit was written into a case study report. These visits were conducted under structured interview, i.e. interview proceeded with well structured questions which were decided in advance. To effectively convert these interviews into reports, a tape recorder was used for recording the

conversations. However, these conversations were difficult to be filled into a questionnaire form.

3.1.2 Hurdles Encountered when Choosing Companies to Visit

At the stage of choosing companies to visit, the following difficulties were encountered by the author: 1). among the expert system users is a considerable number of large size companies who treat their usage of expert systems as confidential to third parties, thus they refused to be visited; 2). some companies would rather concentrate on developing their own system than waste time being visited; 3). a lot of companies were just beginning to their experimental development of expert system application, they would like the visit to be postponed till their development are finished, whilst the author had not enough time to waste for the postponement.

3.1.3 The Choice of Company

In Chapter 2, Sell's approach to classifying expert systems was discussed. The main reason for choosing the six companies was to match their type of application to Sell's classification. Here the four types of application are detailed as follows:

(1). Analysing data and interpreting meaning

(2). Diagnosing the reasons for or sources between expected and actual states or operations of a system.

(3). Prognosticating the next state or action of a system, specifically give warning of impending malfunction.

(4). A combination of the above three functions,

(5). Teaching and Training new experts.

Among the six companies, the developers of two of these companies - Company A and F, were MBA students of the university. Company B was selected by telephone contact. Companies C, D, and E were chosen based on published articles (56-62) and a thesis (48) that introduced the companies as using expert systems. These six companies are more or less in leading positions in their business fields - Company A is a famous petfoods manufacturer, Company B is a nuclear power designer, Company C is one of the largest communications firms in the world; Company D is the world's largest travel agency, Company E is the leading foreign bank in the U. K., Company F is a multi-national guard security company.

Among those six companies, Company D's expert system is capable of deciding itinerary routes according to the passenger's selection of constraints for his/her journey. e.g. date, cost, destination, etc. The system should be thus categorised as the first group.

The application of Company A is a rather small system with only 38 rules. This system was developed as an MBA project by using an expert system shell for the purpose of eliminating factors that cause product problems in quality. It is appropriate to categorise Company A's system as the second group.

Similar to Company A's system, the system developed by Company F was also for the developer's MBA project by using a shell. This system was also developed for the purpose of fault diagnosis of a security alarm system. The system can also be categorised as the second group.

Although Company B is just at the preliminary stage of investigating expert systems, a project involving a fully computer-controlled plant is being proposed. Once the project is accomplished, the system would be categorised as the third group in Sell's classification.

Company C's expert system is totally self-developed by using an AI language. The system is capable of maintaining a fault tolerant data communications network which solves the problem of handling the regular and inevitable changes of network configuration. The system can be categorised as the third group.

Company E's application is the best example of building a teaching and training system. The system was initially developed for the purpose of teaching the company's customers to be aware of any possible discrepancy in the opening of a letter of credit before they claim payment from the bank.

3.2 Looking at the Business Use

Among the six companies, the extent of their usage of systems can be distinguished by the following classifications according to task:

- investigating expert systems technology, e.g. Company B,
- developing an initial system for experimentation and acquaintance with certain techniques, e.g. Companies A and F.
- having developed their first application as a trial system so that they could realise the capabilities of expert systems and recognise business opportunities for further applications or development of expert system, e.g. Company D.
- having developed their commercially ready expert systems which are available on the expert systems market, e.g. Company C and E. Meanwhile, the companies act as consultants and advisers to their customers.

Approaches in Developing Expert System Applications

As for the approaches in developing an expert system, four companies adopted the quickest way - by using shells, whilst two companies started from the very beginning by using AI languages, OPS 85 or Prolog 2.

Having decided the problem domain and the expert, and having chosen the tool for development, consultancy advice is another factor worthy of consideration in the development of business applications.

Among the six companies, both Company D and E adopted consultancy from outside software companies. Indeed, use of external consultancy advice for developing a company's first application is a good decision when a good knowledge engineer is not available and the expert is inexperienced in computers or even unable to describe his/her knowledge adequately.

It can be seen from the six case studies that none of the expert systems is required to be able to deal with uncertainty factors to any great extent, whilst in most of the literature, textbooks or articles, capability of dealing with uncertainty is one of the most important criteria for evaluating an expert system. In the six case studies, experts are required to express their knowledge with certainty and knowledge engineers do try to avoid any uncertain answer made by the user when being questioned.

In Chapter 2, four methods of knowledge elicitation recommended by Hart were introduced. If one relates these methods to the practical application, it can be seen that the interview is the most common method for eliciting knowledge from the expert. Of these six companies, both Company A and F developed their systems by combining the roles of expert and knowledge engineer, thus these two companies had no problem of engineer - expert interaction in knowledge elicitation.

In Company B, the knowledge engineer is of engineering background rather than of chemical background. He had to interview the

expert for 3 or 4 sessions each an hour's duration, then he isolated himself from the expert to analyse the knowledge and then presented his findings to the expert for further clarification. This approach belongs to the first method - the interview, for knowledge elicitation.

Company C spent 5 or 6 man years developing expert systems. For the process of knowledge elicitation, the company gathered a number of software engineers, knowledge engineers and experts to decide what their system should do. The blackboard technique was used in this company's knowledge elicitation.

In Company D's case a combination of interview and protocol analysis was used. Much of the knowledge came from a written form, such as timetables, from which the expert could express his knowledge clearly in the interviews. The knowledge which was not available in written form, such as the expert's years' experience in tackling clients' enquiries, was gathered by recording the expert's conversations and analysing the transcripts, and by talking through examples with the expert.

Although Company E's expert had no knowledge about computers, especially not in the field of expert systems, his knowledge was the most structured of that of all the six companies. All his knowledge was based on word-perfect memory of the relevant international rulebook of old and new case law and on many years'

experience of the bank's customers. His knowledge was simply elicited by means of interviews.

3.3 Analyses, Reviews and Results for the Visits

3.3.1 Overall Review on the Visits

The following table gives analyses for the visits. This table is composed of 15 factors that can be considered as the important issues for reviewing the six comapnies' developing expert systems:

Company	System type	Approacch used for development	Developer's role	Mathod used for knowledge acquisition
A	diagnoeis	shell	expert & KE	N/A
B	prognosis	shell	KE	interview
C	diagnosis	self-programming	KE	blackboard
D	analysis	outside consultance	KE	interview & protocol analysis
E	training & teaching	outside consultance	KE	interview
F	diagnosis	shell	expert & KE	N/A

	Dealing with uncertainty	Years spent	Investment	Using in operation
A	no	1	N/A	no
B	no	not available	£20,000	not yet
C	no	5-6 man years	confidential	yes
D	no	2	£80,000	yes
E	no	2.5	confidential	no
F	no	1	N/A	no

	Shortage of resources	professional assistance	Deficient in high-level support	Experts' capability of expressing
A	yes	no	yes	N/A
B	yes	yes	yes	no
C	no	no	no	yes
D	yes	yes	no	no
E	yes	yes	yes	no
F	yes	yes	yes	N/A

	Time consuming for K.A	Relied on shells	Relied on expert for validation
A	N/A	yes	yes
B	yes	yes	yes
C	no	no	no
D	yes	no	yes
E	yes	yes	no
F	N/A	yes	yes

Note: KE represents "knowledge engineer"; N/A = not applicable

Table 3.1 Analyses and results of the visits to six companies

System type

In Chapter 2, Sell's(22) second approach for classifying expert systems by the task that expert systems are developed to perform can be used for classification. It can be seen from this table that in the 6 companies visits diagnostic systems were most often chosen as the initial expert system to develop.

Approach used for development

This factor concerns the tools used for building an expert system. As discussed in Chapter 2, the shells are convenient to companies for first experience. Apart from using shells, hiring an outside consultancy for implementing expert systems in companies for first application is another good approach.

Method used for knowledge acquisition

The available methods for acquiring knowledge from experts have been discussed in Chapter 2. there it is indicated that interview is the most common method for knowledge acquisition. This view is not unsubstantiated. The reason that both Company A and F were given N/A for their method used for knowledge acquisition was that both of the two companies' developers were in the position of both expert and knowledge engineer that the knowledge engineer needed no specific method for acquiring knowledge from the experts of themselves.

Developer's role

The types of the role(s) acted by the developer of an expert system had been detailed in Chapter 2. Whether the developer should act as a single or mixed role depends on the knowledge possessed by the developer. The developers in Company A and F had the knowledge about computers and the development shells; therefore, by being the experts in these two companies, they developed their expert systems by themselves; the developers of Company B and C had a computer/engineering background that had no knowledge about the experts' work. Therefore, they had to acquire the experts' knowledge.

Dealing with uncertainty

Although three major approaches for reasoning with uncertainty (fuzzy logic, certainty factors and Bayesian theorem) have been discussed in Chapter 2, none of these six companies encompassed uncertainty into their systems. from the interviews, the author found that this was caused by the following 2 reasons:

1). All the required answers from the users must be unambiguous . For example, Company A, B, E and F. By considering the nature of their problem domain, these companies required their systems to be designed as "single answer to each question".

2). For the purpose of avoiding inconvenience or confusion, the systems were not programmed to deal with uncertainty, e.g. Company C and D. Although the users may not be confident

on giving their answers, the systems prompt them to give the best one.

Although the incapability of dealing with uncertainty does not cause problems for some particular domains, such as those domains tackled by the Companies' visited, an ignorance of the capability of dealing with uncertainty would cause the knowledge representation to be incomplete and would restrict the accuracy of the answers given by the users.

Years Spent

From this factor, it can be found that developing and implementing an expert system is time consuming. Among these six systems, Company A and F's systems were two of the smallest, however, both companies spent one year on development.

Investment

The costs associated with expert systems development were treated as confidential by company C and E, and not available for company A and F. Based on the costs incurred by Company B and D, it can be said that development of a real expert system is likely to cost thousands of pounds.

Using in operation

This factor can be used as an assessment of the practicability of developing an expert system. Although each company can give reasons of developing an expert system, there are still reasons that prevent the system from being used in daily operation. For

example, Company A and F considered their systems as their first experience of expert systems technology, further expansion and development are required; Company E insisted that an expert system would never take over from human beings, especially for the ability of thinking and judgement, therefore, the developed system is only used as a training tool to those inexperienced staff. Although Company D's system is currently used in daily operation, the system is only used by staff inexperienced in computers. Strictly speaking, this system is not fully used in the company's daily business operations.

In the author's view, Company C, D, and E's systems are much more successful than the others - this was evaluated based on the domain chosen and the task the systems can perform. This may be because of the approach and the technology of development applied by the companies, e.g. self-programming conducted by experienced staff in Company C and hiring outside consultancy to ensure the success of development in Company D and E. Also, using the appropriate method for knowledge acquisition is another factor of success.

The systems developed by Company A and F were more or less for the purpose of familiarising themselves with expert systems technology, all the approaches and technology applied in these projects were very simple. Consequently, the tasks the systems could perform were just a small part of the experts' routine jobs.

Since Company B's system is still in progress, the success of the system cannot be measured. However, according to the interview, the author expects that the system will be an another successful one.

3.3.2 Problems Discussion on the Visits

As indicated in the reasons for the visits, the author wanted to find out the factors that are likely to be helpful in the currently available approaches, and forecast future trends and recommend future research directions of expert systems development.

Section 3.3.1 discusses the 8 factors that are important issues in evaluating current status of expert systems in UK business. The author's points of view on the subject will be discussed in Chapter 5 which will be mostly based on the discussion in that section.

This section is concerned with the discussion of problems and difficulties that will affect the future of expert systems development based on the visits to six companies. Discussions on the forecasted future trends and recommended future research directions of expert systems development are detailed in Chapter 5.

a). Shortage of resources:

Here the resources represent all the relevant factors for developing an expert system - the people, the equipment and the necessary softwares, etc. Successful expert systems must be developed by well organised people with good knowledge background. Also, the development need high efficient development tools. Among the six companies, only Company C meets these requirements. This company has good software engineers, qualified experts and sufficient equipment.

Although Company B and D had invested large amount for their expert systems development, these two companies did not have qualified employees who were able to accomplish the development. This resulted in Company B sending the knowledge engineer for intensive training and Company D to seek for an outside consultancy help. As for Company E, since the expert had little knowledge about the computer and even less about expert systems, the company could only also ask for outside consultancy.

b). Shortage of support from computing professionals:

This shortage can be found from the description of process of developing systems om the six case studies. Among the six companies, only Company A asked for its computing department's assistance to conduct a survey of expert systems shells available on the market.

In Company E's case, the computing department simply deals with routine jobs within the confines of traditional DP/MIS activities rather than I.T development. The lack of support from computing professioonals caused this company to pay for hiring an outside consultancy.

Overall, the lack of support from the couputing professionals appeared to cause time to be wasted in development and extra costs in employee training.

c). Deficiency in the support from high-level management:

From the visits to six companies, the author found that most of these companies did not attain the full support from their high-level management. This might because that the top management had not found the real necessity of developing an expert system for the companies' daily operation or even for decision-making use.

This situation would obstruct the desire for any possibility of further development of expert systems. In the author's view, both Company A and F's systems could have been developed much better if the top level management had been involved in the development.

d). The experts were not able to express their knowledge clearly:

From figure 2.2 of Chapter 2, it can be seen that the existence of a real expert is one of the most important factors in the creation of an expert system application. In the cases of visits, although all of the six companies had their real experts, the experts were not all capable of expressing their knowledge clearly. For example, Company D and E, the expert had years of experience and necessary knowledge about how to handle their job perfectly, however, they could not express their knowledge in a logical and thoughtful way during conversations with the knowledge engineers through repeated interview.

e). The time taken to acquire and represent knowledge:

The problem was partly caused by the fact described in d) and partly caused by the method used for knowledge acquisition by the knowledge engineers. for example, the knowledge engineer of Company B had devoted himself to know about the expert's job in detail which would take a long time for not only knowledge acquisition buy also the knowledge representation. Knowledge acquisition is acknowledged to be the main difficulty in the production of an expert system, but it is not necessary for the knowledge engineer to learn all the details necessary for an expert's job in order to capture the knowledge.

f). Too much reliance on the use of shells:

It is found from the visits that 4 companies in total adopted shells as their expert systems development tool. The advantages and disadvantages of using shells has been discussed in Chapter 2. These disadvantage reflect what had concerned the manager of Company D - using a shell would restrict their knowledge representation and confine their plans for applications as well as diminishing their development efforts.

g). Too much reliance on the experts for system validation:

The author found that few of the six companies applied a systematical methodology for system validation. Most of the companies relied on their experts' opinion on the performance of systems. Among the six companies, Company E adopted the most practical way of validation - the knowledge engineer tested the logical part of program, the expert tested the system against actual cases. However, the other companies just relied on their expert's comments on the systems according to their playing with the systems. This would cause the risk of any neglected errors, such as the programming error, the deficiency of expert knowledge, etc.

Comparing the above problems with the author's own experience of building a working system for her project, the first 3 problems are not applicable to this comparison. For the fourth problem of the expert's capability of clearly expressing his knowledge and the fifth problem of time consuming on knowledge acquisition, the author was lucky enough that she and her supervisor were working together in high cooperation that there was no problem with these 2 issues. Success of these 2 aspects is to be discussed in section 4.4. However, the author encountered the same problem of limitation when using a shell. The efforts of seeking for a more powerful tool to accommodate the problem encountered is also to be discussed in section 4.5.

As for the validation, the project adopted a practical way similar to the way Company E did. However, since the system has not yet been recognised as a working system, the term 'validation' should be substituted by 'testing'.

The availability of an expert was the main problem that caused time consuming for the development of the system. This problem was not encountered by any of the six companies. In the author's opinion, it is essential that experts need time for concentrating themselves on organising and thinking about their knowledge.

It is perceived by the author that the above discussions will more or less be important issues for reviewing the current status of expert systems in UK business, forecasting the future trends

of expert systems development as well as recommending future research directions of expert systems in next 5-10 years. These subjects are to be discussed in section 5.2, 5.3 and 5.4 respectively. Also, another two approaches of the research - survey of expert systems shells and production of a working system are to be discussed in Chapter 4.

CHAPTER 4 - PRODUCTION OF AN EXPERT SYSTEM FOR THE SELECTION OF FINANCIAL PLANNING SOFTWARE

4.0 Introduction

There are a large number of financial planning packages available on the market, the crucial points for business to select a financial planning package do not purely lie in the different prices to be compared with companies' budget. There are many factors to be taken into account. It can be said that those factors which decide the selection for a financial planning package would be a good subject for developing an expert system application.

The author's supervisor, with many years' teaching experience in the field of financial planning, has provided consultancy advice to outside firms on solving their problems of financial planning, one aspect of this consultancy is the type of financial planning package for organisation to acquire.

In order to experience at first hand the problems associated with the development of expert systems, the author was involved in a project to produce a working expert system which would give advice to organisations on the selection of appropriate financial planning packages. During the development, the author's

supervisor was acting as the expert so the knowledge was given by him. The author structured and represented the knowledge in an inferable form and built up the system using an expert system shell.

This chapter begins with the literature review of the issues which could provide prescriptive data for the construction of expert systems, then follows a discussion of the methods and approaches applied. In the discussion of the selection of a specific shell for carrying out the project, a survey of eight shells is conducted. This review is necessary for comparative purposes before deciding to use a particular shell as the development tool.

4.1 The Literature

Although there has been much literature produced on the subject of artificial intelligence and expert systems, especially during the past few years, and there have been many articles about practical business applications, most of these articles do not discuss the details of an application, such as the knowledge structure, this is even more apparent in those articles which discuss the application of expert systems in financial planning. For example, Humpert and Holly(63) have given a review on several expert systems developed for financial planning purpose whilst there is no discussion about the structure of knowledge in their review. Also, as Bramer(64) indicates, the area of financial application on expert systems is notably a new area in the U.K.

that many of which are still under development and they have not reached the technical literature. Both of these two articles do not discuss the knowledge of selecting a financial planning package.

There were two dissertations available written by Hemus(54) and Dickman (65) at the time of writing. They have developed their own diagnostic expert system by using Xi Plus respectively as part of their MBA project at Loughborough University of Technology.

The author's project was to develop a working expert system for selecting an appropriate financial planning package by using Xi Plus. Reasons for using this shell will be discussed later. However, the projects completed by Hemus and Dickman were diagnostic systems for the production process control or security alarm system, not advisory system for financial planning.

Literature discussing relevant knowledge for structuring the selection of computer software can be found in Finlay's papers (66,67) of 'Mathematical Modelling and Expert Systems' and 'Using an Expert Approach to Structure the Selection of Computer Software', and another paper 'Administering Rule Development in Rule-Based Expert Systems' by Finlay, King and Burnett (68). None of these papers discuss as the development of a practical expert system by using Xi Plus for selection of financial planning packages as a management tool in business.

4.2 Approaches for Development

4.2.1 Why Choose a Shell to Carry out the Project?

To most of the expert system developers, the first decision they have to make is whether to choose a shell, a toolkit or a programming language. Features of these three kinds of tool have been discussed in Chapter 2. In this project the shell was chosen as a tool for development based on the following reasons:

a). As mentioned in Chapter 2, the cheapest and quickest way of acquiring an expert system is by buying a shell. Cost is the major factor which affects choice, because most of the toolkit products are priced over $50,000.

b) Resources and timing are further considerations. The learning curve required for Prolog programming is considerable in view of the time constraints imposed upon the author of one year's duration for the research.

c). It was assured that the developed system would be a small enough system whose knowledge a shell would be capable of handling. Despite the inflexibility of expert systems shells that restrict the developer to a narrow choice of knowledge representations and inference mechanisms, the project was decided to be developed by using a shell.

4.2.2 A Survey of Expert Systems Shells

As mentioned before, there are a large number of commercially available shells for sale on the European market. The rapid

growth of shells reflects the advantages of using a shell for business application.

There have been quite a few papers published which review shells. This section evaluates eight shells based on the following sources:

- Direct use of shells, e.g. Xi Plus, Crystal, Expert Edge and SuperExpert and the user's manuals of these four shells (69-72). These four shells will be examined in detail later in this section.
- Learning with the NCC Starter Pack which consists of Micro Expert, ES/P advisor, Micro SYNICS, and Expert Ease as the initial step to having an elementary knowledge of expert systems(73-77). These four shells will be examined briefly in this section.
- A report on 'Evaluation of expert system shells for construction industry applications' issued by the Civil Engineering Department of Loughborough University of Technology(78).
- A report published by Ovum Ltd. (50) which introduces a number of expert systems shells.
- A survey of tools for getting started in expert systems published by Data Processing journal (79).
- A 'Software Review' of Xi Plus written by Forsyth (80).
- Two booklets which introduce 'know-how' programming issued by Expertech (81).

- A comment on the Crystal expert system shell by Linderholm (82).

When describing the shells' characteristics, a simple example of 'the place to go on holiday' is used to explain these packages. The approach is adopted so that the reader can easily gain an idea of each shell's function and utility. A detailed examination of four packages is given below:

Xi Plus (Produced by Expertech Ltd.):

Xi Plus has been actively on sale since 1986 and is an improved version of the original Xi Package. Written in Prolog, this package is emphasised by Expertech as using know-how programming which in other words expresses human experts' knowledge in rule format, i.e. if x then y. Thus the knowledge is primarily represented as IF....THEN rules. The constructed rules can then allow users to query the system for particular problems and ask for the necessary explanation of reasoning.

Apart from rules, Xi Plus supports facts, demons (to be discussed later), default values and related questions. Relations amonst all of the variables are mostly predicated by " is ", " includes ", " is a " with some others specified by the developer.

The following example represents the basic contents that form an application in Xi Plus, they are questions, rules, and queries.

Questions:
---

question 1

day is

holiday

not holiday

question text: What day is today?

question 2

weather is

sunny

raining

question text: What is the look of the weather?

Rules

If day is holiday

and weather is sunny

then place is park

If day is holiday

and weather is raining

then place is home

If day is not holiday

then place is office

Query
---

query 1

place

When starting a query, Xi Plus will ask the type of the day. If the user's answer is 'not holiday', then Xi Plus stops the query and gives the conclusion as 'place is office'. Whilst if the user's answer is 'holiday', then Xi Plus asks another question about the weather. If the answer about the weather is 'sunny' then Xi Plus will give its conclusion as 'place is park', otherwise, the conclusion is 'home'.

From the above, it can be seen that Xi Plus has the advantages of ease of use and use of natural English. In this example, the knowledge base is entered through the keyboard in any order. Interaction with Xi Plus can be either using menu screens or typing commands.

The above example is for querying the decision (the place to go). The user is required to enter at least one query which comes from the consequence statements. The system will then determine a suitable answer from the represented knowledge base by asking necessary questions.

Although rules can be input in any order, Xi Plus reaches its conclusion by working through the rules in the order in which they are entered. Therefore, pre-ordered rule input can make Xi Plus work more efficiently. There is an internal task list called 'agenda ' in Xi Plus, which controls the working path for each reasoning. At every step in the reasoning process, Xi Plus consults its agenda for the next step. By using a 'demon' it is

possible to force Xi Plus to consult rules that have a higher priority than normal rules. This is useful for saving time spent on running a set of rules which are invoked in a cycle of backward or forward chaining according to the sequence.

A demon is initiated by a keyword 'when' instead of 'if'. The inference in Xi Plus is capable of both forward and backward chaining. This is quite flexible and convenient for knowledge engineering, especially as for complex applications the knowledge engineer usually requires an expert system shell to be able to control the path by following the reasoning process with greater accuracy rather than just adhering to its built-in control strategy.

Although Xi Plus is considered so far as the most flexible shell on the market, it has several shortcomings:

1). Speed problem - Xi Plus runs rather slowly on an IBM XT machine, especially when it is dealing with a complex knowledge base. The authors, Expertech, have been aware of the problem of speed in Xi Plus, and although a new version of Xi Plus has been launched in 1987, the problem still exists. It is known that Expertech are dealing with this problem, they consider the task as one of their projects in the near future (83).

2) Incapable of dealing with uncertainty - There is no mechanism e.g. fuzzy logic, certainty factor or Bayesian inferencing built into Xi Plus' inference engine. All

uncertainties must be handled by either the expert or the knowledge engineer himself/herself. For example, for uncertainty about a weather forecast, the following rule might be written:

If low pressure is around
and sky is very cloudy
and humidity is high
then rainfall is very possible

With the user answering question about pressure, cloudiness and humidity. To Xi Plus, both 'very cloudy' and 'very possible' only have the meanings given within the context, they are, in other words, in the same positions as 'holiday', 'raining' or 'sunny' in the previous example.

3). Incapability of handling rules initiated with 'OR'- For some reasoning the same conclusion can be reached by a combination of two or more statements of different conditions. For example:

If weather is raining
or weather is snowing
then place is home

The above rule will violate the rules of language so that the message 'Syntax error' will be given when loading the rule into the Xi Plus knowledge base. The user has to either

separate the above rule into two rules or combine the second statement with the first one as 'If weather is raining or snowing, then place is home':

a). If weather is raining
    then place is home

b). If weather is snowing
    then place is home

4). Poor user interface, especially the explanation facility - It can be said that all the explanations in Xi Plus, which answer the user's questioning about how was the conclusion reached, are just traces of the rules used in the reasoning process. Also, Xi Plus does not give an explanation of the reasons why the user's input is not acceptable. Unless the user is familiar with Xi Plus, it is not easy to find out the exact error simply from the message 'Syntax Error' given by Xi Plus.

5). Deficient documentation - Although user friendliness or ease of use is emphasised by Expertech for Xi Plus, the documentation of the user's manual is not as useful as expected. This causes a long learning curve for Xi Plus - even if the user is able to input his knowledge into Xi Plus within the first two or three days of his using the package, it takes quite a long time to make the 'workable' system operational.

Crystal (Produced by Intelligent Environments Ltd.)

The first version of Crystal was published at the end of 1985. This old version has been replaced by an improved version since 1987.

Crystal is written in C language and runs on IBM PCs and compatibles. It has an inference network which infers with backward chaining. The reasoning in hierarchical structure is begun with a master rule. Rules are entered by putting the goal at the top to be followed by conditions beginning with IF, AND, OR. The previous example of holidays is now used for the demonstration of Crystal. Firstly, the user has to input the goal into the master rule:

Place is park

The user will then have to press the function key F10 to make 'Place is park' as the conclusion of the following conditions:

If day is holiday
and weather is sunny

By pressing the function key F10, the user is allowed to expand the knowledge for any sub-rules. A friendly text for the conclusion can be input by the user by pressing function key F6. For example, the conclusion accessed by Crystal for 'Place is

park' can be 'It is nice to be able to tell you that you are going to the park' by entering these words into the display form.

The major features of Crystal include it's aesthetic design of overlapping windows, e.g. the conclusion display and the master rule so that the user could see many displays of different message at one time. Also, it is able to deal with uncertainty reasoning. Furthermore, Crystal is capable of numerical computation, it has the best calculating facility of the four shells discussed in this chapter. This is helpful for developing an expert systems which is required for numerical calculation, for example, the tax advisory system which is capable of giving advice on the amount of tax payable/exempt, this would need a large amount of calculation.

However, Crystal has its failings which prevent it from being chosen as the tool for developing the expert system of 'Selecting a financial planning package':

1). Incapable of deciding necessary question - Unlike Xi Plus, Crystal is not able to decide which information is required or which question it is necessary to ask. This incapability means that the developer must decide the necessary questions by himself. The system developed with Crystal would not reach the right conclusion if any question is omitted by the developer.

2). User unfriendliness - The developer or user must use Crystal in a sophisticated way, in particular with the use of function keys and commands for building up a system. The complicated way of establishing the knowledge base makes it rather difficult to structure the necessary rules to make a system, especially a complex system.

3). Lack of practical examples for documentation - A rather simple example of 'Credit giving' is given by the Crystal user manual, which is quite easy to follow that it implies its ease of use. However, it is hard to build a practical system for more complicated problem purely based on learning the simple example from the manual.

4). Use of words which are not easily understood without reading the manual, for example, master rule, succeed, test expression, etc. - All of these represent functions which are not immediately obvious to the user from the terms themselves.

Expert Edge (Produced by Helix Expert Systems)

Similar to Crystal, Expert Edge is written in C for running on IBM PCs. It is shown in Ovum's report (50) that by April 1986, 50 % of sales of this product were sold in the USA, only 25 % in the U.K..

Expert Edge uses Bayesian statistics to handle uncertainties and lack of complete information. It employs the backward chaining control strategy. Knowledge representation through Expert Edge

is in the form of 'subject - verb - object if subject - verb - object'. All rules refer to Bayesian probabilities, although the answers can be made definitely affirmative by the user.

Expert Edge rules can incorporate calculation, logical reasoning facts and uncertainty. The facility of name trees allows the user to replace a name by the group organisation of names. This facility also enables the user to write a rule about a specific name in general terms. Using the example of a 'holiday', the name tree of the 'Place' is shown in figure 4.1. In name trees, 'noun', 'undefined', 'constant', etc. are arbitrary structures used by Expert Edge for the organisation of the knowledge base. The terms 'weather' and 'holiday' are subsets of the 'decision'.

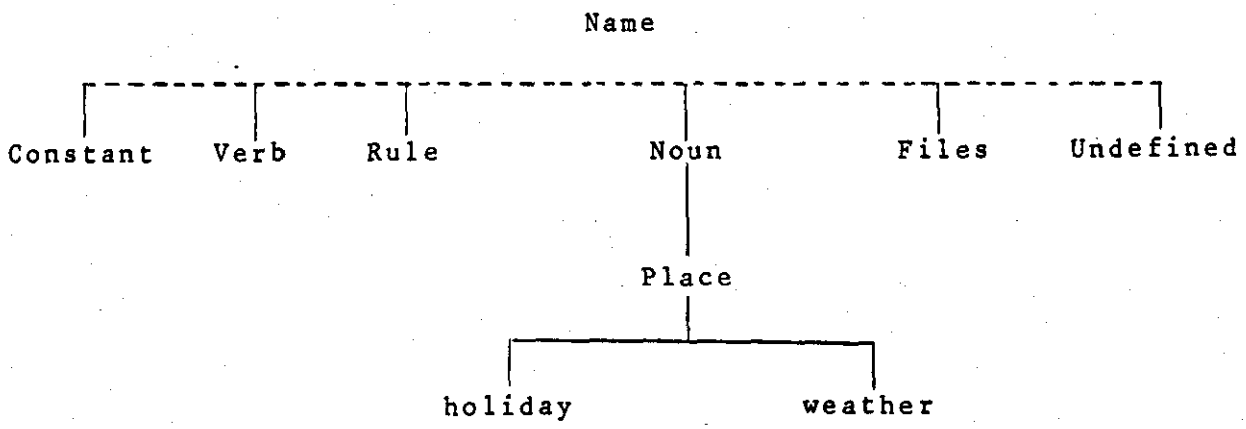


Figure 4.1 Name tree of Expert Edge using example of 'holiday'

Other features are the 'tell' facility and the window system for interactive rule entry and prototyping. The 'tell' facility allows the user to answer questions before Expert Edge asks them and to change answers already given. It also provides powerful facilities for marking questions so that they may be accessed in a group. 'Tell' can be used to volunteer data. Furthermore, Expert Edge has six windows on the screen, the user can obtain several pieces of information at the same time. The six windows are introduced below:

Names of Window	Contents Displayed
Output window	Information, conclusions and progress of interaction.
Dialogue window	A record of questions, answers and conclusions.
Question/Answer window	Questions are asked and answers are entered.
Command window	Main menu commands.
Status window	System status data (system name, date, percentage of memory free).
Message window	Error messages.

There is an extra window available in Expert Edge, i.e. Help window.

Disadvantages of using Expert Edge can be summarised as follows:

1). Unfriendliness - Expert Edge is not easy to use, especially when inputting the knowledge base. Since Expert Edge uses backward chaining, all rules are input from the conclusion part whilst people are more likely to think in terms of forward chaining. Moreover, unlike Xi Plus, statements of rules must be split into three parts - subject, verb and object, each part is separated by pressing 'Return'. The user must constantly remember to press 'Return' for building up his knowledge base. In the example 'holidays', the rule of 'if day is holiday and weather is sunny, then place is park' has to be input by pressing 'Return' 9 times.

2). Too much reliance on probabilities - Although it is possible that the user can use Expert Edge without understanding the statistical theory, the ideas behind the probabilities in Expert Edge are fairly complex. Like the Bayesian theorem discussed in Chapter 2, Expert Edge requires its users to answer questions about prior probabilities for each rule. A user may not be able to decide how often the evidence is true when the conclusion is true/false.

3). Insufficient documentation - There is no sample screen display introduced in Expert Edge's user manual. The tutorial instructs the user with a simple advisory system, but it would be difficult for a user to a practical system using Expert Edge simply based on the user manual.

Using Koppen's (53) approach for a survey of expert system development tools, the following is a summarised table of the above 3 packages - Xi Plus, Crystal and Expert Edge.

Shell	Xi Plus	Crystal	Expert Edge
Knowledge Representation			
Rule-Based	+	+	+
Semantic Net	N.A.	N.A.	N.A.
Frame-Based	N.A.	N.A.	N.A.
Rules			
Input in one screen	+	-	-
Uncertainty	-	+	+
Inference			
Control Strategy			
Forward	+	-	-
Backward	+	+	+
Interface			
User interface			
Explanation	+	+	+
Languages	+	+	+
Hardware configuration			
P.C.(other than IBM's)	+	+	+
IBM	+	+	+
Minimum memory	256 K	384 K	256 K
Costs	£ 495	£ 395	£ 1,250

SuperExpert

SuperExpert is an expert system shell using rule induction to derive rules. It can examine a set of example decisions and generate a decision-making rule from these examples. All commands are available by menu-selection, so changes can be made to SuperExpert quickly and easily. Moreover, new examples can be added or old examples updated or deleted at any time. The enquiry system for a problem can either be completed by SuperExpert or by the user who designed the particular problem, the user can use his own terminology and style.

SuperExpert is very simple and easy to use as it uses examples to induce rules, the user does not need to decide each possible situation for different results. An illustration of the example used by SuperExpert is shown below. Using the example of 'holidays', there are a number of factors that affect the user's selection of a place to go on holiday. These are called 'Attributes' in SuperExpert. The formulation of the attributes and their values for the example mentioned are given below (Note: the attribute heading - class, is given by SuperExpert which is similar to the definition of 'decision'):

Attributes:

Day	Weather	Class
holiday	sunny	park
not	raining	home
		office

The user is allowed to enter his example into the example screen once he/she has entered the attributes and their values. The following is the example used for 'holiday':

Day	Weather	Class
holiday	sunny	park
holiday	raining	home
not	*	office

In this example, * means 'don't care'. The rule in the first line can be read as 'if day is holiday and weather is sunny, then place is park'. After the examples are entered, the user is able to command SuperExpert to induce the rules. The induced rules are listed as follows:

```
SuperExpert: Rule listing
Problem:/ place

day
   holiday : weather
        sunny : park
        raining: home
   not : office
```

In a practical business environment, there should be considerable data for decision-making yet the expert may not know how to translate them into a rule format. Using SuperExpert can solve this problem.

Although SuperExpert is easy to use and thus user friendly, it has a few disadvantages which as given below:

1). Lack of a calculating facility - This is the major disadvantage of SuperExpert. Any calculation required for building an expert system must therefore be done by another program. This disadvantage may be obvious when developing an expert system which is required to handle numeral calculation, this disadvantage can be referred to the advantage of Crystal's capability of handling numerical computation discussed before.

2) Reliability of data completeness and accuracy - Since SuperExpert is a rule induction shell, all its rules come

from the given examples. Therefore, completeness and accuracy of data input are most important factors for building up an expert system through a rule induction shell.

3). Reliability risk - The process of induction within SuperExpert is not made known to the users. Although this is not a problem in SuperExpert itself, the users have to take a risk by using the rule induction shell without a knowledge of the process of induction. A risk which is particularly apparent for diagnostic systems, because there are more or less exceptions which need specific rules, however, these exceptions would be disregarded by a rule induction shell under the low frequency of occurrence. This will result in danger in a diagnostic system, such as medical treatment.

Having examined four packages in detail, in the following section the author discusses another four packages briefly:

ES/P Advisor (Produced by Expert Systems International Ltd.)

ES/P Advisor is a small expert system shell for the development of an advice-giving expert system. It is suited to applications involving complex rules, regulations, standards or procedures, since the main feature of ES/P Advisor is its technique of converting the recorded text data into a knowledge base in the form of regulations or instructions - this technique is called text animation.

A fact that has been emphasised about ES/P Advisor is that it is not a general purpose expert system shell, indeed it is only a

simple package. This package is designed to excel in a specific type of application, i.e. to use text animation to exploit the potential of computer applications for advice given, such as:

- how to start a car;
- statutory responsibilities for employees' holiday leave;

Expert-Ease (Produced by Intelligent Terminals Ltd.)
Similar to SuperExpert, Expert-Ease is a rule-induction package which induces rules from examples of expert decisions and generates enquiry systems for reaching conclusions.

The major feature of Expert-Ease is its general purpose character, the shell can be used in a large number of areas, by using examples input into a spreadsheet format. This eliminates the need for programming knowledge by either the expert or the user.

Micro Expert (Produced by ISI Ltd.)
Micro Expert is a general purpose expert system shell containing a simple knowledge representation language and a runtime system. The maximum number of rules allowed to be input into Micro Expert is 500, thus this package is only suitable for small models.

This product was launched in 1980by ICI. Initially it was priced at £2,500 but is now at £300. ICI was one of the early customers

of Micro Expert, and made comments on this product. This led ISI to design another product - Savior in co-operation with ICI.

MicroSYNICS (Produced by Professor E. Edmunds at Leicester Polytechnic)

MicroSYNICS allows an expert or a knowledge engineer to create a user-friendly dialogue. According to the user's response, MicroSYNICS can provide information, prompt the user for further response and decide the next stage of the dialogue. The system has two basic components: the network compiler and a network interpreter. The former creates a dialogue file and checks the syntax of the relations between all the attributes; The latter runs the dialogue and controls the user interface.

Similar to ES/P Advisor, MicroSYNICS can be used to develop an expert system where a large amount of explanatory text is given to allow the user to make a decision. The system makes no decision itself but just prompts the user for responses in its network. This package is useful as a simplifier for the more complex traditional programs.

4.2.3 Why Xi Plus Was Chosen for Developing the Expert System?

According to the evaluation of expert systems shells carried out in this chapter, Xi Plus has its advantages as well as its disadvantages of use. From the above discussion of expert system shells survey, it can be found that although there were more disadvantages listed for Xi Plus than for Crystal or Expert Edge

in section 4.2.2, this is not because that Xi Plus is worse than the others but because that the author is more familiar with Xi Plus than with Crystal or Expert Edge. The following considerations constituted the reasons for choosing Xi Plus as the development tool for developing an expert system for selecting an appropriate financial planning package:

1). In this project, the author's supervisor performed the role of the expert, his knowledge was more easily to be expressed as IF....THEN type with forward chaining than backward chaining. Both Crystal and Expert Edge are backward chaining packages, thus using them would cause some inconvenience in building up the knowledge base whilst Xi Plus can handle both forward and backward chaining.

2). Although Xi Plus is incapable of dealing with uncertainty, the knowledge necessary for the project was clearly defined so that no uncertainty reasoning was required.

3). As for the NCC Starter Pack which contains four packages, although it was available in the university, it belongs to the computer centre of the university and the pack had been in great demand so that it was difficult to reserve the pack for building an expert system.

Based on the above considerations, despite the problem of speed, Xi Plus was selected for the project development.

4.3 Structuring the Problem

4.3.1 The Expert System Domain

As mentioned in Chapter 2, the initial stage in the development of an expert system is to define the domain of the problem. Financial planning packages can be used in two different but overlapping areas, namely: management information and decision support.

In Chapter 1, both MIS and DSS were discussed, these two types of system have little in common and they are developed for different purposes. Whilst it may be seen that both types cover financial planning applications . There is no need to separate the financial planning packages into 2 categories, because it is significant that "the development of decision support systems will not be successful unless the management information systems are firmly established" (67). Therefore, in this system the term 'financial planning packages' includes both MIS and DSS applications.

As mentioned before, the author's supervisor acted as the expert for the project, and he was competent to be an expert to contribute his knowledge in the area of how to select an appropriate financial planning package in organisation.
Having acted as the expert during the development of the system, the author's supervisor provided his knowledge about classes of financial planning packages that can be described as follows:

Classes of Financial Planning Packages

Five broad classes of financial planning packages have been described in detail by Finlay (84) and Finlay and Servant(85). In this project four broad classes are defined together with sub-divisions of two classes: thus there are six classes in all. The following describes the major characteristics of the four broad classes, which are dedicated, spreadsheet, advanced language and database.

Dedicated Packages

The word 'dedicated' is regarded as something designed to totally fulfill one particular purpose. Dedicated packages are different from all other types of package. They represent a considerable asset to those who have little time or computer experience or willingness to construct a well-defined task, for which they require computer assistance. The major difference between dedicated packages and others is that they do not offer a framework within which the user can specify his own logical model. They provide a set of completely predefined routines covering whole areas of accounting and finance, such as budgeting and capital investment analysis. The user does not need to be concerned about the data model structure and report specifications because these have been predefined. All he has to do is to insert the necessary data in response to the computer prompt. Considering an example of tax payable, all the user has to do is to choose the optional answers from the computer's

prompt, dedicated packages can give the standard report accordingly. Planalyst is a good example of a dedicated package.

Spreadsheet Packages

The major difference between spreadsheet packages and others is that the input is 'cell-specific'. A spreadsheet is considered as representing a large-sized electronic paper with rows and columns, data insertion is carried out by keying in a cell of data into each row or column.

Spreadsheets are simple and easy to use and this type of package is ideally suited for activities that perform calculations frequently and need little modification. e.g. depreciation, employee salaries, and simple budgeting. Lotus 1-2-3 and SuperCalc are examples of 2-dimensional spreadsheet packages, Report Manager is an example of a 3-dimensional package.

Advanced Language Packages

These packages offer a high-level language in which to write financial models in addition to the framework of spreadsheets. The fact that the logic and data are kept separate from one another gives rise to the other term for this type of package as 'separate logic packages'. Unlike the spreadsheet packages, these packages allow the same logic to apply to several sets of data. Mastermodeller and FCS are examples of 2 Dimensional advanced language packages.

Database Financial Planning Packages

These packages are developed along database lines particularly for financial modelling, but they should not be confused with database packages themselves (such as dBase III).

The main feature of database financial planning packages is that they provide considerable freedom for the construction of models. Here the author would like to regard this feature as a similar to the 'frame-base' knowledge representation discussed in Chapter 2. The link between the form of input and that of output allows the user to specify simply the basic features of his model, such as the elements, the relations between the variables and the input data. These features (in terms of a frame-base representation, can be regarded as 'chunks') will then be manipulated and displayed by the package. These packages are useful to corporate users with large volumes of data and complex data structures. PCExpress and Demon are examples of database financial planning packages.

The developed system's aimed to decide the ideal class of financial planning package according to organisation's particular conditions. The ideal class of financial planning packages is the type of package which is most appropriate to an organisation with no constraints.

Considering the goal of the system is to decide the ideal class of financial planning package (first level of the hierarchy of

knowledge), three major factors cover both data and logic aspects and the relevant requirements within the system. These factors can be regarded as the second level of the hierarchy of the knowledge. The third level of the knowledge hierarchy comprises those variables which decide the above three factors. For example, variables which affect data model structure include the builder's willingness to learn, the dimensions (e.g. time, amount,etc.) a system is to be catered for, the variables which are to be considered in the system (e.g. production volume, cost of sales, etc.) and the aggregation level(s) constituted by those inputs of data. Structure of the knowledge is diagrammed as figure 4.2, 4.3, and 4.4.

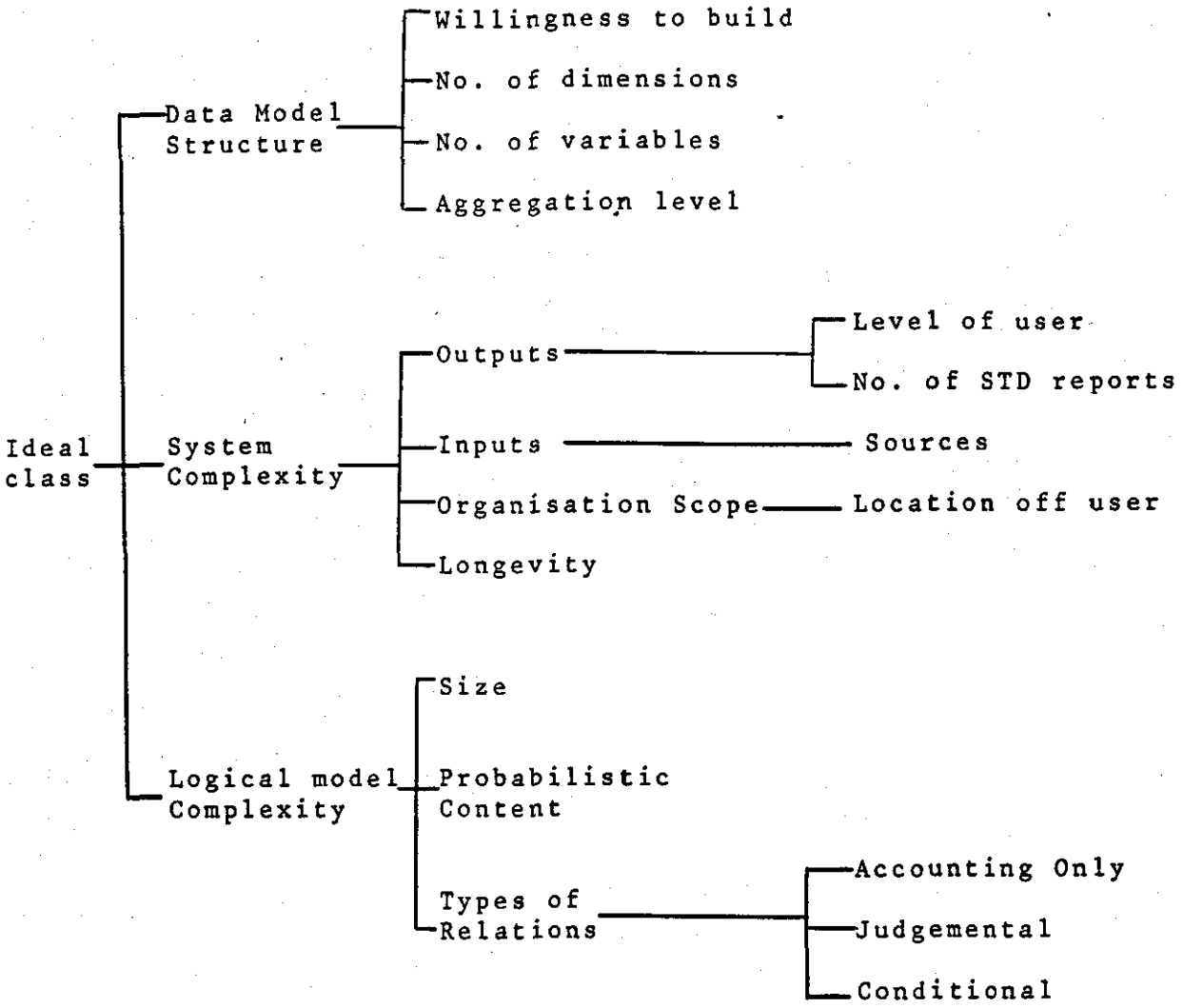


Figure 4.2 Structure of the expert system for selecting a financial planning package(67)

Since it was agreed that the main purpose for which the author is constructing the system are to experience building an application, investigate the features of a number of expert system shells, and gain familiarity in using expert systems, etc. not simply to design an expert system. Therefore, although there should be another three different levels of logical model complexity - high, medium and low, the developed system lacks the identifying rules for this variable. There are only two levels given to this variable - high and not high.

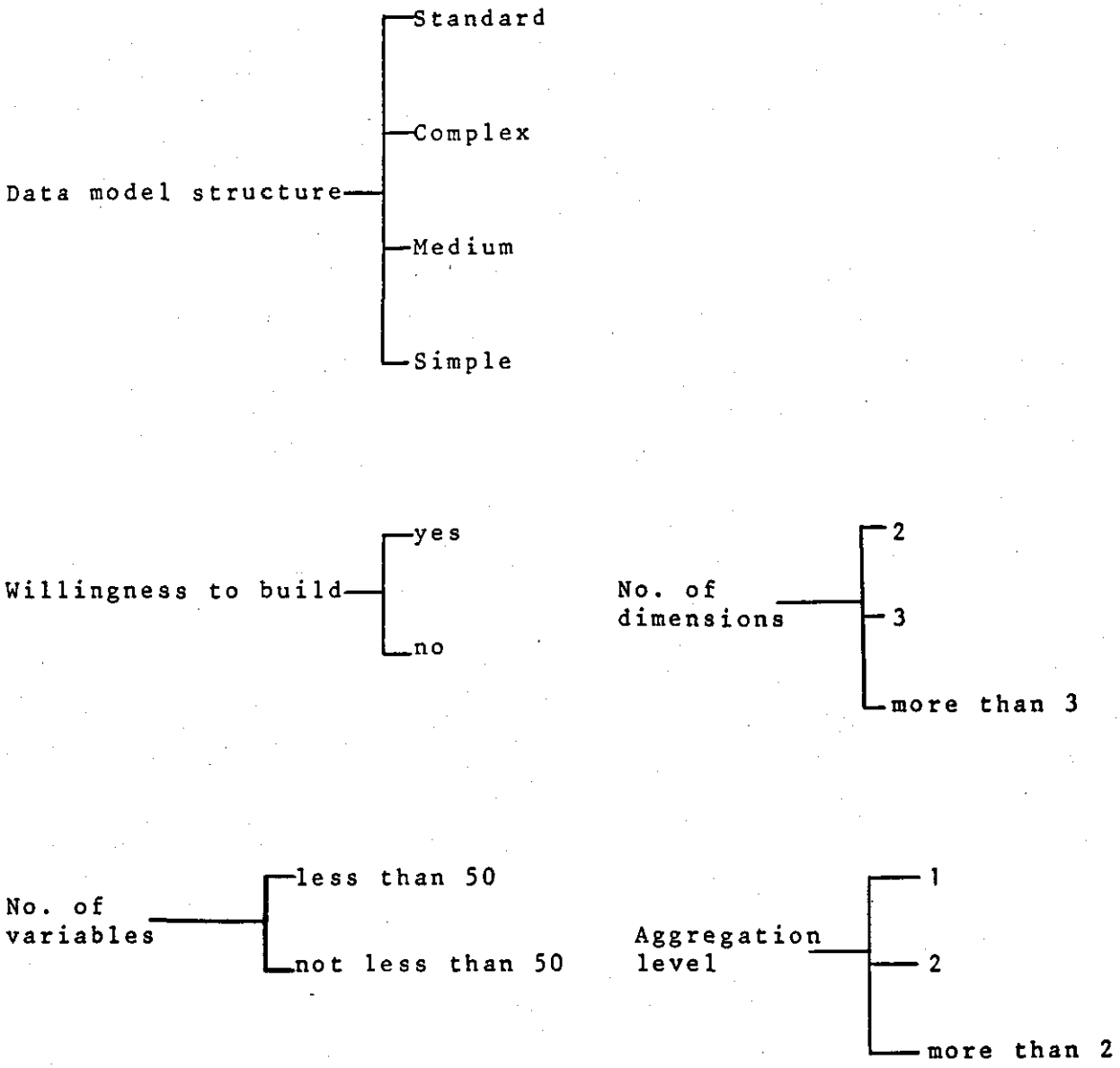


Figure 4.3 States of variables decide the Data Model Structure (67)

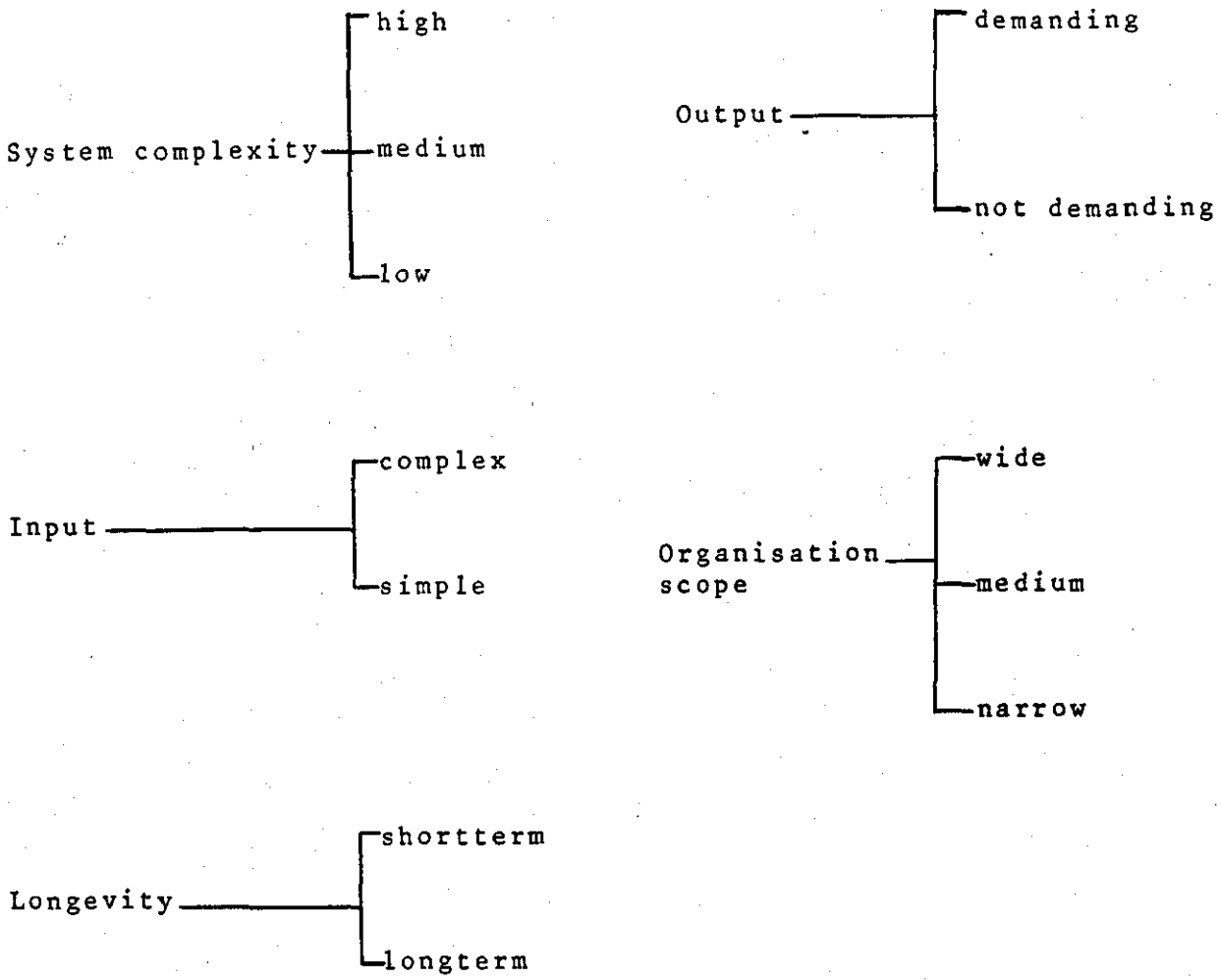


Figure 4.4 States of variables decide the System Complexity(67)

By repeated interviews conducted between the author and her supervisor, the above knowledge structure for the problem domain was finally decided, the author then converted them in an inferable form - the rules of IF....THEN.... type by logical rules sequence.

4.3.2. The System Rules

The expert system rules were established in the form of production rules, that is

If A is B

then X is Y

Applying this formula to the project, the rules were entered into Xi Plus knowledge base in a form such as:

If data model structure is complex
and system complexity is high
and logical model complexity is high
then class is database

The full set of rules for this system is attached in Appendix C.

4.4 Review of the Approaches Applied in the Project

In Chapter 2, a proposal for creating an expert system application was given, and different methods at various stages were discussed. This section reviews whether the approaches applied in developing the project correspond to the methods provided.

a). Selection of a clearly defined domain and well boundaried knowledge:

The purpose and subject of the proposal were clearly defined as selecting an appropriate financial planning software package, the hierarchy of knowledge was decided when the project was started.

b). Little mathematical knowledge is required:

In this project, no numeral calculation or mathematical techniques are required. All the rules in the system are expressed in non-mathematical form.

c). The expert can clearly express his knowledge:

During interviews, the expert (the author's supervisor) was able to give clear answers to the questions raised by the author for eliciting his knowledge.

d). There is the real expert:

The expert, the author's supervisor, is one of the authors of the book 'Financial Planning Package' (85) which provides a practical guideline to the selection of financial planning packages for use within organisations.

e). The expert supports the project:

In this project, the expert did offer full support, because he was the initiator in developing the working expert system to accompany the publication of the book mentioned above.

f). The task is not too difficult to be understood by naive users or the knowledge engineer:

With friendly supporting text and necessary examples given to prompt the user's response on a query, this problem is not too difficult to be understood by a naive user. Also, the author was an Accounting supervisor of an American company in Taiwan before she came to England for advanced study. With a background of financing and planning, the author had no difficulty to understand the project.

g). The task is not too difficult to develop:

Both the supervisor and the author have backgrounds in MIS, so that with co-operation the task presents no technical difficulties.

h). As for the testing, the system has been tested many times since its first development. The testing was conducted in a practical way, i.e. the supervisor checked the match of the rules represented by the author with his own knowledge in his head; the author checked the represented knowledge from the logical points under operation to ensure that no conflicting values would be given by system. After both sides were satisfied with their checking, this system was demonstrated to the supervisor - the expert.

4.5 Further Work

When the ideal class of financial planning package was successfully decided by the system, the author anticipated that companies should be allowed to obtain advice on the best financial planning package within the ideal class of financial planning packages induced by the system by entering their

individual constraints. The constraints can be either the cost of the package or the hardware facilities.

To decide the financial planning software which is most suited to a company's particular requirement, a database of relevant data for all suitable financial planning packages was required. To obtain these data, questionnaires were sent to 50 producers of financial planning packages. The information needed by the questionnaire include the price of the package, the memory required for running the package, the operating system used by the package, etc. Details of the questionnaire is attached as Appendix B of this thesis.

It was found that Xi Plus was not able to support a large database. Therefore, in order to achieve the goal of developing an expert system which is able to give advice based on the information of a database, the linking between Xi Plus and the database was considered.

Having obtained a response from the producers, the author keyed in all the relevant data of a financial planning package into dBase III Plus. The reason for choosing dBase III Plus for setting up the database was that dBase III Plus is a well known database package, and this package was available in the student's own university department.

However, it was finally realised by the author that the direct interfacing between Xi Plus and dBase III Plus was not possible, although the ability of interfacing external programs with Xi Plus is described in the user manual. In fact, the problem of interfacing Xi Plus with dBase III Plus had not been properly addressed until June 1987(86). Whilst the author tried her best to follow this instruction, it was not possible to achieve the interface.

Having failed to link Xi Plus directly with dBase III Plus, the author then considered using a rule induction shell as a substitution for the linking. SuperExpert, a rule induction shell was available in the department. The author input into SuperExpert those factors which affect a company's choice of a financial planning package with details of those financial planning packages obtained from the producers. They were: the names of packages, the prices of packages, the memory required for running packages and the operating systems used by packages. An example listing using SuperExpert for this interfacing is illustrated in figure 4.5.

	integer memory	integer price	logical oprsyst	logical class	logical decision
1	*	1000	*	dedicated	planalyst
2	2000	13000	other	ddadlang	express
3	640	1500	pc-dos	ddadlang	pcexpress
4	640	1500	pc-dos	database	pcexpress
5	640	1500	pc-dos	resources	pcexpress
6	384	975	ms-dos	ddadlang	mstrmodlr
7	384	975	pc-dos	ddadlang	mstrmodlr
8	384	975	aix	ddadlang	mstrmodlr
9	384	975	currentcpm	ddadlang	mstrmodlr
10	512	6000	ms-dos	ddsprdsht	twenty
11	512	6000	pc-dos	ddsprdsht	twenty
12	512	6000	unix	ddsprdsht	twenty
13	512	1200	ms-dos	ddadlang	demon
14	512	1200	pc-dos	ddadlang	demon
15	512	1200	unix	ddadlang	demon
16	512	1200	cpm	ddadlang	demon
17	512	1200	currentcpm	ddadlang	demon
18	512	1200	ms-dos	database	demon
19	512	1200	pc-dos	database	demon
20	512	1200	unix	database	demon
21	512	1200	cpm	database	demon
22	512	1200	currentcpm	database	demon
23	512	395	other	ddsprdsht	oxcalc
24	256	298	ms-dos	ddsprdsht	bottomline
25	256	298	pc-doc	ddsprdsht	bottomline
26	640	1000	ms-dos	dddsprdsht	reportmgr

Figure 4.5 Example screen of SuperExpert for selecting a financial planning package

This example was given by using the producers' feedback on the details of each package. Looking at the third example, it means that Express is a 2 dimensional advanced language package, using

PC-DOS operating system. Its price is £1,500 with requirement of a 640K computer memory.

After the example data are set up, rules are induced by SuperExpert itself. The induced rules for the above examples given by SuperExpert can be found below:

```
SuperExpert: Rule listing
problem: /decision

class
    dedicated : planalyst
    ddsprdsht : price
          <  3198 : memory
               <  384 : bottomline
               >= 384 : oxcalc
          >= 3198 : twenty
    dddsprdsht : reportmgr
    ddadlang : memory
          <  448 : mstrmodlr
          >= 448 : memory
               <  576 : demon
               >= 576 : memory
                    <  1320 : pcexpress
                    >= 1320 : express
    database : memory
          <  576 : demon
          >= 576 : pcexpress
    resources : pcexpress
```

Figure 4.6 Rules induced by SuperExpert for selecting a financial planning package

Looking at the first rule, it can be read as:

If class is dedicated
then decision is Planalyst

(Note: Since there is only one dedicated package in the data, i.e. Planalyst, the system reaches the conclusion without giving alternatives.)

The second rule which is much lengthier than the first one can be read as:

If ideal class is 2 dimensional spreadsheet
and maximum cost < 3198
and hardware memory < 384
then decision is Bottomline V

Having decided the ideal class of financial planning package which is appropriate to a company's situation, the system would then carry on asking the user about the maximum cost the company could afford for the purchase and the memory size of the existing computer the company has. According to the user's answers, the system would then reach the conclusion on the name of the financial planning package which is most suitable to the company's individual conditions. Having been advised by the system on the most suitable package, the user would be also

allowed to read all the details of any package which was recorded in the database file of dBase III Plus.

The main purpose for the above trial was that the author attempted to find a substitution for the link between Xi Plus and dBase III Plus, so as to extend the system's function to a better position of being able to decide the 'best' financial planning package, not just the ideal class of financial planning package.

However, it was finally realised that the above attempt did not successfully achieve the author's goal, because of the following findings:

1). Incomprehensive rule induction - Although SuperExpert can induce rules from a large quantity of data, it can not induce rules covering all the necessary factors for making a decision. For example, the fourth rule induced by SuperExpert in figure 4.6,the rule can be read as :

If ideal class is 2 dimensional advanced language
and hardware memory is less than 448K
then the decision is MasterModeller

In this rule, both price and operating system are not considered by SuperExpert.

Another example can be found from the rule which concludes the decision for Express. Here Superexpert suggests that if the hardware memory size is larger than 1320K then the 'best'

financial planning package recommended is Express. Referring back to the example listing given in figure 4.5, the memory size needed for running Express is 2000K, also the price for Express is I13,000. For a company which has computer with memory size available between 1320K and 2000K, or which can not afford such high amount of I13,000 spending, this rule is unhelpful.

2). Illogical rule induction - It was discovered that SuperExpert induces rules from a mass of numeral data by simply averaging them. For example, the fourth rule induced by SuperExpert in Figure 4.5, the number of 1320K was the average of both 2000K for Express and 640K for PcExpress. SuperExpert has no facility to give recommendation with upper and lower limit for numeral conditions.

Based on the above two findings, the author would say that her attempt to develop an expert system application with capability of recommending the 'most appropriate' financial planning package for a company failed by using SuperExpert as a substitution for the interfacing between Xi Plus and dBase III Plus. Had the author had enough time for her research, she would like to try a frame-base development tool, such as Leonardo, for her project rather than using a rule-base shell, such as Xi Plus which is not capable of storing a large quantity of data so as to induce rule from its own storage of these data.

The lesson which can be learned from this failure is the disadvantages of using a rule induction development tool for inducing rules from a large quantity of data. The author expects the further prospect of her project to be re-developed by a frame-based shell that the new system would be able to perform this task.

4.6 Conclusion

Although the developed expert system does not work as perfectly as the author hoped, the project for developing an expert system which is capable of deciding the most appropriate class of financial planning package according to organisations' particular requirement has been completed. However, with the failure of using a rule induction development tool as the substitution for linking a rule-based expert system shell and a database package, there is further work pending for improving the existing system:

a). This system needs further identification of the elements that decide selection of an ideal class of financial planning package, such as the logical model complexity and longevity.

b). Few completed questionnaires have been returned. Up to the time of writing, only seven responses have been received. The system is for research and trial purposes only, it cannot be recognised as a real expert system because it uses only seven database records out of fifty questionnaires.

c). As stated, the author expects the system to be re-developed using a more powerful development tool, such as a frame-based

shell. Therefore, a further survey of expert system shells so as to decide the appropriate tool would be required.

The system is workable and has been constructed using 63 rules with 2 demons. Being induced by SuperExpert, the final part of the knowledge base (rule 52-63) of the system is not correct. Those rules included in this part of the knowledge base attached in the Appendix C are corrected by handwriting so that the readers can have an idea of the extent of errors made by a rule induction expert system shell.

CHAPTER 5 - FEATURES OF CURRENT EXPERT SYSTEMS AND FUTURE TRENDS OF EXPERT SYSTEMS DEVELOPMENT

In the UK interest in expert systems is increasing, and more and more companies are looking at the possibility of applications of this new technology (50). This chapter is concerned with an overview of the features of current expert systems, the current status of business applications of expert systems in the UK, the future trends in expert systems development and the future research directions for expert systems.

5.1 Summary of the Features of Current Expert Systems

5.1.1 Features of Current Expert Systems

Having had no opportunity to practically experience any expert system available on the market, the author could only give her views on the current status of expert systems in the UK of business applications based on her visits to six companies and her review of the literature. These views will be discussed in section 5.2. As for the discussion on the features of current expert systems, the author believes that it would be more suitable to review this subject based on the author's literature survey.

Taylor (88) summarises the features of current expert systems. All of the points made regard items that have been emphasised by

AI researchers. From the description given below, it can be seen that most of the expected successes have not yet been achieved.

Taylor regards the present expert systems as 'classifiers', because they obtain data from their users about particular problems (usually by using natural language), and carry out the classification or categorisation of the data according to the stored knowledge and applying their inference procedures: the systems then present the results to their users.

Most expert systems tools surport to provide an 'explanation' facility. This facility is for answering the user's query of 'how' or 'why' the conclusion has been reached. At present, most of the 'explanation' facilities provide the ability to trace the rules used in the reasoning process, they do not actually provide any extra assistance to the user for the purpose of understanding the reasoning.

Current expert systems are at most assisting human experts with routine tasks, and are not in a position to replace them. There are still areas with which current expert systems cannot deal.

Although user-friendliness is also emphasised as a necessary feature in expert systems technology, current expert systems are not really friendly, much less intelligible, to the user.

Finally, current expert systems are expensive in relation to the functions and advantages they provide.

5.1.2 Limitations of Current Expert Systems

Muller (89) lists the deficiencies of current expert systems as follows:

- Narrow band of knowledge, with a lack of knowledge of the limitations of the systems, only one method of problem-solving can generally be offered;
- Little or no cross-checking, the systems are unable to cope with multiple situations;
- Mostly difficult to construct and to maintain;
- Insufficiently expressive languages;
- Manual, slow knowledge acquisition processes, with tedious incremental modifications;
- Each system starts from the basics i.e. no knowledge libraries or modules of knowledge on which new systems can build;
- Special hardware and special training are required for large expert systems (Noted by Muller: these limitations begin to disappear for 'small' or 'minor' expert systems);
- Domain experts/specialists are not always available;
- Poor explanation facilities which, typically, simply repeat the rules in the knowledge base;
- Incapable of learning, enabling programs to learn from experience;
- Incapability of model building, programs which are able to weigh alternatives and construct new beliefs.

The above summary and limitations of current expert systems lead to a further discussion of recommended research directions for expert systems in the final part of this chapter.

5.2 Current Status of Expert Systems in UK Business

5.2.1 General Scene

The Ovum's report (50) indicates that the commercial development of expert systems in Europe is about 12 to 18 months behind that of the United States. The gap is much wider in the finance, insurance and defence sectors. On the other hand, the European oil companies, such as Shell, Elf Aquitaine and British Petroleum, are probably more advanced in their use of expert systems than the American oil companies.

As stated, in the UK, interest in expert systems is increasing (50), but according to the author's literature survey and her visits to companies, use of expert systems in business is still in its infancy. Small-scale and trial systems in particular are more popular than large-scale and fully operational systems. As Jones (90) points out, "the UK businesses are the biggest market in Europe, largely through the adoption of smaller micro-based expert systems". Jones' opinion is supported by the Ovum report, where it is indicated that "a unique feature in the UK's expert systems development is the extensive development of small-scale systems, particularly on personal computers, which has encouraged a very large number of

companies - both large and small - to experiment with expert systems". The following are considered as the main reasons for this:

- the widespread awareness of expert systems in the UK, even within small companies

  a rather unambitious attitude towards expert systems development that favours the use of low cost development tools
- an influential report published by Alvey (91) which points to small-scale systems as the area of greatest potential benefit for expert systems.

In addition to the above reasons, the author thinks that the following two reasons may also be responsible for the unique situation in the UK:

- expectation of low cost funding leads to companies looking at developing small- scale systems by using low cost development tools, e.g. shells.
- non-participation of the computing/MIS departments in companies leads to developers using PC's for developing their trial systems because of the lack of technical support.

Although the interest of expert systems is widespread in the UK, the major users of these systems lie in those large companies, such as British Petroleum, ICI and British Telecom. Many of the small companies are using expert systems as a means of gaining experience of the new technology; thus a large proportion of these applications are either for training purposes for their

staff or are still under trial. This can be found from the author's interviews with companies, there all of the interviewees agreed with it. Also, from the analysis of visits given in Chapter 3, it can be found that a large proportion of these companies have not put their expert systems into daily business operation.

In October 1987, Alvey (92) published a second short survey report on expert systems in the UK business world, the findings of which are summarised below:

a). Most large organisations have project teams, but experience of operational applications is still limited.

b). The giant corporations have an average 10 operational applications.

c). The second largest organisations have 2 - 5 operational applications whilst many well-known organisations have not yet delivered a system into business usage.

d). The pace of development is nevertheless accelerating and the number of operational applications should rise sharply within a year.

e). Constraints and limitations such as lack of management commitment, business secrecy, poor organisation, fear of the nature and the cost of new technology are ever present.

f). Quoted by the Alvey report, Johnson (49) of Ovum observes that "work stations and their associated American software have made very little impact on British business

applications which is a market dominated by British personal computer shells".

5.2.2 Technical Overview of UK Business Applications

As stated in Chapter 3,where the problems encountered when choosing companies were discussed, the six companies were considered to be representative when reviewing the status of expert system applications in UK business. Based on these findings and coupled with a literature review, the author gives her views on the current status of expert system applications in UK business.

This section reviews UK business applications from the technical viewpoint. The general features of this technical viewpoint have been discussed in Chapter 2, under the leadings of domain choice, knowledge acquisition, knowledge representation, reasoning, development tools, explanation and validation.

Choosing the Domain

Since the largest part of UK business applications of expert systems are small-scale systems, their domains are consequently narrow: examples are production process fault diagnosis, bank lending etc. These narrow domains are suitable for a company's experimental development and their quick successes suit the company's wishing for a quick return on relatively small outlays.

Knowledge Acquisition

Of the four methods recommended by Hart (41), the interview and blackboard technique are the most common methods. In practical business applications, there is a certain degree of role merging between the knowledge engineer, the expert and the user. For example, the developer may not only be the expert but also the user.

Knowledge Representation

Rule-based representation is much more common than both semantic network and frame-based representations. The IF....THEN rules are more acceptable to business applications than the other types of knowledge representation.

Reasoning

Although the facility for uncertainty reasoning is emphasised as a requisite by AI researchers, many of the developers involved in the practical UK business applications disregard it, many applications even attempt to avoid uncertainty in answers that may be given by users. This situation was encountered in the company visits as described in Chapter 3. These systems question their users in a way to elicit definite answers, either single or multiple answers with certainty.

Explanation

This facility is not always required by all of the developers especially those shells users. Being restricted by the shell's

capability of giving satisfactory explanation, the developers do not required a more friendly explanation.

Tools

Among the three major kinds of tools for building expert systems, namely programming languages, toolkits and shells, the shell is the most popular tool adopted by UK businesses for building applications.

In the UK many shells are developed that are aimed specifically to be run on the IBM PC and compatibles. Xi Plus, Crystal, and Expert Edge are three examples. It is believed that Xi Plus is leading the European market, and Crystal is the strongest challenger (50).

As stated, Prolog is more popularly used in the UK than Lisp for self-programing an expert system. Table 5.1 is the list of commercial versions of Prolog available in Europe (50). Disregarding the USA versions, there are 23 European versions. Of these 23 versions, 10 are of UK origin. This figure indicates how popular the Prolog language is in the UK.

Name	Supplier	Location	Hardware
Arity Prolog	Arity Corp	USA	IBM PC
BIM Prolog	Belgian Institute of Management	Belgium	Sun
D-Prolog	Delphia	France	Bull & Others
Edinburgh Prolog	AIAI, Edinburgh Uni.	Scotland	Various
ICL-Prolog	ICL	England	ICL 2900
IF/Prolog	InterFace Computer GmbH	Germany	Unix machines
MI4	Electronique Serge Dassault	France	IBM VM environments
MProlog	Epsilon	Germany	Various
MacProlog	Logic Programming Associates	England	Macintosh
Maialog	Amaia	France	M1000
MicroProlog	Logic Programming Associates	England	MS-DOS, CP/M 86
PCE	SCS/Uni. of Hamburg	Germany	Sun
Prolab	GMD	Germany	Unix machines
Prolog II	PrologIA	France	Various
Prolog Profrssional	Logic Programming Associates	England	IBM PC
Prolog-1	Expert Systems International Ltd	England	Various
Prolog-2	Expert Systems International Ltd	England	IBM PC
Prolog/P	CRIL	France	Various
Quintus Prolog	Quintus Computer Systems Inc	USA	VAX, Xerox 1100 Series Apollo, Sun
SD-Prolog	Systems Designers	England	IBM PC

Sigma Prolog	Logic Programming Associates	England	Unix machine
TOP-ONE	Telecomputing	England	ICL & IBM mainframes
Turbo Prolog	Borland International Inc	USA	IBM PC
VM/Prolog	IBM	USA	IBM mainframes running VM
V Prolog	Amaia	France	Amaia PS-C
Xilog	Bull	France	Bull Micral 30 & 60
Zyx Prolog	Zyx	Sweden	HP, Macintosh

Table 5.1 List of commercial versions of Prolog in Europe (50)

As for the toolkit, it has been stated in Chapter 2 that in the UK this type of higher-level development tool is not so widely used as in the USA. There have not been many packaged expert systems sold on the market up to the time of writing. One example is the Employment Law expert system jointly developed by Expertech and Robson Rhodes Chartered Accountants using Xi Plus. This package provides advice on employee legislation to clarify dismissal and maternity rights. Another example is the Letter of Credit Advisor jointly developed by Helix Expert Systems and the Bank of America using Expert Edge. This package provides advice on preparing letters of credit for claiming payment from banks. These applications can release companies from in-house programming that requires input contributed by experts and knowledge engineers.

Validation

To most of the developers the method used for validation is simple, and most validation is aimed simply at making the system workable. Although many methods of validation have been discussed in the literature, these methods are not actually applied in business.

5.3 Future Trends of Expert Systems Development

From the previous sections, it can be seen that the first step to the commercialisation of expert systems has been achieved. What will be the way forward in the next five - ten years?

Although there have been many discussions in the literature concerning the future of expert systems, most of these discussions are concerned with the general future research directions and take on optimistic point of view. For example, Gevarter (29) forecasts that there will be few domain or functional limitations in the ultimate use of expert systems, thus expert systems will eventually find use in most endeavors which require symbolic reasoning with detailed professional knowledge - which includes much of the world's work. Therefore, Gevarter expects that 'intelligent, friendly and robust human interfaces and much better system building tools' are to be seen by the late 80's. Moreover, he anticipates that:

> Somewhere around the year 2000, we can expect to see the beginnings of systems which semi-autonomously develop knowledge bases from text. The result of these developments may very well herald a maturing information society where expert systems put experts at everyone's disposal. In the process, production and information costs should greatly diminish, opening up major new opportunities for societal betterment.

Harmon and King (30) also give an overview of the possible applications of expert systems in businesses by focussing on common functions. Table 5.1 is adapted from their book. This table shows the possible domains to which expert systems can be

applied in the next few years and the objectives to be achieved in each domain application.

Domain	Problems
Senior management: Executive officers Senior managers Strategic planners Senior Staff Consultants	Need to reduce organisational complexity Need to monitor an increasing volume of information Need to access experts and consultants for specific advice
Operations: Manufacturing services complex equipment operation Energy exploration Quality control Inventory control	Need to improve coordination of organisation, scheduling, and management Need for overview of complex systems for rapid decision making Need to monitor/control complex equipment
Support services: Public relations Legal Personnel & training DP service Building & maintenance Research & development	Need to train/retrain lots of people to handle complex jobs Need to communicate new, complex procedures Need to examine/explain policy decisions/options Need to control/reduce costs of computer software development and maintenance
Finance: Portfolio managers Accounting Financial managers Auditing Controlling	Need for overview of complex existing system Need for smart, goal-directed financial planning tools
Marketing: Sale Advertising Marketing research Customer service Ordering	Need for expert assistance in examining marketing questions Need for sales assistance programs that provide product knowledge and help configure proposals/packages
Office automation: Word Processing Data Management	Need to increase productivity in handling, filing, communicating, retrieving, and distributing information in offices

Professional services:	Need to monitor an increasing volume of information
Management consulting	Need to access other experts and consultants for advice
Lawyers	Need to submit "smart reports"
Physicians	Need to prototype & simulate knowledge systems rapidly without having to know about traditional computing
Accountants	

Table 5.2 Overview of knowledge problems common to most companies or professions (30)

Another positive future of expert systems is forecasted by Weiss and Kulikowski (93). They indicate that expert systems will be considered as smart programs and will perform with many of the attributes that we take to be expert when they are present in a human.

Contrary to the above representative optimistic forecast, the author questions the bright future of expert systems along the following lines:

a). Senior managers will not use expert systems in their work: Presently, although there are much Decision support System software available on the market for managers' use, most of these in use are spreadsheets (94). The main reasons of this fact are that for those senior managers they do not have to learn how to 'convert' their knowledge. This is convenient to senior managers especially they do not have to spent time on programming their knowledge. This situation can be used when considering the future of expert systems development. For effectively performing managers' task on decision making, expert systems must be developed through the deep involvement of managers. However, managers are unlikely to have enough time to spend on this development, or have the willingness to learn how to convert their knowledge into production rules. This situation will not be changed unless the method of knowledge representation of existing expert systems can be much improved, such as by using a

spreadshee style for the representation. There is unlikely to be much change in this situation over the next five to ten years.

b). No possibility for catching knowledge of senior managers without their involvement:

It has been stated in Chapter 1 that the evolution of expert systems has distinct relationship with traditional computing systems. Although the development of IT can be theoretically reviewed as some major types - DP, MIS, and DSS, a very large number of the current computing applications are still at the most basic requirement of computing using - data processing. These systems are developed by either the professionals with a computing background or by the managers' subordinates. This means that most of the existing computer systems do not disturb managers in organisations for their development, especially the senior level managers (95). This situation will also obstruct the future development of expert systems, because, to those developers other than managers themselves, it is not possible to completely catch managers' knowledge without managerial involvement.

c). Human factors obstruct the development of expert systems:

In general, experts do not like to share their expertise with a machine or any other non-expert (such as an operator or a clerk). Also, people may not totally trust the decision made by a machine. For example, a seriously illed patient is unlikely to trust the suggested treatment made by

a computer system; the passengers may fear of the emergency situation by taking a train controlled by a computer system without any human driver's watch.

d). Long period of time taken for developing a really friendly, useful and helpful expert system:

Useful, convenient word processing software had not become popular until the late 70's. It took nearly 2 decades for the software to be widely used. This might because of the price decrease of computers from the mainframe or minicomputer to PC, or because people had high quality typewriters, but the quality of work produced by word processing has been much improved since the early 80's. Applying this concept to the development of future expert systems, in the next five to ten years time, the really friendly, intelligent expert systems will not be produced.

e). Difficulties on gaining comprehensive knowledge for those areas common to business with unstructured knowledge:

The software packages for traditional DP purposes were developed under the condition that all the knowledge is well defined and structured. For example, Accounting is a popular area of computing application because it has reliable accounting principles and fully defined variables expressed numerically. On the other hand, for high-level problem-solving or decision-making, the knowledge of experts tend to be unstructured and non- numerical

and their knowledge includes many years' expertise, including the capability of dealing with emergencies. Experts may not know how to express their knowledge properly or completely and the knowledge engineer may not know how to elicit this knowledge. Unless the techniques of knowledge elicitation can be considerably improved, difficulty of covering comprehensive knowledge into an expert system for many common fields in business will be still exist. This situation is unlikely to may not be change in the next five to ten years.

The inability of some expert systems in dealing with uncertainty will restrict the completeness of knowledge representation and the accuracy of expert systems users giving their answers.

f). Interest in expert systems will fall off:

Although the number of people interested in expert systems is increasing and this trend is forecast to continue, this increased interest in expert systems may subsequently fall when results are not found to match expectation.

g). Expert systems will never perform task as well as humans:

As mentioned, the knowledge of experts tend to be unstructured which include the experts' many years' expertise. Also, experts can learn new knowledge from their performing tasks: existing expert systems still lack this the capability of learning.

Apart from the above 7 points, the author would like to give more discussion about the future trends of expert systems development based on her findings from the visits to six companies reviewed in section 3.3.2:

h). Shortage of resources:

In a similar way to the development of traditional computing systems, the pressing demand for professionals with a computing background has been evident for more than a decade and continues. In the next 5-10 years, professionals with an I.T background, especially with knowledge of expert systems, will be in great demand. However, the academic training from regular education will not be able to cope with this demand.

i). Shortage of support from computing professionals:

With the above shortage of professionals with I.T background, the development of expert systems will be obstructed if support is lacking from the computing professional. The developers, with little knowledge about computers, can only develop expert systems that are small scale and cover a simple domain by using personal computers with expert systems shells.

j). Deficiency in the support from high-level management:

Without the involvement or support from top management, the development of a new technology would be considered as 'not important' and the developers would not pay much attention to the development.

k). The experts are not able to express their knowledge clearly:

This situation will be even worse in developing large scale or complicated systems. Also, the current rule structure of IF....THEN type is an obstacle to the experts contributing their knowledge. This is not totally the experts' fault, but this situation is unlikely to change in the next decade.

l). Time taken to acquire & represent knowledge:

There is no recognised methodology for knowledge acquisition and representation. In the next decade knowledge acquisition will remain the main difficulty in the production of expert systems. However, this level of difficulty will depend on the cooperation between the expert and the knowledge engineer.

m). Too much reliance on the use of shells:

With the advantages of using shells and the rapid increase in the number of shells available on the market, the situation of using shell for developing expert systems will not change over the next decade. Moreover, it can even be forecasted that there will be more and more shell users/developers in the next decade.

Literature in which the negative future of expert systems is discussed can be found from Feigenbaum (96) and Drefus (97).

Applying the above negative points of view on the future of expert systems to Harmon and King's (30) consideration of the future of expert systems, they summarise five needs which they anticipate will drive the knowledge engineering market during the next five years. These five needs are:

. New approaches to business organisation and productivity:

The computer is playing an important role and is expected to play an even more significant role in the next five years for reconceptualising how businesses can be organised to increase their productivity.

. Expertise:

It takes 10 to 15 years for an individual to master a speciality. The really good experts are in short supply in almost all areas, and non-experts need to turn to them when problems occur. When problems become increasingly complex, experts are in greater demand.

. Knowledge:

Managers do not want to have to wade through volumes of statistics, they want information organised in a useful way so that they can learn the crucial components of the decision-making process and know how to deal with the situations affected by those components. This is the knowledge they require.

. Competence:

Businesses need competent employees who are able to explain their products and options to customers. However, products have become more numerous and more complex, and competent employees may be difficult to find. Yet a consistently competent service is required by business.

. Smart automated machines:

Businesses require the machines which share their offices to be programmed to function in more intelligent ways.

With the above five needs in mind, Harmon and King forecast a three-tiered market which reflects the relevance of the needs among these tiers.

Figure 5.1 shows the forecasted knowledge engineering market:

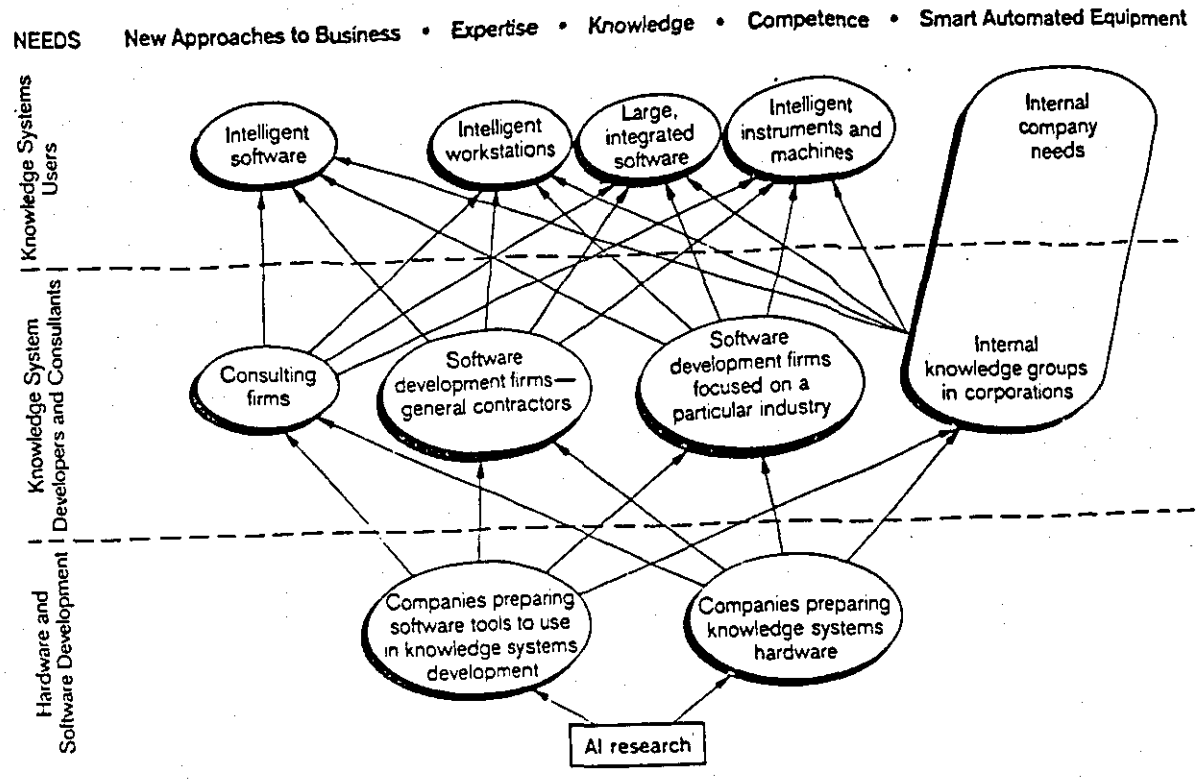


Figure 5.1 The forecasted knowledge engineering market in the next five years (30)

At the bottom tier, the market consists of the companies which are preparing to sell hardware and software for expert systems development.

The second tier will be the main part of expert systems suppliers, which includes consultancy and software firms. Customised products should be easily developed by the companies in this tier.

The third level comprises companies and individuals who will sell expert systems to consumers, probably incorporated in a product.

This tier is entirely devoted to practical applications of expert systems.

The developmental problems such as knowledge acquisition and knowledge representation will be obstacles in the second tier. In the third tier of the market, it will meet the problems of experts' using rules to represent knowledge with reluctance because spreadsheet style is more preferable. Also, lack of involvement by senior managers will result in an incomplete knowledge representation being produced by other people for representing senior managers' knowledge into expert systems.

5.4 Research directions of Expert Systems in the Next 5-10 years

Within the area of business applications of expert systems, the problems of future trends of expert systems have been discussed in the previous section. These problems indicate the directions for further research which can be stated as follows:

- investigating training schemes to produce professionals with an I.T background as well as the knowledge about expet systems,
- developing systems with more capability of capturing complicated knowledge and which will be suitable for more types of applications.
- developing more powerful knowledge acquisition tools so as to ease the task of knowledge acquisition.
- developing systems of greater intelligence which are capable of learning rules from experts' experience and then constructing the systems themselves.

- developing systems with greater friendliness and explanation facilities.
- convergence of expert systems and other technologies (89). For example, linkage of expert systems with database systems; linkage of development tools with conventional software packages- e.g. spreadsheets, databases and wordprocessors, integration with other fields- e.g. Decision support systems and Operations Research.
- formalisation of methods or a range of methods for building expert systems, so as to overcome the shortage of software development skills (89).

To assure the success of future expert systems in business, an awareness of the problems from which businesses are most likely to suffer must be maintained as an important direction for future research.

CHAPTER 6 CONCLUSION AND SUMMARY

Expert Systems have emerged from innovations within the field of artificial intelligence. They have distinct relationship with information technology which has played a prominent role in business during the past few decades. The benefits generated by the use of IT have led to an increasing desire to computerise human expertise and knowledge for problem-solving tasks.

It has been more than two decades since the first expert system - DENDRAL- was developed. From the characteristic differences that exist between conventional systems (such as DP and MIS) and expert systems, such as the natural language for knowledge representation and the capability of dealing with uncertainty reasoning possessed by expert systems, it can be said that expert system applications will become more and more important to businesses operations.

Expert systems can perform different tasks, such as analysing and interpreting, diagnosing, prognosticating, and training. Whether these tasks can be successfully performed by an expert system application relies on making a good decision concerning the selection of the appropriate domain and methods chosen for constructing the system. Proposals for selecting an appropriate

knowledge domain and creating an expert system have been detailed in this thesis. In the process of producing an expert system, knowledge acquisition is recognised as the most difficult stage. Successful knowledge acquisition depends not only on the complexity of the knowledge structure but also on the cooperation of the expert.

The author used the common factors derived from her visits to six companies, as a basis for reviewing the current status of expert systems in UK business. The the problems and difficulties found during the development of expert systems in these companies are important pointers to help forecast the future trends in expert systems and to support recommendations as to the future research directions of expert systems in the next decade.

The author's literature survey showed that expert system user/developers in U.K. business were more involved in the use of shells than programming languages or toolkits. The results from the visits were in accord with this. from the survey of the eight expert system shell, it can be seen that current shells still need further improvement with regard to their flexibility and explanation facilities. If improvements are not made, the scope and type of business applications using shells will remain on a small scale and be able to perform simple tasks only.

It is also discovered by the author from her visits to the companies that for those companies that wished to develop an

expert system application but had no experience in programming, hiring consultants advisors is an effective approach. By this approach, the nature of expert systems and the skill of knowledge elicitation can be learned by companies in the process of system development. Companies can then apply this learning in subsequent to expert system developments.

Although uncertainty reasoning is emphasised as a necessary facility in most of the literature on expert systems, practical applications are not so concerned about this aspect of the new technology. From the visits to the six companies, the author discovered that none of the companies applied uncertainty reasoning; moreover, uncertainty was deliberately avoided by the developers. However, although this situation would not seriously obstruct the future development of expert systems, this would cause the incompleteness of knowledge representation and restricts the expert systems users' in the answers they can give.

With the understanding of the practical use of expert systems in business, the author was involved in producing a working system as part of her project. The complexity and difficulty of knowledge engineering for producing an expert system was fully realised when the author was involved in the production of an expert system for selecting an appropriate financial planning package. Acting as a knowledge engineer, the author had to elicit knowledge from her supervisor - the expert, by repeated interviews. However, it was realised that the success factors of

the creation of expert systems applications is not only the well bounded domain, but also the expert's capability of expressing his knowledge logically.

Choosing an appropriate development tool for developing an expert system application is another important factor in helping the success of an expert system application. In the author's case, although her project of developing an expert system application for helping organisations to select an appropriate class of financial planning packages has been successfully completed, her attempt of linking an expert system shell with a database package was failed. Had the author chosen another development tool, such as a frame-based shell, from the very beginning of her project, the result of her attempt might have been different.

From the author's visits to those representative companies and from reading the literature, current expert system applications in U.K. business is found to be at the primary stage. In the U.K., most of the developed systems are on a small scale and of a basic type. From the technical overview of the business applications given by the author, it can be seen that almost all of the current UK businesses involved with the new technology are applying the basic technology for building their applications. More complex and advanced technology, such as self-programming for large scale systems, is not yet being exploited because of the wide use of shells.

With the understanding of the advantages of shells and the rapid proliferation of expert systems shells available on the market, the author recognises the helpfulness and convenience of using a shell for developing an initial expert system on a small scale. However, the author expects that this situation may be improved if the developers could be more involved in the development of larger and more practical systems with more powerful development tools.

By looking back on the history of computer usage and the benefits that have been brought to business by traditional computing systems, the future development of expert systems is regarded as an extension of the expanding demand for computer systems. Also, it is expected that business will need expert systems more and more, and almost every part of business life can have an expert system applied to it. Whether this exceptation will be realised depends on the improvement of existing expert systems; these need well planned future research directions over the next 5 - 10 years. In the author's view, the future trends of expert systems development are not as bright as those optimistic forecast expected by many authors. The author bases her views on her own experience on developing an expert system, and the findings from her company visits .

Apart from the recommended future research directions of expert systems given by the author in Chapter 5, it is also strongly recommended by the author that the high-level management should

give more support and encouragement to their staff who are developing expert systems. The support and assistance from an existing computing department is an important factor of success for development of expert systems.

Although the author has raised the negative future trends of expert systems development, there are developments in this area that will accelerate the development and usage of this new technology. Examples are the support of the government for those projects, the increase of people interested in the field, and the increase of investment contributed for the development in companies. However, to accelerate the commercialisation and development of expert systems and realise the potential of expert systems, both theoretical and methodological efforts are needed. These issues should not only be investigated by the research and academic laboratories, but also by business itself to cope with its need for competence in areas of practical application.

REFERENCES

1. R. O. C. Information Industry Yearbook 1987, Taiwan Electronics, PP. 12-19.
2. Taiwan, The Unlooked-for Dragon, in The Economist. 27 Feb.-4 Mar. 1988. PP.46
3. G. Burch, Jr., F. R. Strater and G. Grudnitski (1979) Information Systems: Theory and Practice (2nd edition). Wiley, 1979.
4. P. N. Finlay and M. Forghani (1987)
   The Concept of Management Intelligence Systems, Journal of Applied Systems Analysis, Vol. (14) 1987.
5. J. Moss (1978)
   Introduction to Data Processing. Input Two-Nine, 1978.
   (Modular Computer Studies, Study Unit, E1)
6. J. M. Richardson (1979)
   Quoted in the Preface of Handbook of Information Technology and Office Systems, edited by A. E. Cawkell. North-Holland, 1986.
7. A. Burns (1984)
   Information Technology - for Better or Worse, in New Information Technology, edited by A. Burns, PP. 188-190. Ellis Horwood, 1984.

8. C. J. Martin (1988)

Series of discussion regarding the contributions of IT to business in May 1988.

9. R. S. L. Martin (1982)

An Introduction to Data Processing. Holt, Rinehart and Winston, 1982.

10. G. C. Lester (1980)

Data Processing, Polytech. Vol. (1) - Hardware and Programming, 1980.

11. H. C. Lucas (1973)

Computer Based Information Systems in Organisations, Science Research Associates, 1973.

12. D. W. Kroeber and H. J. Watson (1986)

Computer -Based Information Systems: A Management Approach. 2nd ed. Macmillan, 1986.

13. W. A. Freyenfeld (1984)

Decision Support Systems, NCC Publications.

14. P. N. Finlay (1987)

Series of discussion regarding information systems in Dec. 1987.

15. T. J. Murray (1979)

Data, Information and Intelligence in a Computer Based Management Information System, Journal of Applied Systems Analysis, Vol.(6). PP. 101-105.

16. D. Michie (1986)

Current Development in Artificial Intelligence and Expert Systems in Handbook of Information Technology and Office Systems, edited by A. E. Cawkell, PP.763-777. North-Holland.

17. P. Zorkoczy (1982)

Information Technology: An Introduction. Pitman, 1982.

18. M. J. Winfield (1982)

Expert Systems: An Introduction for the Layman, Computer Bulletin, Dec. 1982.

19. J. L. Alty and M. J. Coombs (1985)

Expert Systems: Concepts and Examples, NCC Publications.

20. J. R. Quinlan (1982

Fundamentals in the Knowledge Engineering Problem in Introductory Readings in Expert Systems edited by D. Michie, Gordon and Breach Science Publishers.

21. A. E. Cawkell (1986)

Expert Systems and Artificial Intelligence in The Handbook of Information Technology and Office Systems, edited by A. E. Cawkell. PP.141-148. North-Holland, 1986.

22. P. S. Sell (1985)

Expert Systems - A Practical Introduction. Macmillan, 1985.

23. P. Jackson (1986)

Introduction to Expert Systems. Addison-Wesley, 1986.

24. J. Cendrowska and M. Bramer (1985)

Inside an Expert System: A Rational Reconstruction of the MYCIN Construction System in MYCIN in AI Tools, Techniques, and Applications edited by T. O'shea and M. Eisenstadt. PP.453-497. Harper & Row Publishers, Inc.

25. C. C. Lee (1986)

Artificial Intelligence & Expert Systems, (Chinese Edition). Shung-Kung(1986).

26. E. A. Feigenbaum (1986)

The Handbook of Artificial Intelligence. Pitman. Vol. (1), edited by A. Barr and Feigenbaum.

27. N. Graham (1979)

Artificial Intelligence. Tab Book Inc.

28. N. J. Nillson (1980)

Principles of Artificial Intelligence, Tioga Publishing Comp. 1980 .

29. W. B. Gevarter (1984)

Artificial Intelligence, Expert Systems, Computer Vision and Natural Language Processing. Noyes, 1984.

30. P. Harmon and D. King (1985)

Expert Systems : Artificial Intelligence in Business. Wiley, 1985.

31. T. R. Addis (1982)

Expert Systems: an Evolution in Information Retrieval, Information Technology: Research and Development, 1. 1982, PP.301-324.

32. G. L. Simons (1985)

Expert Systems and Micros, NCC Publications.

33. M. Stefik, J. Aikins, R. Balzer, J. Beniot, L. Birnbaum, F. Hayes-Roth & E. Sacerdoti (1982)

The Organisation of Expert Systems: A Prescriptive Tutorial, Research Report VLSI-82-1, XEROX PARC, Palo Alto, CA.

34. Butler Cox Foundation (1983)

Report on Expert Systems

35. Dretske (1981)

Knowledge and the Flow of Information. Blackwell, 1981.

36. D. S. Prerau (1985)

Selection of an Appropriate Domain for an Expert System, AI Magazine, Vol. 6, No. 2.

37. R. W. Blanning (1984)

Knowledge Acquisition and System Validation in Expert Systems for Management, North-Holland Human Systems Management, 4 (1984), PP. 280-285.

38. M. Welbank (1983)

A Review of Knowledge Acquisition Techniques for Expert Systems, British Telecommunications, Marthesham Consultance Services, U.K.

39. M. D. Grover (1983)

A Pragmatic Knowledge Acquisition Methodology, Proc. 8th Joint Conference, Artificial Intelligence PP. 436-438.

40. A.Kidd (1985)

Human Factors in Expert Systems, Data Processing, Vol. (27) No. 4, May 1985, PP. 15-17

41. A. Hart (1985)

Knowledge Elicitation: Issues and Methods, Computer Aided Design, Vol 17, No. 9, Nov. 1985, PP. 455-462.

42. P. L. Alvey, C. D. Myers and M. F. Greaves (1985)

An Analysis of the Problems of Augmenting a Small Expert System in Research and Development in Expert Systems, edited by M. A. Bramer, Cambridge University Press, Cambridge

43. M. T. H. Chi, P. J. Feltovich and R. Glaser (1981)

Categorisation and Representation of Physics Problems by Experts and Novices, Cog. Sci. Vol. 5, No. 2 (1981), PP. 121-152.

44. H. A. Sowizral (1985)

Expert Systems, Annual Review of Information Science and Technology (ARIST), edited by M. E. Williams, Vol. 20, 1985, PP. 179-199.

45. L. A. Zadeh (1983)

Commonsense Knowledge Representation Based on Fuzzy Logic, IEEE Computer 16 (10). PP. 61-65.

46. E. H. Shortliffe (1976)

Computer Based Medical Consultation, MYCIN, Elsevier N.Y

47. C.L. Ramsley, J. A. Reggia, D. S. Nau and A. Ferrentino (1986)

A Comparative Analysis of Methods for Expert Systems, International Journal of Man-Machine Studies, Vol. 24, 1986, PP. 475-499

48. A. Burnett (1986)

An Investigation into the Design, Construction and Use of Expert Systems in Assisting Management Decision Making, Thesis submitted to Loughborough University of Technology for degree of Bachelor of Science, 1986.

49. T. Johnson (1984)

The Commercial Application of Expert Systems Technology, Ovum Ltd. 1984.

50. J. Hewett, S. Timms and G. d'Aumale (1986)

Commercial Expert systems in Europe, Ovum Ltd.

51. M. McLening(1987)

Thinking, Making Decisions with Clever Programs, Which Computer, Nov. 1987, PP. 112-118.

52. M. Arnold and D. Rush (1987)

An Expert System Approach to the Appraisal of Accounting Software, Management Accounting, Jan. 1987, PP. 32-35.

53. J. van Koppen, Philips C ISA/CAD Centre, The Netherlands (1986)

A Survey of Expert System Development Tools, Second International Expert Systems Conference, PP.157-173.

54. J. Hemus (1986)

Development of an Expert System Shell into a Management Tool, MBA thesis, Loughborough University of Technology.

55. P. N. Finlay, G.Forsey and J. W. Wilson (1987)

The Validation of Expert Systems - Contrasts with Methods for Mathematical Models, to be published by Journal of the Operational Research Society

56. NCONF, Network Configuration Expert System

Internal document of CASE plc.

57. CASE Communications and Artificial Intelligence, News Release, Nov. 11, 1986

58. S. Gibson and E. Horwitt (1986)

Mux Net Expert System Tops TCA Rollouts, Computerworld, Sep. 1986

59. Changing Data Networks, Financial Times, Nov. 7, 1986.

60. A. Goodall (1986)

Railway Route Planning in Australia, Data Processing, April 1986, Vol. 28, No. 3, PP.136-138.

61. R. G. Ruckwood (1985)

Thomas Cook's Experience in Developing a Micro-Computer Expert System, Internal document of Thomas Cook Group Ltd.

62. P. Kelly (1986)

Expert Systems Using Shells, Data Processing, Apr. 1986, Vol. 28, No. 3, PP.139-141.

63. B. Humpert and P. Holly (1988)

Expert Systems in Finance Planning, Expert Systems, May 1988. Vol. 5, No.2.

64. M. A. Bramer (1988)

Expert Systems in Business: a British Perspective, Expert Systems, May 1988. Vol. 5, No.2.

65. S. Dickman (1987)

Development of an Expert system for Use in the Fault Diagnosis of Computer Equipment, MBA thesis, Loughborough University of Technology.

66. P. N. Finlay (1987)

Mathematical Modelling and Expert Systems, submitted for publication, Journal of the Operational Research Society

67. P. N. Finlay (1987)

Using an Expert Systems Approach to Structure the Selection of Computer Software, submitted for publication, Journal of the Operational Research Society

68. P. N. Finlay, M. King, and A. Burnett (1987)

Administering Rule Development in Rule-Based Expert Systems, submitted for publication, Journal of the Operation Research Society.

69. Expertech Ltd. (1986, 1987)

Xi Plus user manual, Expertech Ltd.

70. Intelligent Environments Ltd. (1986)

Crystal user manual, Intelligent Environments Ltd.

71. Intelligent Terminals Ltd. (1986)

SuperExpert user manual, Intelligent Terminals Ltd.

72. Helix Expert systems Ltd. (1986)

Expert Edge user manual, Helix Expert systems Ltd.

73. NCC Publication

ES/P Advisor user manual

74. NCC Publication

Expert-Ease user manual

75. Intelligent Terminals Ltd.

Product Overview: Expert Ease

76. NCC Publication

Micro Expert user manual

77. NCC Publication

MicroSYNICS user manual

78. R. J. Allwood, D.J. Steward, C.Hinde and B.Negus (1985)

Report on Expert System Shells Evaluation for Construction Industry Applications, Dept. of Civil Engineering, Loughborough University of Technology

79. Expert Systems Go to Work, Data Processing, Vol. 28, No. 3 April 1986, PP.121-135

80. R.Forsyth (1987)

Software Review - Expertech Xi Plus, Expert Systems, Feb. 1987, Vol. 4, Nov. 1, PP.48-51.

81. Expertech (1986)

Guide to Know-How Programming

82. O. Linderholm (1987)

Crystal & VP-Expert, PC Week, April 1987, PP.142-146.

83. Expert Ltd. (1987)

Annual General Meeting held in Coventry on July 2nd and 3rd.

84. P. N. Finlay (1985)

Classes of Financial Planning Package - A Management Perspective, Long Range Planning, 38, 6, PP 94-98.

85. P. N. Finlay and T. Servant (1987)

Financial Planning Package, NCC Publications

86. Expertech Ltd. (1987)

XiUser Newsletter, June 1987

87. P. N. Finlay and T. Servant (1987)

Financial Planning Package

88. J. M. Taylor (1986)

Expert Systems: Where do We Go from Here?, KBS 86, Knowledge Based Systems, International Conference, PP.313-324.

89. R.C. Muller (1986)

New Directions in Expert Systems, KBS 86, Knowledge Based Systems, International Conference, PP.325-333.

90. R. Jones (1986)

Commercial Expert Systems - How to Avoid the Pitfalls, Data Processing, vol. 28, No. 3, April 1986, PP.115-119.

91. A. d'Agapayeff (1984)

Report to the Alvey Directorate on a Short Survey of Expert Systems in UK Business, Alvey News, Supplement to issue no. 4, PP.11.

92. A. d'Agapayeff (1987)

Report to the Alvey Directorate on a Short Survey of Expert Systems in UK Business, Alvey News, PP.1-2.

93. Weiss & Kulikowski (1983)

A Practical Guide to Designing Expert Systems, London Chapman and Hall, 1983.

94. P. N. Finlay (1988)

Series of discussion regarding the future of expert systems.

95. C. J. Martin (1986)

A Behavioural Analysis of Interactive Computer Systems by Senior Managers. Ph. D thesis at the University of Durham.

96. Feigenbaum and P. McCorduck (1983)

The Fifth Generation: A.I. and Japan's Computer Challenge to the World, Addison-Welsley Publishing Company.

97. H. L. Drefus (1983)

Expert Systems versus Intuitive Expertise, University of California, Berkeley.

APPENDIX A CASE STUDIES

a. A Petfoods Manufacturer

b. A Nuclear Power Designer

c. A Datacommunications Firm

d. A Travel Agency

e. A Foreign Bank

f. A Guard Security Company

CASE STUDY A - A PETFOODS MANUFACTURER

I. The Company

Company A is a major petfoods manufacturer which has been established for more than 12 years with two thousand employees and a 40 million pound turnover located in the Midlands of England.

In order to prevent their experts being consulted about minor problems of procedures by the operators, especially during the night, an expert system was deemed necessary. At the start of the development, in 1985, the company conducted a general survey of expert system shells available on the market.

II. The Expert System Shell

A. Obtaining Information about Expert System Shells

The company had looked at 3 shells before selecting Xi Plus. From information about which expert systems were available obtained from sources such as Computer Weekly, PC User and 2 or 3 expert systems journals, the company found that unless they bought an expert system shell, it would be very difficult to obtain sufficient details about the shell. In this company's case, they used Burnett's (48) opinions on Expert Ease from his thesis. In addition, they obtained some detailed information from the NCC starter pack. Expert Ease is one of the shells in the pack. Furthermore, they spent time persuading their computer

department of the necessity to investigate expert system expert shells thoroughly, some of them were put in touch with the expert systems experts who had gone through working systems with Xi Plus. With the help of their computer service group, they had conducted a limited survey of which expert system shells were available on the market.

B. Reasons for Choosing Xi Plus

1. Easy to use

Unlike some of the other expert system shells, Xi Plus is easy to use. The company had had experience of a sophisticated shell before, the manual for which had proved difficult to follow, unless the user was a computer expert, very familiar with PC operations, knew how to load/transfer files quickly and understood the computer jargon written on the screen.

2. Good manual

The manual of Xi Plus is thought to be one the best features of the system, with easy access to the tutorial manual from the screen, Xi Plus is easy to learn.

3. Free reasoning method

Of the expert system shells investigated on the market, some are dedicated forward chaining, some are dedicated backward chaining, some are able to deal with uncertainty, whilst Xi Plus is not only capable of both forward and backward chaining, but also of handling uncertainty reasoning to a certain extent.

(It should be noted that the author questions the handling capability of uncertainty by Xi Plus, because a definite answer (e.g. yes, no, maybe, don't know, etc) for a question query is necessary, no matter how uncertain the response may be).

III. The Expert System

As mentioned above, the company established the boundaries of the domain on product quality control process. Having carried out this step, they organised their knowledge into a knowledge base.

A demonstration of a working system named Soft Texture was given for the interview. This system was developed for the purpose of eliminating the factors which cause problems in the quality of the product. Using forward chaining, it contains 38 rules. Due to illogical rules and erroneously identified variables, conflicting values or unlikely conclusions sometimes resulted.

The demonstrator held the position of 'expert as knowledge engineer'. This represents a major advantage for debugging and testing, because the combination of the two roles makes it easy to diagnose errors and rectify the knowledge base quickly.

During the demonstration, a problem was raised by the demonstrator that the message "syntax error" was displayed on the screen whenever he added any new query or rule to the knowledge base. It was found by the author that he used capital letters

instead of lowercase. Although the screen gives no indication of why the system message "Syntax error" is displayed at this point, the manual does mention the necessity for using lowercase rather than capital letter when adding any new information to the knowledge base.

IV. Process of System Building

Four people have been involved in this application. Of the four people, one person provided technical advice on the usefulness of expert systems, whilst three experts from different areas provided their professional product knowledge. The demonstrator was therefore not the only expert and he had to ascertain the knowledge of product quality from the other two experts in order to conduct the interview which was not held in a formal way; it took a long period of time in conversation with the others to extract the knowledge.

Being from an engineering background and also being one of the experts, the demonstrator was confidently able to give his diagnosis when problems arose, and thus he had no problem in structuring the problem domains and setting up the rules. Five units of problem domains were consequently structured and built as expert system which transferred the experts' knowledge into the computer. So the demonstrator was definitely able to assess whether he had obtained enough information from the other experts. An important point made by the demonstrator was that a

general engineering background is good training for being an expert system builder.

For testing and debugging, the demonstrator would show his system to the other two experts individually, each would then look through the rules from a basic level and give their comments about the system. After agreement had been reached by the experts, the system would be used by the production quality department and shift managers. Because this system was not sophisticated enough, it was not possible to examine the problem of wrong inferences. To prevent a problem of this nature, they could only keep running the system and testing out all of the different possible combinations as well as making sure that they had been obtaining the right answers.

V. Comments from the Shell User

Generally, the demonstrator is satisfied with the shell (Xi Plus) which he used. A few comments with regard to Xi Plus were given as follows:

a. Difficulty in getting problems ready structured before inputting into the computer.
b. Difficulty in becoming familiar with the IF....THEN model for rule input.
c. Difficult to think about problems of backward chaining or forward chaining.

d. Different values for two identifiers which actually have the same meaning but are input differently or have different means of expression.

As the interviewer knows, it is not only Xi Plus that causes these inconveniences but also many other shells present the same kinds of problems to their user (See Chapter 4).

VI. Conclusion

Measuring the success of the system has nothing to do with cost reduction, since the demonstrated system was related to product quality control. As can be seen from the description of the expert system, this system is still in the developmental stage, and is not ready to be used by the end user - the operators as yet.

The knowledge engineer should always be aware of ways to improve his system, this should include reviewing the integration of the knowledge of experts and familiarity with the manual of the shell which is being used. This necessity may be obvious, especially in a company which is developing a working system, but from the interview regarding the essential fundamentals of developing the system at Company A, a user-friendly system has not been developed yet.

CASE STUDY B - A NUCLEAR POWER DESIGNER FOR NAVY SUBMARINES

I. The Company

Company B is a nuclear-power-supply designing organisation whose main business is designing nuclear power applications for navy submarines. This company was set up in 1959, and wholly dedicated to military work up to a few years ago. But now, with approximately 1,300 employees, they develop products for outside customers, because of the unreliability of military support.

More than three years ago, the technical director showed an interest in expert systems, this encouraged many people of the company to get more actively involved in this area at the beginning of 1986.

Up to now, there has not been a working system successfully developed by this company, but they have projects in mind and are heavily involved in the preparation of systems development. A lot of effort has been put into projects, and a separate group of four people is designated to be specifically involved. Of this group, two members are from the computing department whilst the other two are from the engineering department.

One of the interviewees, who attended an MSc course about knowledge base systems, has been attending the training course for 4 months held by Edinburgh University. The company perceived

that this is the only way they can obtain individual help (indeed, they have obtained some useful suggestions of awareness of approach from the training.) in the use of expensive machines and packages. The total cost of company investment will be around £20,000.

The company has Expert Ease,Crystal and the NCC Starter Pack. During the next few months,they will conduct a review and evaluation of the largest shells on the market. Hopefully, by the end of this year, they would come to a decision about which one is suitable for purchase.

II Reasons for Using Expert Systems

As most other companies, Company B has established its computing department for the reasons that the computer can do things more easily, handle tedious and routine work without complaint and does not make error, providing it has been properly programmed. In addition to their knowledge of computer applications, they know expert systems can capture the knowledge of experts to make that knowledge more universally available. Hopefully, this will enable their experts to be given time to become more involved in constructive and productive jobs. Naturally, management will have to introduce expert systems' usage to the experts in the right way and give proper encouragement.

III. Obtaining Information about Expert Systems Shells

In 1986, the interviewees went to an expert system shells exhibition at Brighton, there they chose to buy Expert-Ease and Crystal. As well as obtaining knowledge about SAVOIR from the Welding Institute, they are now learning SAVOIR from the training course. Furthermore, a general evaluation of expert system shells was published by Data Processing in April 1986 (77), which gave the company important indications for future purchases.

The manual of SAVOIR is considered very difficult to follow, therefore, it is believed that the training course offers a better structure than learning through following the manual.

IV. Factors for Choosing an Expert System Shell

Major factors for selecting a suitable expert system shell for Company B are:

a. Compatibility - the diagnostic system of the shell must be compatible with the current inference methods used in their projects, e.g. rule-base induction, backward or forward chaining.
b. Cost - the vendor should provide the cheapest price with the relevant services that fulfil their requirements.

There will be more factors for choosing an expert system shell when the company has decided to buy another shell.

V. The Projects

Company B has 2 very different projects in mind. The major one is for plant diagnostic analysis of on-line signals, which should give the operator the required advice in any emergency situation. In this plant,there are more than 2,000 factors that could cause signal problems, therefore, estimation for the number of rules for the system is a complex task, probably thousands, because it is such an intricate major system.

The second project is smaller with about 30 rules, which is designed to search for situations for extracting chemical transients from sea water.

In addition to these 2 projects, Company B also provides seminars and training with regard to expert systems for their engineers, technicians and operators.

VI. Eliciting Knowledge from Experts

Although Company B is contemplating the major system, the interviewee is currently dealing with the smaller project which has 30 rules and trying to expand it through SAVOIR, because of the expectation of obtaining help from the training course.

Being of an engineering background, but not a chemist, the interviewee must elicit knowledge from an expert in chemistry for 3 or 4 sessions of approximately one hour's duration each before he attends the course. Also, with some study of the chemistry

manuals, he can learn something further about chemistry. During the interview, he does not talk in terms of rules, i.e. IF....THEN , but just ask the experts to tell him what they are doing or how they tackle a specific condition. After their interview, he isolates himself from the experts and tries to recall and understand what was said until he feels confident enough of his knowledge and is able to generate more questions, for which he returns to the experts for further clarification.

Therefore, he has to structure the interview, clarify the questionable points in his mind himself or through the experts, until he grasps the whole picture. At the end, he shows his results to the experts and asks for their corrections, if any.

VII. The Future of Expert Systems Application and Foreseeable Problems

Company B expects that the advisory diagnostic system could take over the control of the plant in the future. To Company B, it will be a major step and there is a long way to go from the present before allowing the machine to control the plant.

The major question of concern is the validation of expert systems. Company B has highly trained experts to deal with emergency situations. However, they may not know how to retain their expertise through lack of practice if all expertise is passed into an expert system. They may find that reliance on the expert systems may usurp their expertise which may further

decline through lack of practical use. Moreover, after becoming reliant on expert systems, if a computer failure occured, the company will be in major difficulties, especially if there is no well-trained operator present. Such difficulties represent a particular danger in the field of nuclear engineering.

VIII. Conclusion and Review

The lessons which can be learnt from the interview with Company B are those relating to the preparation and effort required for developing large-scale expert system applications and the procedures for developing an expert system , as well as validation of developed systems.

In the procedure of eliciting knowledge from experts performed by Company B, it seems to the author that the 'knowledge engineer' pays too much attention to learning or simulating the experts' jobs. It should be noted that knowledge engineer is never regarded as an expert of a particular job, and it is also not possible for a non-expert to learn all the details necessary for a complicated, major system.

A methodology is needed for Company B which will lessen the knowledge engineer's need to know complex details before programming or structuring the rules based on the knowledge of the experts and they are endevouring to develop this methodol

CASE STUDY C - A DATACOMMUNICATIONS FIRM

I. The Company

Company C, established fourteen years ago, is a multinational company with headquarters based in North London, UK. and subsidiary companies in America, Australia and Hong Kong. It has an annual turnover of £100 million with approximately 2,000 employees worldwide (1,100 strong in UK). Affiliated with network distributors throughout Europe, Asia and other parts or the world, this company is one of the largest datacommunications companies in Europe.

The main concerns of Company C are centred upon the development and manufacture of point-to-point communication devices (e.g. DCX range, modems, etc) into full data networks which can be spread across a number of countries in the media of data transmissions from terminals or computer boards. These types of products represent huge, private datacommunications work for companies. To date, this company has installed more than 800 networks worldwide.

Three service groups comprise the computing department of company C:

- the first establishes communication services within the company i.e. a 20 nodes network installed in th UK is also connected to the part of the company located in the USA.

- the second deals with the traditional data processing applications to enable the company to operate on a routine basis e.g. book-keeping, inventory control and the generation of consolidation reports worldwide.
- the third group's role is to develop new products principally involving the usage of expert systems and their possible applications.

II. Reasons for Developing/Using Expert Systems

Company C is a market-driven company and two years ago it decided to become involved in the area of expert systems and to set up a project for the following reasons:

a). There were very few AI applications available on the market. Yet, as a set of knowledge bases, AI has the potential to enable software technology to become marketable products, profitable for the company.

b). Some of the technical tools used in the field of AI are ideally suited to Company C's products because the company is involved in datacommunication devices. It is of fundamental importance to the company that they develop a methodology or a technical tool to increase productivity or improve the product's image. An expert system developed inhouse would greatly assist in enhancing the company's reputation.

c). The management of the company recognised that using and developing expert systems of a high standard to reduce time consuming tasks or to assist in cost-saving projects would represent a useful, long-term investment for the company.

The project was finally developed as an expert system which includes a database system which is able to control, monitor and configure a very large data network and is regarded as one package in a large system of network management tools.

III. Reasons for Using OPS 83

The interviewee, a technical Artificial Intelligence strategist of Company C, had made a thorough review of the literature, evaluating all available software tools for AI application.

OPS 83, a general purpose AI language developed at Carnegie Mellon University in the US, is defined as an innovative production system. In the interviewee's opinion, OPS represents the fundamental rule - based language on which all such languages are based.

The version OPS 83 is a compiled procedural language as well as a rule based language. The most important feature of OPS 83 is the control mechanism for inducing the rules which can be used as a customised inference engine. In addition, two other aspects of OPS 83 render it a useful system for the datacommunications industry. Firstly, it can be used to develop a large rule - based system of 20,000 - 40,000 rules. Secondly, it is interpreted by a fast compiler, thus it can operate at a rapid rate.

Compared with PROLOG, OPS 83 is easier to modify, with an efficient capacity for debugging the knowledge base. Moreover, PROLOG does not possess all of the required facilities to develop the application which Company C desires e.g. efficient number crunching, generating data for analysis etc. Furthermore, the price of the compiler for PROLOG exceeds that for OPS 83.

IV. Application and Knowledge Base Development

Before deciding to develop the expert system, Company C conducted a survey for its 800 customers who were working with data networks. It was discovered that each customer had been spending anything from a few hours to 10 man-years to configure the networks. The basic reasons for this are that these networks are very complex and connected to a number of computer boards which are updated on a more or less weekly basis: whenever the boards were updated, the customers had to reconfigure them, whilst most of the network maintenance is carried out inhouse, it was decided that the particular application chosen for expert system was for the purpose of problem-solving.

At least 5 to 6 man-years had been devoted to the development of the system. During development, Company C gathered together a number of software engineers and knowledge base engineers as well as consultants who were experienced in this field i.e. configuration and application of data networks. Joint consultations took place to determine criteria for the system's functionality. The experts' advice was incorporated into the

system as a series of improvements until the knowledge base was completed.

The system OPS 83 can, in fact, accommodate uncertainty on the part of the users, but Company C avoided inputting ambiguous questions. Although it is recognised that some answers should be optional, very few system users are capable of coping with uncertainty, therefore, Company C has chosen to create an expert system which can produce an expert solution without rendering it necessary for the user to understand the configuration of the system as a whole.

V. The System

The product, Network Configuration, designed and manufactured in the UK is currently sold in the US. It is an expert system designed to configure and reconfigure datacommunication networks and solve the problems experienced by operators of large datacommunications networks of how best to handle the regular and inevitable changes in the physical and logical structure of the network.

The system, can effectively emulate human expertise by applying the techniques of logical inference to a knowledge base. Once given the topology of a network, with the number of sites, links, high and low speed channels, the system produces a physical configuration comprising a device map, a channel map and DCX

routing map. In addition, the system generates the actual load maps for each node.

Company C, itself, operates a data network which contains 40 nodes. According to the interviewee, a reconfiguration, which had previously taken several weeks, can now be carried out in five minutes.

VI. Validation of System

Two methods used to test and validate the system were:

1). Dividing the process of configuration into segments to ensure that the logical sequence of the system is procedural.

2). To try to assess the quality of result produced by the expert system comparing it with results produced by humans and asking experts for their comments.

This system, being sold in America, represents part of a large network of management tools has a network configuration. Company C had to take into consideration the criteria for integration before the system was published.

According to the interviewee, no modification has been carried out to date, because it is believed that Company C has better knowledge and expertise in configuring data networks than the

customers, especially as some experts are using OPS 83. The difficulties first time users experience can also be overcome.

The system was demonstrated at the interview. Its advantages can be detailed as follows:

- easier physical network implementation
- cost saving
- improved planning of network
- better network manageability and easier maintenance
- faster network development

In the light of these advantages, Company C does not foresee any problems in selling the system in the UK.

VII. Conclusion

The lessons which can be learned from this case study are establishing criteria for developing an inhouse customised expert system, how to choose the expert systems development tools, and the validation techniques to ensure quality control of the system. It is not so easy to develop a customised expert system from the very beginning of inhouse programming. Fortunately Company C has experienced programmers and experts, with the survey of 800 customers' daily problems in data networks, they finished their first commercial expert system.

CASE STUDY D - A TRAVEL AGENCY

I. The Company

Company D, having been established for nearly 150 years, is one of the world's largest travel agencies with its head office in Peterborough.

As part of the company's policy is to monitor the progress of information technology, three years ago the Research and Development department started to study knowledge base systems, and eventually it engendered an interest in expert systems much greater than expected. A proposal to develop a trial system which could handle both human knowledge and numerical data emerged, then the researchers began to evaluate the advantages and explain the nature of the technology to convince their senior managers of the benefits involved. Finally, they obtained an £80,000 budget spread over 4 years.

II. Finding a Suitable Project

The company did not start development until March 1985 for the following 3 reasons:

. Before 1984, they had no experience of developing an expert system. They needed consultancy advice to help them identify a suitable application.

. They were unfamiliar with knowledge base systems, and yet they wanted to participate in the proposed development, so they needed time to learn about the new technology.

. They were seeking the appropriate consultancy advisors who would be able to provide training and advice on developing expert systems.

Having chosen an outside software company, Company D then tried to find an application which would be suitable for a pilot system. The company considered that the aim of the trial expert system was to educate themselves and make the expert's knowledge available to the inexperienced staff.

With consultancy advice, a project to build an itinerary planner for the Railways of Australia network was defined based on the following factors:

. The problem must be of a small scale and typical of the travel business because the company did not wish to overburden existing staff with an unfamiliar problem domain.
. The area of application must be restricted to an identifiable expertise so that the knowledge acquisition could proceed smoothly.
. The expert must be enthusiastic and must think logically.
. The completed system would have potential for further application.

Company D deals with all enquiries and bookings relating to the Railways of Australia. Whenever the staff concerned answered telephone enquiries with regard to travelling in Australia, they used to refer to the expert who was the only one in the company

who had travelled in Australia, so he had direct knowledge about it. The company treated the trial system as part of their learning programme, from the knowledge acquisition they wanted to know how successful the expert was and how to make the expert's knowledge available to the others.

The above points constitute the reasons why Company D chose itinerary planning for Australia Railways as their first application. Therefore, the company sought a shell accompanied by an application for itinerary planning. Unfortunately there was no such application available on the market at that time. Also, they found that using a developed packaged system would restrict their knowledge into one limited area and confine their plans for further applications as well as diminishing their development efforts. So, they decided to develop their own system using the Prolog language.

III. Knowledge Acquisition

For developing the system, Company D devoted 1.5 to 2 man years. The system was developed jointly by Company D and an outside consultancy firm. The company was able to program the basic features and the outside consultancy advisors programmed the more complex features.

In the process of development, the programming stage proceeded more smoothly than originally anticipated because a great deal of the information used by the expert came from printed timetables.

The expertise was his skill in using the timetables, knowing relevant information which was outside the scope of the printed form, for example, sleepers are either all double or all single in a carriage, and the regulations regarding train passes and booking procedures. However, the real problem was the effort needed to maintain up-to-date information for the timetable and this difficulty affected the development process by necessitating a longer period of time for interfacing the expert system with the database.

For the unwritten knowledge concerning travelling on the railway network, Company D collected data from all telephone conversations between the expert and the customer, and talked through example itineraries with the expert. Even so, they can not be certain that they have input the complete knowledge, not only of the updated timetable information but also of the 'unwritten' information since it is possible that the expert may be unaware of certain details outside his experience, these gaps in his knowledge may not easily be rectified because of the additional problem posed by the fact that Australia is so distant from the United Kingdom.

IV. The System

The system was designed based on three criteria:

1. It is to be used mostly by inexperienced staff who have little keyboard skills, and, moreover, this system is to be used when

staff are answering telephone enquiries from customers. Therefore, the system must be easy to use.

2. Bearing in mind the necessity to update the timetables and 'unwritten' knowledge, the system must be easy to maintain.
3. The system should be capable of dealing with more than one constraint given by the customer, e.g. lowest cost, shortest journey time.

When beginning the query, the system displays a coloured map of Australia with some major towns of the railway network system highlighted on the screen. Routes are planned from the points of departure and arrival, and the dates are chosen from a calendar displayed electronically on the screen.

The expert's knowledge is represented as knowledge rules to guide the system searching the optimum schedule and indicating additional information available that does not appear in the timetables. The knowledge is therefore not just confined to timetable information, but also includes additional useful information supplied by the expert. These are essential features in planning high-quality itineraries to meet customer requirements. The system consists of approximately 3,000 facts and rules.

This system is currently used by staff inexperienced in computers to answer telephone enquiries.

V. Validation and Future Prospects

For an expert system with a particular application with knowledge elicited from one specific expert, the validation procedure performed by Company D simply consisted of the expert's comments, whilst the company agreed that a more comprehensive validation plan should be investigated in advance of developing a large sized expert system.

From the pilot system Company D recognised the practicabilities of making effective use of computers. Most importantly, they are convinced of the value of expert systems technology. Therefore, the company is considering using their expert systems technology to publish electronic timetables as well as developing an actual route planning system on a mainframe machine.

VII. Conclusion and Comments

This system was jointly developed by both the client and the outside software company, it consists of written timetables, written information, and, unwritten information supplied by the expert. Although the system can initiate the expert's dexterity fully in planning a journey to meet the customer's requirements, it should at most be considered as an expert/information retrieval system, because:

. Approximately 3,000 facts and rules comprise the system whilst a large part of them are timetable data and itinerary maintenance & retrieval.

. This system was designed for a particular purpose for Company D, it may not be suitable for other travel agencies unless the same business is done by both..

The main reason Company D treats the system as an expert system is because the implementation of the program involved using Prolog, an artificial intelligence language. However, in the author's opinion, although using conventional languages, e.g. COBOL, FORTRAN, would make the program much longer, no reason could be seen why the system would not work using such a language.

This case study reveals the importance of choosing an expert system or a conventional system to solve an identified problem. One of the major characteristics of expert systems is its use of human thought, but if the expert's knowledge in performing his job is completely procedural and the size of application domain is small, then it is not necessary to use an expert system. In the case of Company D, this system is only a trial system as well as a training aid for their staff for being familiar with the new technology, evaluation of the benefit is not important, nor of the discussion of the necessity for system developing.

CASE STUDY E - A MAJOR INTERNATIONAL BANK

I. The Company

Company E is a foreign investment bank located in central London with its headquarters in the United States and branches all over the world.

One of the bank's major business tasks involves Letter of Credit (L/C) affairs. When the overseas issuing bank issues a L/C on behalf of an importer (the buyer), the L/C is sent to the bank in London, which represents the exporter (the vendor) to draw funds up to a specific maximum total from the bank. The bank advises the exporter of the terms of L/C, the exporter ships goods to the importer and prepares the necessary documents in accordance with the requirements of the L/C and submits them to the bank to ask for payment. The bank checks if the documents provided meet the L/C requirements. If there is no discrepancy or only a minor defect which can be accepted by the bank, the bank pays funds to the exporter, otherwise, payment is withheld until amendment of the L/C is completed.

According to the interviewee, a member of the senior staff of the Documentary Credits department, nearly 50% of documents submitted to the bank for negotiation under the L/C were returned for amendment because of discrepancies. If the discrepancies are caused by the importer, the exporter asks the importer to make

the corrections. It was calculated that £16 million Pounds loss per year was incurred alone in this country in terms of interest, goods damaged etc. Therefore, for the purpose of saving money and the time of the exporters, importers and the bank itself, the bank decided to design an expert system in th summer of 1985.

II. The Expert Systems Shell

A. Reasons for Using Helix's Expert Edge

The bank has its own computing department which only deals with traditional DP/MIS activities rather than I.T development. The interviewee, who was in the position of 'expert' during the period of developing the system, had no knowledge about computers. Therefore, the bank decided to ask the help of an outside software company. Helix Technology Group was the one technical company that the bank contacted and it was their confidence in developing the expert system for the letter of credit advisor by using the Expert Edge shell which made the bank choose Helix as the developer.

B. The Shell

Expert Edge is a shell for IBM PC written in C language with a good text handling ability, backward chaining logic and a window system. Detailed discussion of Expert Edge is given in Chapter 4 of this thesis.

III. The Process of System Building

For developing the system, both Helix and the bank provided the knowledge engineer and the expert respectively. The knowledge engineer had no knowledge about letter of credit, therefore the methodology used for knowledge acquisition was carried out by the most conventional methods of an interview with the expert and modification of rules, a cycle that was repeated as necessary.

The bank originally expected the system to meet all the requirements of international trade regulations for L/C documentation, but it was found that although the knowledge engineer could technically implement thousands of rules, a commercial expert system could not be built because of the considerable costs involved. Eventually, a compromise was made to exclude uncommon terms and exceptional cases of L/C from the expert system. This means that manual manipulation for some particular cases is needed. This development resulted in a knowledge base of some 260 rules.

IV. The Expert System

The system was developed based on the following requirements:

- can be used as teaching material for the bank's training courses in trade finance.
- can be used as a training aid for the bank's own staff from the Documentary Credits department.
- can be sold as a software product to other banks and exporters.

The domain was restricted to common types of L/C documentation, i.e. draft, invoice, packing list, certificate of origin, insurance document and transport document.

The intention of making the expert system a training aid, as well as to be useful to both banks and exporters, means that the system should have an effective user interface, i.e. a friendly design and wording of the interaction between the user and the system. For example, some particular jargon may not be comprehensible to trainees yet they need education and training. The system was therefore designed using different texts for different users.

The system asks the user many questions. The user needs to know about draft, invoice, bill of lading, etc. and to be able to understand the language of L/C. The user has to find the correct answer from the documentation supplied in response to the questions asked by the expert system. Answers for each question are either yes or no, no uncertain answers are allowed. the system asks the next relevant question according to the user's answer given to the previous question. If any discrepancy is found, the screen will display it an tell the user the correct action.

The knowledge base of 260 rules can easily be expanded to incorporate any particular regulation to meet the requirements of a particular user, such as rules for specific countries or

uncommon documents. Therefore, this system can seemingly be called a 'core expert system' which contains knowledge applicable to any user but which can be expanded to meet the individual's needs. It can be said that this system is easy to maintain by individual users.

V. Validation and testing of the System

A two-stage approach for the validation of the system was applied during this project.

Firstly, the Helix knowledge engineer tested the system logically without participation of the expert in order to eliminate programming errors.

Secondly, the expert tested the system against actual selected L/C's in order to ensure that the system behaved as the expert expected. Twenty-five L/C's were carefully chosen to cover the full range of L/C issues dealt with by the bank until the frequency of disagreements between the expert and the system reached an acceptable low level.

VI. Comments and Conclusion

According to the interviewee, the system is mainly used as a training tool for both the bank's staff and training courses for trade finance, rather than to relieve the expert's daily workload, because, as the expert said, the bank believes that an expert system can never be a substitute for human's work,

especially as the L/C verification needs the expert's judgment frequently, for example, whether or not minor discrepancies caused by some particular companies are acceptable.

The bank's recognition of and efforts in developing an expert system are appreciable whilst its concept of using expert systems up to certain limitations renders any usurping of the expert is debateable. Although up to now there has not yet been a real expert system which is able to replace the human expert completely, there are cases,e.g. Case Study C and D, which have proved the success of using expert systems as an assistant to the expert so that the expert may be released to do more work at a higher level.

CASE STUDY F - A GUARD SECURITY COMPANY

I. The Company

Company F is a company owned by a large multi-national Australian firm. Located in Nottingham, this company has been established for seven years with around 80 employees and a 2 million pound turnover.

The main business activities of Company F are centred upon the installation and maintenance of close circuit television systems, fire monitoring systems, intruder detection, alarm systems and access control systems. A small but important proportion of Company F's business involves the installation and maintenance of intruder systems, which are manufactured by the company, for its own use and for exporting to Europe. This equipment can be used for process control in industry. All of these systems are monitored 24 hours a day by the company's Central Station facilities located in Nottingham and London.

There are over 3,500 clients covered by these systems at present, the cumulative risk is believed to be several millions of pounds. All of the responsibilities for maintenance of the high security monitoring equipment lie with the Systems Department of Company F.

II. The Project

Company F has four sections that support its daily operations, they are: the administration section, the national sales representatives, the systems department and the control station. The service group of the National Sales Representative section is responsible for the regular servicing of the monitoring alarm system. The service and installation engineers are responsible for the call out service in the event of alarm system failure, the call out service is on a 24 hours basis.

The systems department is a small group in Company F consisting of three persons: the National Engineering Manager and two systems engineers. One of the main functions of this department is to respond to systems failure in the UK. The sites for possible failure in the UK are situated at Derby, Sheffield, Alfreton, Leicester, Nottingham, London and Wolverhampton. The two systems engineers are on-call 24 hours a day, normally on a one week on, one week off basis.

The interviewee, the manager of this department, undertook the project of the development of an expert system for use by the employees of Company F as his MBA project at Loughborough University of Technology.

III. Current System Overview

The system now being used by Company F for monitoring basically consists of a PDP-11 computer system at the central station. This system has two computers working in parallel, i.e. if one

machine fails the second can take responsibility. The necessary security against unauthorised access to the computers and the back up power supplies for power loss or generators for long term power failure are also supplied as facilities.

The two computers communicate with the outside world via the multiplexer which consists of a controlling circuit and a number of modem cards. Via the leased Telecom circuits provided by British Telecom, these computers are able to communicate with Company F's District Data Centres (DDC) which can communicate to the company's clients or alarm panels. All communication between the alarm panels and the DDC is monitored by the computers.

IV. Reasons for Developing the Expert System

In order not to overload the operators with false alarms and fault reports, Company F uses various methods to reset or re-start the computers and microprocessors within the system when a fault occurs which causes them to cease functioning.

In spite of the methods, problems in the system occur in the data transmission between DDC and the clients' panel or between DDC and the multiplexer. These problems are mainly caused by 'noisy' lines which result in data errors, however they are overcome by using error checking and re-try techniques on all of the telecommunication links. Even so, these problems inconvenience the complete network and may cause equipment failure.

There were four reasons why Company F decided to develop an expert system. Firstly they needed an expert system which would be able to perform the fault diagnosis of faults occuring within the company's system from the multiplexer through the DDC's and onto the client's alarm panel. Secondly, the company wanted to reduce the call-out cost for computer faults of DDC which required the two systems engineers to travel extensively in the country. Thirdly, the reduction of an additional call-out cost for alarm panel faults which needed the service engineers' attention was also considered. Fourthly, in addition to the above reasons, the frustration felt at being continually called out for similar faults and the disruption caused to other work prompted Company F to investigate the possibility of developing an expert system to solve repeated failures occurring regulary.

V. The Expert System

This expert system deals with the diagnosis of three faults which occur on Company F's system - DDC faults, alarm panel faults and telecommunication faults. It was calculated that in the last year the above faults represented over 90% of the total faults occuring in the system.

The development system would ultimately be used by the company's operator at Central Station. Upon receiving the fault condition the operator would consult the expert system which would give instructions about how to resolve the fault or instruct the

operator to call out the systems engineer if necessary. The knowledge of the systems engineer was built into the expert system to enable faults to be cured in the most efficient and cost effective manner. The knowledge was converted into 48 rules in this system.

This system was developed in the interviewee's MBA project, the interview was conducted in April 1987, at that time the project was not yet finished so no demonstration could be given. This project was completed four months later, and a diskcopy of the system was sent to the author for running on the PC.

When consulting the system, the operator is required to make his initial choice from among three queries, they are: the fault is, the job includes, and the action is. Each query has its own variables tree. Normally the system is started with the enquiry concerning the fault. Once the fault is found, the operator may proceed to the second query to ascertain the necessary jobs required to rectify the faults. If the operator can really rectify the faults, the third query of the expert system will give the conclusion 'the action is complete', otherwise the answer will be 'the action is to call out the systems engineer'.

V. Reasons for Using a Shell and Choosing a Specific Shell for Developing the System

This project was originally considered for development using the production rule representation and written in Prolog or another

language. This idea was considered because of the interviewee's experience in writing software systems as part of his current job function. However, due to time constraints and also the consideration of the benefit of having a small scale expert system developed by a considerable amount of effort and cost, using an expert system shell for development was finally decided upon.

The main criteria for Company F's selection of a shell were that:

a). it had to be easy to learn and use.

b). it had to work on specific computer hardware that was available to the interviewee, and available as a user system for the operator.

c). it had to be available within the University because the interviewee was conducting the research as his MBA project.

Xi Plus was finally chosen by the company, because it was available at the University and would run on an IBM PC which was available in the company. Also, the shell was relatively cheap which met the company's budget requirements.

VI. The Process of System Building

Having chosen Xi Plus, the interviewee started to learn this system. To the interviewee, it was not too difficult to familiarise himself with Xi Plus because of his computer background based on his current job of functions.

The most important part of the process of system building was the knowledge elicitation. For the project two types of knowledge were obtained. One was the analysis procedure called Fault Tree Analysis; the other was the 'expert' knowledge and experience. The Fault Tree Analysis was conducted with regard to the structure of the system domain. The second type of knowledge was the real 'expert' knowledge which on occasion refered to the expert's many years' experience or expertise. This part of the knowledge was seen as vital to the construction and testing of the expert system.

Three methods were applied for obtaining knowledge from the two experts:

a). Reference to the manuals supplied with the equipment or by direct questioning of the expert.

b). Prioritising the production rules. The expert gave the order of the failures (components) so as to minimise the consultation time.

c). Using the so-called non-field testing to examine the accuracy and priority assigned which would verify whether the rules would reach a similar conclusion to that normally reached by the experts.

VII. Human Response to the System

There were two experts involved in the contribution of knowledge - the interviewee and a systems engineer. The interviewee involved in the development had very high motivation, not only

because this was his MBA project but because he realised the benefits which would be provided to both the company and the systems engineers. However, the other expert was not as enthusiastic as expected. The first reason was a fear that a reduction in manpower would result after the implementation of the expert system. The second reason was a fear of no longer being respected for his expert knowledge and experience. Reluctance to divulge knowledge concerning his expertise based upon these fears was overcome by the interviewee by spending a great deal of time on the construction and specification of questions so as to avoid the possibility of ambiguous answers and reduce the necessity for voluntary information.

VIII. Further Development of the Expert System

Up to now, Company F has not used the expert system in its daily operations, because further expansion of the system is expected.

The present central station computers of Company F is being respecified with more powerful new machines. It is therefore hoped that the expert system could be developed onto these computers with the following anticipated benefits resulting:

a). the reduced need for a separate computer for running the expert system.

b). the functions of multi-user and multi-tasking provided by the new machine enabling multiple and simultaneous consultations.

c). the faster speed.

There will be enhancement of the company's network systems by using more sophisticated equipment. A more advanced and sophisticated expert system will be required other than produced by Xi Plus. A new system using a natural language such as Prolog will be required.

IX. Conclusion

This case study highlights the importance of the skills needed to overcome the difficulties encountered when an expert is reluctant to contribute his knowledge because of the psychological element of fear.

Furthermore, this case study is a good example of building up the initial expert system application in a company, i.e. using a shell for developing a small scale system which is then expected to be expanded in the future by using more advanced and sophisticated techniques. Developing a small system is a stepping-stone for company to act as a training aid and to promote familiarisation in the knowledge of expert systems for further development.

APPENDIX B

A LIST OF EUROPEAN AVAILABLE SHELLS (50)

Name	Supplier	Location
APES	Logic Based Systems	England
Acqusint	Lithp	Netherlands
Adviser	ICL	England
Bewgle	Warm Boot	England
Candi	Battelle Institut	Germany
Cognitif	Cognitech	France
Crystal	Intelligent Environments	England
Diaess	SEL	Germany
ES/P Advisor	Expert Systems Int'l Ltd	England
Envisage	Systems Designers Int'l	England
Epikur	Triumph Adler	Germany
Expert Ease	Intelligent Terminals Ltd	England
Expert Edge	Helix Expert Systems Ltd	England
Expert systems Environment/VM	IBM	
Extran 7	Intelligent Terminals Ltd	Scotland
Frame Engine	Expert Systems International	England
Golem	Microinformatic	France
Hypnotist	Intelligence Products	England
Intelligence Service	Tecsi	France
K.1	Framentec	France
KES II	Software A&E	Arlington, VA
Kiss	Brainware	Germany
L'Experkit	ACT Informatique	France
L'Expert	Mindsoft	France

Ludwig	Triumph Adler	Germany
M.1	Framentec	France
Macexpert	Mindsoft	France
Med 1 and 2	Univ.of Kaiserslautern	Germany
Micro Expert	ISI Ltd	England
Morse	Cril	France
MP-LRO	Cril	France
Nerus	Mindsoft	France
OPS5	DEC	
Ourse	PrologIA	Marseille
Parsec	Aquitaine Systemes	France
PC Plus	Texas Instruments	USA
Rule Master	Intelligent Terminals Ltd	Scotland
S.1	Framentec	France
Sage	Systems Designers Ltd	England
savior	Systems Designers Ltd.	England
Super Expert	Intelligent Terminals Ltd Scotland	England
Superfile ACLS	Southdata	England
Twaice	Nixdorf	Germany
VIE-PCX	Austrian Research for AI	Austria
Xi Plus	Expertech Ltd	England
Xsys	Saia	France

APPENDIX C

KNOWLEDGE BASE FOR

SELECTING A FINANCIAL PLANNING PACKAGE

```
Xi Plus v1.50 C1---------------------------------------- 00:02 1/01/1
              Current Knowledge Base : Selecting a financial planning package
-------------------------------------------------------------------------------

rint kb

uestion 1
 willingness to build is
   yes ,
   no
 question text Are you yourself willing to build the system ?
uestion 2
 sources is
   more than 3 ,
   not more than 3
 question text Does the data come from more than 3 sources ?
uestion 3
 use is
   personal ,
   department ,
   company
 question text What is the level at which it is intended to use the package
           and ( you may select any number of levels ) .
uestion 4
 no. of users in department is
   more than 10 ,
   not more than 10
 question text Please specify the number of users you envisage for the majo
           and system.
uestion 5
 no. of standard reports =
 question text Please specify the number of standard report you estimate yo
           and will produce using the system.
uestion 6
 dimensions is
   2 ,
   3 ,
   more than 3
 question text How many dimensions are to be catered for ? include TIME as
           and a dimension.
uestion 7
 aggregation level is
   one ,
   two ,
   more than two
 question text How many levels of aggregation are envisaged ?
uestion 8
 longevity is
   shortterm ,
   longterm
 question text This question involves the lifetime of the major systems th
           and you envisage building with the package.
           and Please specigy whether the use will be shortterm - i.e. for
           and ad - hoc analyses or lonterm.
```

```
uestion 9
no. of variables is
   less than 50 ,
   not less than 50
question text Approximately how many variables do you expect to have in the
            and model ?
            and Note: Variables are considered to be of the same 'type' if
            and they have different types of attribute.
            and Example: 'Sales of Product A' would be considered to be of the
            and same 'type' as 'Sales of Product B'.
            and Example: 'Sales of Product A' would be considered to be of a
            and different 'type' of 'Cost of Product A'.
uestion 10
 logical model complexity is
   high ,
   not high
 question text Is logical model complexity high or not ?
uestion 11
 maximum cost =
 question text How much is the maximum cost you are willing to pay ? ( expres
umber in
            and Pound e.g. 1000 , 2000 , etc. )
uestion 12
 memory =
 question text Please input the size of your computer memory. ( express in K
.g. 512 ,
            and 640 , 1000 , etc. )
emon 1
   when willingness to build is no
   then report data model structure is standard
emon 2
   when data model structure is standard
   then report class is dedicated
ule 1
   if   sources is more than 3
   then input is complex
ule 2
   if   sources is not more than 3
   then input is simple
ule 3
   if   use is personal
    and use is not department
   then organisation scope is narrow
ule 4
   if   no. of users in department is more than 10
   then department is large
ule 5
   if   no. of users in department is not more than 10
   then department is small
ule 6
   if   use is department
    and department is small
   then organisation scope is medium
```

```
rule 7
    if    use is department
     and department is large
    then organisation scope is wide
rule 8
    if    use is company
     and use is not department
    then organisation scope is wide
rule 9
    if    use is department
     and department is large
     and no. of standard reports > 0
    then output is demanding
rule 10
    if    use is department
     and department is small
     and no. of standard reports >= 25
    then output is demanding
rule 11
    if    use is personal
     and no. of standard reports >= 25
    then output is demanding
rule 12
    if    use is company
     and no. of standard reports >= 25
    then output is demanding
rule 13
    if    output is demanding
     and input is complex
     and organisation scope is wide
     and organisation scope is not medium
     and organisation scope is not narrow
     and longevity is longterm
    then system complexity is high
rule 14
    if    output is demanding
     and input is complex
     and organisation scope is wide
     and organisation scope is not narrow
     and organisation scope is not medium
     and longevity is shortterm
    then system complexity is high
rule 15
    if    output is not demanding
     and input is complex
     and organisation scope is wide
     and organisation scope is not medium
     and organisation scope is not narrow
     and longevity is shortterm
    then system complexity is high
```

```
rule 16
   if   output is not demanding
    and input is complex
    and organisation scope is wide
    and organisation scope is not medium
    and organisation scope is not narrow
    and longevity is longterm
   then system complexity is high
rule 17
   if   output is not demanding
    and input is complex
    and organisation scope is narrow
    and organisation scope is not medium
    and organisation scope is not wide
    and longevity is shortterm
   then system complexity is low
rule 18
   if   output is not demanding
    and input is simple
    and organisation scope is narrow
    and organisation scope is not medium
    and organisation scope is not wide
    and longevity is shortterm
   then system complexity is low
rule 19
   if   output is not demanding
    and input is complex
    and organisation scope is medium
    and organisation scope is not wide
    and organisation scope is not narrow
    and longevity is shortterm
   then system complexity is medium
rule 20
   if   output is not demanding
    and input is complex
    and organisation scope is medium
    and organisation scope is not wide
    and organisation scope is not narrow
    and longevity is longterm
   then system complexity is medium
rule 21
   if   output is demanding
    and input is complex
    and organisation scope is medium
    and organisation scope is not wide
    and organisation scope is not narrow
    and longevity is longterm
   then system complexity is medium
rule 22
   if   output is demanding
    and input is complex
    and organisation scope is medium
    and organisation scope is not wide
    and organisation scope is not narrow
    and longevity is shortterm
   then system complexity is medium
```

```
rule 23
   if   willingness to build is no
   then force data model structure is standard
rule 24
   if   willingness to build is yes
    and dimensions is 2
    and no. of variables is less than 50
    and aggregation level is one or two
   then data model structure is simple
rule 25
   if   willingness to build is yes
    and dimensions is 2
    and no. of variables is less than 50
    and aggregation level is more than two
   then data model structure is medium
rule 26
   if   willingness to build is yes
    and dimensions is 3
    and no. of variables is not less than 50
    and aggregation level is one or two
   then data model structure is medium
rule 27
   if   willingness to build is yes
    and dimensions is more than 3
    and no. of variables is less than 50 or not less than 50
    and aggregation level is one or two or more than two
   then data model structure is complex
rule 28
   if   willingness to build is yes
    and dimensions is 3
    and no. of variables is not less than 50
    and aggregation level is more than two
   then data model structure is complex
rule 29
   if   willingness to build is yes
    and dimensions is 2
    and no. of variables is not less than 50
    and aggregation level is one
   then data model structure is simple
rule 30
   if   willingness to build is yes
    and dimensions is 2
    and no. of variables is not less than 50
    and aggregation level is two
   then data model structure is medium
rule 31
   if   willingness to build is yes
    and dimensions is 2
    and no. of variables is not less than 50
    and aggregation level is more than two
   then data model structure is medium
```

```
rule 32
   if   willingness to build is yes
    and dimensions is 3
    and no. of variables is less than 50
    and aggregation level is one or two
   then data model structure is medium
rule 33
   if   willingness to build is yes
    and dimensions is 3
    and no. of variables is less than 50
    and aggregation level is more than two
   then data model structure is medium
rule 34
   if   data model structure is standard
   then class is dedicated
rule 35
   if   data model structure is simple
    and system complexity is low
    and system complexity is not medium
    and system complexity is not high
    and logical model complexity is not high
   then class is 2 dimensional spreadsheet
rule 36
   if   data model structure is simple
    and system complexity is medium
    and system complexity is not low
    and system complexity is not high
    and logical model complexity is not high
   then class is 2 dimensional advanced language
rule 37
   if   data model structure is simple
    and system complexity is high
    and system complexity is not low
    and system complexity is not medium
    and logical model complexity is not high
   then class is 2 dimensional advanced language
rule 38
   if   data model structure is medium
    and system complexity is low
    and system complexity is not medium
    and system complexity is not high
    and logical model complexity is not high
   then class is 3 dimensional advanced language
rule 39
   if   data model structure is medium
    and system complexity is medium
    and system complexity is not low
    and system complexity is not high
    and logical model complexity is not high
   then class is 3 dimensional advanced language
```

```
rule 40
   if   data model structure is medium
    and system complexity is high
    and system complexity is not low
    and system complexity is not medium
    and logical model complexity is not high
   then class is 3 dimensional advanced language
rule 41
   if   data model structure is complex
    and system complexity is low
    and system complexity is not medium
    and system complexity is not high
    and logical model comolexity is not high
   then class is 3 dimensional advanced language
rule 42
   if   data model structure is complex
    and system complexity is medium
    and system complexity is not low
    and system complexity is not high
    and logical model complexity is not high
   then class is data base
rule 43
   if   data model structure is complex
    and system complexity is high
    and system complexity is not low
    and system complexity is not medium
    and logical model complexity is not high
   then class is data base
rule 44
   if   data model structure is simple
    and system complexity is low
    and system complexity is not medium
    and system complexity is not high
    and logical model complexity is high
   then class is 2 dimensional advanced language
rule 45
   if   data model structure is simple
    and system complexity is medium
    and system complexity is not low
    and system complexity is not high
    and logical model complexity is high
   then class is 2 dimensional advanced language
rule 46
   if   data model structure is simple
    and system complexity is high
    and system complexity is not medium
    and system complexity is not low
    and logical model complexity is high
   then class is 2 dimensional advanced language
rule 47
   if   data model structure is medium
    and system complexity is low
    and system complexity is not medium
    and system complexity is not high
    and logical model complexity is high
   then class is 2 dimensional advanced language
```

```
e 48
if   data model structure is medium
 and system complexity is medium
 and system complexity is not low
 and system complexity is not high
 and logical model complexity is high
then class is 2 dimensional advanced language
e 49
if   data model structure is medium
 and system complexity is high
 and system complexity is not medium
 and system complexity is not low
 and logical model complexity is high
then class is 2 dimensional advanced language
e 50
if   data model structure is complex
 and system complexity is low
 and system complexity is not medium
 and system complexity is not high
 and logical model complexity is high
then class is 3 dimensional advanced language
e 51
if   data model structure is complex
 and system complexity is medium
 and system complexity is not low
 and system complexity is not high
 and logical model complexity is high
then class is data base
e 52
if   data model structure is complex
 and system complexity is high
 and system complexity is not low
 and system complexity is not medium
 and logical model complexity is high
then class is data base
e 53
if   class is dedicated
then decision is planalyst
 and report decision is planalyst
e 54
if   class is 2 dimensional spreadsheet
 and maximum cost < 3198
 and memory < 384
then decision is bottomlinev
 and report decision is bottomlinev
 and report detailed information for bottomlinev is available from DBase III

e 55
if   class is 2 dimensional spreadsheet
 and maximum cost < 3198
 and memory >= 384
then decision is oxcalc
 and report decision is oxcalc
 and report detailed information for oxcalc is available from DBase III +
```

```
e 56
if   class is 2 dimensional spreadsheet
 and maximum cost >= 5170 >=6000

then decision is 20 / 20
 and report decision is 20 / 20
 and report detailed information for 20 / 20 is available from DBase III +
e 57
if   class is 3 dimensional spreadsheet


then decision is reportmanager
 and report decision is reportmanager
e 58
if   class is 2 dimensional advanced language
 and maximum cost <1000
 and memory < 448
then decision is mastermodeller
 and report decision is mastermodeller
 and report detailed information for mastermodeller is available from DBase
     III +
e 59
if   class is 2 dimensional advanced language
 and 1000 < maximum cost < 1375
 and memory >= 448
 and memory < 576
then decision is demon
 and report decision is demon
 and report detailed information for demon is available from DBase III +
e 60
if   class is data base
 and maximum cost >1200
 and memory < 576
then decision is demon
 and report decision is demon
 and report detailed information for demon is available from DBase III +
e 61
if   class is 2 dimensional advanced language
 and maximum cost >1375
 and memory >= 576
 and memory < 1320
then decision is pcexpress
 and report decision is pcexpress
 and report detailed information for pcexpress is available from DBase III +
```

```
le 62
 if    class is 2 dimensional advanced language
  and maximum cost > .13000
  and memory >= ~~1320~~ 2000
 then decision is express
  and report decision is express
  and report detailed information for express is available from DBase III +
le 63                                    rule 64
 if    class is data base                  if class is data base
  and maximum cost > 1375                  and 1100 < maximum cost < 1375
  and memory >= 576                        and memory >= 576
 then decision is pcexpress                then decision is demon
  and report decision is pcexpress
  and report detailed information for pcexpress is available from DBase III +
ery 1
class
ery 2
decision
```

APPENDIX D

A FORM OF QUESTIONNAIRE

FOR FINANCIAL PLANNING PACKAGES

SELECTING FINANCIAL PLANNING PACKAGES

Package Name XXXXXXXXXXXXXXXXXX

1. Vendor organisation:
   Name XXXXXXXXXXXXXXXXXXXXXX
   Address XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   Postal Code XXXXXXXX Telephone XXXXXXXXXXXX
   Person to Contact XXXXXXXXXXXXXX
   Turnover(last financial year) XXXX
   Turnover for Parent Company(if any) XXXX
   Year when Organisation set up XXXX

2. Organisation Owning the Package(if different from the above)
   Name XXXXXXXXXXXXXXXXXXXXXXX
   Address XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   Postal Code XXXXXXXX Telephone XXXXXXXXXXXX
   Turnover of Owner(last Financial Year) XXXX

3. Number of Installed Systems
   Total Number of installed systems--UK XXXX
   Total Number of installed systems--Worldwide XXXXX
   Please specify the target market for your productXXXX

4. Overall type of Package(Please tick for yes) abcdef
   a. Dedecated b.2 Dimensional Spreadsheet c.3 Dimensional SpreadsheetXXXXXX
   d. Advanced La guage e. Database f. Resource-based
   Other(Please specify) XXXXXXXXXX

5. Minimum Hardware Requirements:(Pls put 0 if a facility is not required)
   Memory Size XXXXX (Kilobytes)
   Hard Disk XXXXX (Megabytes)
   Diskette XXXXX (Kilobytes)
   Please specify any special requirements not found on 'standard' computers and any combinations of memory, hard disk and diskette that are available
   XXXX

6. Operating System: (Pls tick whichever systems may be used)

Microsystems:	a.MS-DOS b.PC-DOS c.UNIX d.AIX	abcdef
	e.CP/M f.Concurrent CP/M	XXXXXX
	Others (Please specify) XXXX	

Mainframe and Minisystems (Please specify the operating systems that will support your product) XXXX

7. Package Costs: (It is recognised that the price structure for some software is complex. If this should be the situation for your product, would you please append your current price list to this questionnaire. If you quote for specific situations, please write this next to each item).

Purchase Price	XXXXXXXXXXXX
Annual Maintenance Charge	XXXXX
Charge for Enhancements	XXXXX

If training is mandatory please specify the additional cost and condition XXXX

8. Accompaniments to the Basic Software: (Pls tick A if available in the normal purchase price. Pls tick B if available at extra cost, and specify this cost)

	A	B	
Manual	X	X	X
Tutorial	X	X	X
Tutorial Diskette	X	X	X
Demomstration Models (other than for dealers)	X	X	X

9. Pre-sales Service: (Pls tick A if available free of charge. Pls tick B if availability depends on the prospect)

	A	B
Demonstration at Your Premises	X	X
Demonstration at the Clients Premises	X	X
Trial Version left with Client	X	X
Do you Write Benchmarks?	X	X

(If you charge for any of the above or offer any other form of pre-sales service, would you please specify in the space provided below).
XXXX

10. After-sales Service: (Pls tick the services that you offer or those offered by third parties whom you would be willing to recommend).

	Offered by Yourselves	Offered by others
Basic Training	X	X
Advanced Training	X	X
Technical Support (other than by hot line)	X	X
Consultancy	X	X
Hot Line	X	X
User Group	X	X
Newsletter	X	X

11. Size and Capacity of the package:

Maximum No. of Cells	XXXXXXXXXX
Maximum No. of Variables	XXXXXXXXXX
Maximum No. of Dimensions	XXXXXXXXXX
Maximum No. of Files	XXXXXXXXXX
Use of a 'sparse' matrix	XXXXXXXXXX

12. Inputs and Creating Input Formats: (Pls tick if available)

a. Data validation	b. Menu creation facility	c. Screen painting	abcdef
Data Modelling	d. Extrapolation	e. Interpolation	f. Spreading	XXXXXX

13. Outputs and Creating Output Formats: (Pls tick if available)

Report generator		X
sign		X
Files	ASCII	X
	DIF	X
	PRN	X
Graphics	Pie charts	X
	Bar charts	X
	Historgrams	X
	Star diagrams	X
	Graphs	X
	others (Please specify)	XXXXXXXXXXXXXXXXXXX

14. Analyses Available(Please tick those facilities that are available)

Sensitivity Analysis(other than simply altering a data value)	X
Goal Seeking(Backward Iteration)	X
Risk Analysis	X

15. Calculation: (Please tick those facilities that are available)

Time and Date arithmetic	X
Matrix calculations	X
Solution of Simultations Equations	X
'Re-entrant'	X
Sorting	X
Look-up Tables	X
Simple Statics	
Mean	X
Variance	X
Standard deviation	X
Linear regression	X
Financial functions	
NPV	X
IRR	X
Amortisation	X

16. Macro/Exec/Command File Language: (Please tick those facilities that are applicable. If your package does not offer a macro/exec/command file' facility, please indicate this by writing N/A across this question.)

Are the macro and modelling languages one and the same?	X
Is the macro language an augmented version of the modelling language?	X
Is the macro language a different language from the modelling language?	X
Is 'learn' mode availavle?	X

17. Debugging aids:(Please tick those facilities that are available)

Full printout of logic	X
User-defined cell names(for spreadsheets only)	X
Ability to use comments	X
English-like modelling language	X
Search and list facility	X
Search and replace facility	X
Trace facility	X
Full screen editing	X

18. Security: (Please tick those facilities that are available)

Passwords	X
'Read only' sections	X
Protection of areas of data	X
Ability to hide data	X

19. Linkages to other Software: (Please specify the packages that your package has been linked to by your present clients. Please state whether any of the facilities listed below form an integral part of your package).

Database	XXXX
Spreadsheets	XXXX
Wordprocessors	XXXX
Graphics	XXXX

20. Other Facilities: (Please tick facilities offered by your package)

Does the pacjage support networking?	XXXXX
How many users can use the package at any one time?	XXXXX
Can hierarchies be defined?	XXXXX
Has the package a curve-fitting facility?	XXXXX
Please specify how consolidation would most easily be carried out using your package	XXXX
Can 'Downward Consolidation' be carried out automatically?	XXXXX

21. Other Issues:

It is recognised that the 20 sets of questions posed above are unlikely to have done complete justice to your financial planning package. Thus we would be grateful if you would list below any important features of your software that haven't been covered. In particular, we would like to know of non-financial planning features. One example is the inclusion of a project scheduling facility within a spreadsheet.

XXXX

NOTES ON THE QUESTIONNAIRE

1. Contact Person: It may be necessary to contact someone in the organisation to clarify entries in the questionnaire or otherwise seek further information. It would be helpful for us to have a name to contact in this eventuality.
2. Turnover: Turnover is asked for as an easily understandable measure of the financial standing of the package producer. However, it is realised that such figures may be considered sensitive information. If this is your situation then please omit the questions where this information is asked for.
3. Types of Financial Planning package: The types of financial planning package have been categorised as follows:
   Dedicated--These are packages that offer completely or almost completely defined sets of routines covering large areas of financial activity. The user does not have to worry thinking about the model logic or of the report specification,as these are predefined: the only concern for the user is to insert data in response to package-generated commands.
   Spreadsheets--These are packages in which the logic is 'cell-specific',ie.

   where the logic applies to one cell only.
   Advanced Language--These are packages where the logic is separate from the data and where the logic is applicable to all relevant time periods. Generally the logic is written in English.
   Database--This type of package is not to be confused with database packages such as DBase III. Database financial planning packages are financial planning software based on database lines. They do not demand a direct link between the format of inputs and outputs that are implicit in all other forms of financial planning package. At input the concern is with the basic 'building blocks' without any real concern for the outputs required. Any required output can easily be created afterwards. Consequently, any number of different biews of the data can easily be made available from the same set of input data.
   Resource-based--All types of financial planning package can be used to assist in the planning of non financial resources. However, resource-based packages offer a particularly easy link between the financial side of a business and the use of resources that underpin the financial outcomes. Separate modelling facilities are available to model costs and physical activities.

4. Printout of the Logic A major aid in debugging software is to be able to obtain a printout of the logic of the model. Most packages allow for this, but not all do so in a way that is useful. What is needed is to be able to do one of the following:
   a) list the logic in English ie. REVENUE = SALES * PRICE
   b) for spreadsheet packages, to list out a variable and directly opposite it the logic by which it is to be calculated,ie.

	A	B
1	SALES	20
2	PRICE	5
3	REVENUE	B1 * B2

In the questionnaire,'full printout of logic' means the capability to achieve either a) or b) above.

5. Re-entrant Software Concurrent users of multiuser software can experience a marked degredation in performance as more and more users become connected. Part of the problem may lie in the need to provide a separate copy of the software for each individual user. Re-entrant software gets over this problem by holding only one copy of the software in memory, accessible by all users. This capability obviously reduces the overall memory utilisation.

6. 'Downward' Consolidation By downward consolidation is meant the passing of data from a higher level to lower level,when the value of the data passed is dependent on the lower values previously passed up the hierarchy. An example should make this clear. Suppose a data processing department in a company is used by several other departments. It may be that the costs for the DP department are to be allocated to the other departments in proportion to the number of people in those departments. To calculate the actual value of the apportioned cost, the total numbers of employees would need to be calculated by adding together the numbers in each department at company level, and then pro-rating the DP costs. These pro-rated costs would then need to be passed down and included in the costs for each department.

A>