

TR 88 - 13

**EXPLANATION IN RULE-BASED
EXPERT SYSTEMS**

submitted in partial fulfilment of the requirements for the degree

MASTER OF SCIENCE
of Rhodes University

by

Kenneth John Carden

January 1988

Table of Contents

1. Introduction	1
1.1. Expert systems research at Rhodes University	1
1.2. This project	2
2. A brief overview of expert systems	5
2.1. The components of an expert system	5
2.1.1. The knowledge base	6
2.1.1.1. Constructing knowledge bases	6
2.1.1.2. Uncertainty	7
2.1.2. The inference engine	7
2.1.3. The user interface	7
2.2. Expert system building tools	8
2.3. Literature	9
3. The P.R.O. Expert System Toolkit	10
3.1. Introduction	10
3.2. Knowledge representation	10
3.3. Inference engine	11
3.4. Uncertainty	11
3.5. User interface	12
3.6. Systems developed using P.R.O.	12
4. The explanatory needs of expert systems users	14
4.1. Introduction	14
4.2. The results of other research	14
4.3. Local case studies	15
4.4. Explanation sub-system guidelines	17

5. Explanation-generating facilities - research issues	18
5.1. Introduction	18
5.2. Mycin	18
5.3. Rule justification	19
5.4. Analytical routines	20
5.5. Natural language, graphics and windowing	21
5.6. Explanatory features in small expert system shells	22
5.7. Conclusion	24
6. The P.R.O. Explanation sub-system	25
6.1. Introduction	25
6.2. The rule trace	26
6.2.1. Strategic information	28
6.2.2. Certainty combination	28
6.2.3. Natural language, graphics and windowing	29
6.2.4. Viewing the trace	29
6.3. The sensitive question identifier (SQI)	31
6.4. Summary	34
7. Applications	35
7.1. FISHFARMER	35
7.2. INVEST	37
8. Computational aspects	38
8.1. Introduction	38
8.2. Input to the explanatory routines	38
8.2.1. Structures representing rule justification	39
8.2.2. The result predicate	40
8.2.3. The rule1cres and rule2cval predicates	40
8.3. Combining these structures into an explanation-generating facility	41
8.3.1. The rule trace	41
8.3.2. The routines of the sensitive question identifier (SQI)	41
8.4. The P.R.O. Expert System Toolkit	42
8.4.1. The development of an expert system using the toolkit	42
8.4.2. Extensions to the toolkit	43

9. Conclusions	44
----------------------	----

Bibliography

Appendix A

An example questionnaire

Appendix B

The INVEST demonstration expert system

List of Figures

2.1. Major Components of an expert system	5
6.1. Example rule base	26
6.2. A basic rule trace through the example rule base	27
6.3. A full rule trace through the example rule base	30
6.4. A graphical representation of the rule trace	31
6.5. Listing produced when a goal is proved untrue	32
6.6. Questions used, their numeric effect and sensitivity	33
6.7. Listings produced by routines of the SQI	33
8.1. The modified checkres and checkvalue condition predicates	40
8.2. The result database predicate	40
8.3. The rule1cres and rule2cval database predicates	41

Acknowledgements

I would like to express my appreciation to the supervisor of this project, Professor Denis Riordan, for his enthusiastic guidance and for reviewing earlier drafts of this documentation.

I would further like to express my thanks to those staff and students of the Departments of Computer Science, and Ichthyology and Fisheries Science who contributed in various ways in the project. I am particularly indebted to Nick Dean, John Bradshaw and Helen Purchase.

My final thanks go to my friends, especially Kirsty, for the many forms of assistance rendered during the year.

Abstract

The ability of an expert system to explain its reasoning is fundamental to the system's credibility. Explanations become even more vital in systems which use methods of uncertainty propagation.

The research documented here describes the development of an explanation sub-system which interfaces with the P.R.O. Expert System Toolkit. This toolkit has been used in the development of three small ecological expert systems.

This project has involved adapting the results of research in the field of explanation-generation, to the requirements of the ecologist users. The sub-system contains two major components. The first lists the rules that fired during a consultation. The second component comprises routines responsible for quantifying the effects on the system conclusions of the answers given to questions. These latter routines can be used to perform sensitivity analyses on the answers given. The incorporation of such routines in small expert systems is quite unique.

1. Introduction

This project forms part of ongoing research into the development of small expert systems at Rhodes University. This research has proceeded largely in co-operation with ecologists who need micro-computer based expert systems to assist in decision-making tasks that they are involved in.

1.1. Expert systems research at Rhodes University

Two expert systems have been completed during the course of this research [DEA87, O'K87], and the development of a further system [PUR87] is at an advanced stage. A review of the progress made during this research is given in [RIO87].

These expert systems, although all of an ecological nature, have had some distinctly differing requirements. Development of the respective systems has, for this reason, commenced with studies of the available expert system shells. Such studies have highlighted both the difficulties and useful features of the shells in relation to the intended applications. A general conclusion was that the expert system shells studied provided too rigid a framework for the development of the required systems [RIO87]. It thus appeared from an early stage that much could be gained from investigating the expert system toolkit approach.

An expert system toolkit has been defined, [JAC84], as containing: a set of software modules which perform well defined functions in the construction of a knowledge-based system, interfaces between these modules, and guidelines on the use of these modules. Toolkits can thus offer increased flexibility over the more traditional expert system shells.

Research has, for this reason, concentrated on the development of a set of micro-computer based expert system building tools, collectively known as the P.R.O. Expert System Toolkit.

The development of the toolkit has proceeded in parallel with progress on the ecological knowledge bases [DEA87, O'K86, PUR87]. Requirements of the respective systems, which could not immediately be met by the toolkit, were identified during the course of this work.

1.2. This project

Of the requisite features, one of the most pressing was the need for appropriate explanations into the actions of the system to be generated during consultations. This requirement was made vital to system acceptance due to the incorporation of uncertainty measures in the domains of these expert systems.

The need to model uncertain information in a great deal of expert system domains has led to the advancement of various methods for representing this information [BUC84, LES85, NIL86, WHI84]. Most methods employed, including that used by the P.R.O. Expert System Toolkit [BRA86a], rely on users specifying their degree of uncertainty about the factual information which is either entered in the knowledge base, or volunteered during consultations. The entry of uncertainty measures leads, naturally, to user subjectivity being introduced into, and hence propagated through, the knowledge base.

Ecological domains are often characterized by uncertain, or even unavailable, information. Expert systems, which provide heuristic methods for reasoning with such information, are increasingly being used by ecologists in preference to the more traditional numerical methods [STA84, BRA86b]. Numerical methods are typically more rigid in their requirements for good data.

The incorporation of methods for the propagation of uncertainty in expert systems has introduced a further dimension of complexity to the output of such systems. It becomes imperative, under these circumstances, that the user is assured of the validity of the output. This assurance can be provided by an efficient explanation-generating facility which can show how deductions were made and how uncertainty measures were utilized. Such a facility could assist the user further if it could direct him to areas where accumulated uncertainty has led to doubtful deductions being made. It would further be useful if the facility could then allow him to analyse the sensitivity of the model to changes in the values of variables in these areas.

The topic of explanation in expert systems is one which has received attention only fairly recently. It remains an area where there is much research currently being conducted. The recency of the interest in explanation-generation emphasizes that the need for such facilities only becomes apparent once an expert system has reached advanced stages of development.

There are as yet few standards in explanation facilities, and there is therefore much scope for initiative in this area. There has, for instance, been very little research on methods for performing sensitivity analyses on the model parameters.

The aim of this particular project was, therefore, to recommend and develop explanation-generating facilities for use in ecological expert systems produced using the P.R.O. Expert System Toolkit. The explanation-generating facility was to have, as a major component, routines designed to assist the user in determining the sensitivity of the model to changes made in variables identified as being critical to the solution.

To accomplish this, it was necessary to conduct thorough investigations into the following areas :

- The explanatory features needed in the ecological systems being developed using the P.R.O. Expert System Toolkit.
- Past and current research in the area of explanation-generating facilities.
- Relevant explanation-yielding facilities exhibited by the more traditional decision support systems.

In summary, the project consisted of adapting the results of recent research in the field of explanation-generation, to satisfy the requirements of local ecologists using small expert systems.

The routines developed during this project are collectively referred to as an explanation sub-system, and form part of one of the modules of the P.R.O. Expert System Toolkit.

As part of this project, the explanation sub-system has been interfaced with the FISHFARMER ecological expert system [BRU86, DEA87]. This system is designed to advise on suitable species, sites and culture methods to be used in fish farming.

A smaller financial expert system, INVEST, has also been developed to demonstrate the features offered by the P.R.O. Expert System Toolkit.

Although the routines comprising the explanation sub-system have been developed to interface with the P.R.O. Expert System Toolkit only, the ideas advanced can be applied to expert systems in general.

The research undertaken is documented as follows:

Chapters 2 and 3 provide the background necessary for the subsequent chapters. Chapter 2 is an overview of expert systems, and introduces the explanation-generating facility as part of the user interface of an expert system. In chapter 3 the P.R.O. toolkit is described.

Chapters 4 and 5 describe the main inputs to this research. In chapter 4, the explanatory needs of expert systems users, in general, and the users of P.R.O. toolkit based systems, in particular, are described, while in chapter 5 a summary of current research into explanation-generating facilities is given. The explanation-generating facilities of commercially available expert system shells are also described in this chapter.

Chapter 6 provides a detailed description of the explanation sub-system developed during this research. Chapter 7 is a description of the applications of the sub-system. These applications include the FISHFARMER system and the INVEST demonstration expert system. The INVEST system, a copy of which is included with this documentation, is a fully comprehensive demonstration system designed to give potential users of the P.R.O. Expert System Toolkit a complete, and quick, step-by-step guide through all the facilities offered.

Chapter 8 is a brief discussion of the computational methods employed in the implementation of the sub-system. The chapter concludes with some details on the operation of the P.R.O. Expert System Toolkit. Chapter 9 concludes this documentation.

Appendix A consists of the questionnaire used to determine the explanatory needs of the users of P.R.O. based systems. Appendix B contains some notes on the use of the INVEST system.

2. A brief overview of expert systems

This chapter is not intended to be a complete overview of expert systems. It serves, rather, to introduce the issues required for the development of the explanation sub-system. The chapter ends with a summary of useful literature.

2.1. The components of an expert system

Professor Edward Feigenbaum of Stanford University, one of the most eminent researchers in expert systems and a contributor in some of the best known expert systems projects, has given a much quoted definition of an expert system as:

"... an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. The knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of the expertise of the best practitioners in the field" [HAR85].

The major components of such an expert system are diagrammatically displayed in Fig. 2.1.

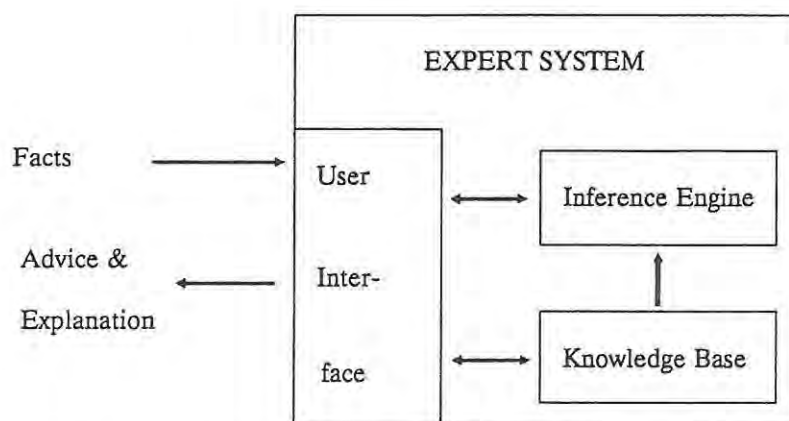


FIG. 2.1 Major components of an expert system. Arrows indicate the flow of information, (From BUC84).

The heart of the system is the **inference engine** which operates on knowledge, represented in the **knowledge base**, and further facts supplied by the user to deduce new knowledge or facts.

2.1.1. The knowledge base

A number of methods for representing knowledge have been developed [JAC86]. The most important of these techniques are semantic networks [ALT84], frame based systems [JAC86] and production rules [BUC84]. Production rules, which gain their power from their easy "If ... Then ..." format, are a popular knowledge representation method. A typical production rule can be written as follows:

*If A and,
B and,
C,
Then conclude D.*

where A, B, C, and D are facts. The full rule can be read as "If A and B and C are known, then a value for D can be derived". This natural rule format facilitates the representation of human problem solving steps. The rest of this research focuses on production rule-based systems.

In a production rule-based system, the knowledge base typically contains all the rules and might contain a number of facts. Further facts needed by the inference engine to make deductions must be asked of the user, or derived from external data-bases. These facts are obviously the variables in the model.

2.1.1.1. Constructing knowledge bases

The process of eliciting the rules from domain experts is known as knowledge acquisition.

Knowledge acquisition most commonly involves "asking the experts to report on knowledge that he/she can directly articulate" [OLS87]. In such direct knowledge acquisition, the services of a knowledge engineer, a specialist in the field of developing knowledge bases, are usually required. The knowledge engineer will most often set up meetings, interviews and develop questionnaires to elicit the expert's knowledge. Knowledge base development then proceeds with the implementation and subsequent demonstration to the experts of a prototype system. Feedback from the experts then results in this process being repeated until both experts and

knowledge engineer are content with the knowledge base.

Various other, **indirect**, methods of knowledge acquisition have been proposed to facilitate knowledge base development. Most of these involve the use of sophisticated mathematically-based software products. Reviews of these techniques are given in [OLS87, BER87b].

2.1.1.2. Uncertainty

Due to the fact that knowledge is often imprecise or uncertain, it becomes necessary to specify the degree of certainty about the knowledge given in the production rules. Various methods for incorporating certainty measures in production rules have been developed [BUC84, LES85, NIL86, WHI84]. Most of these methods involve the knowledge engineer and/or the user specifying certainty measures with the rules and facts. These certainty measures, typically expressed as percentages or numbers in the range 0 to 10, are meant to represent the degree of confidence that the knowledge engineer, or user, has in a particular piece of information. As an example consider the above production rule, with which the knowledge engineer might attach a certainty measure of 60%. This implies that if A, B and C are known, then the value for D can be derived and one can be 60% sure that it is correct. Furthermore, the user of the system, when asked in a consultation to give a value for A, might attach a confidence of 80% to his answer, specifying that he is 80% sure that the answer given is correct.

2.1.2. The inference engine

There are usually one or more system goals or hypotheses stipulated in the rules of the knowledge base, the truth of which the inference engine must attempt to verify. Coded within the inference engine are the routines which define how the system should derive values for these hypotheses from the rules, and how the combination of certainty measures is to be achieved.

2.1.3. The user interface

The user interface controls all dialogue with the user of the system. As indicated in Fig. 2.1, this interface is used both to ascertain facts from the user, through questioning, as well as to give advice on system results.

"It (the user interface) can be thought of as a post-box through which messages are sent by both sides" [BRA86a]. Due to the high degree of interaction between the program and user, there is a need for this interface to be sophisticated.

The role of an explanation-generating facility is to make the reasoning, as regards this advice, transparent to the user of the system. Explanation-generating facilities are thus an essential component of the user interface.

2.2. Expert system building tools

Common reasoning strategies can often be applied to many different expert system applications. Expert system building tools, which offer varying degrees of assistance in expert systems development, have thus evolved.

An expert system shell can conceptually be thought of as an expert system (see Fig. 2.1) without a knowledge base. Implementing a full system then only involves the selection of an appropriate shell and the development of the knowledge base. Expert system shells are designed to be used with knowledge bases developed by users in varying fields, but are often found to be more restrictive in this regard [BAR84, JAC86].

An Expert System Toolkit can be viewed as a collection of modules that can be combined in a variety of ways for the construction of expert systems [HAM86]. Modules are often smaller entities than the components described above (Fig. 2.1). Most toolkits allow modifications to be made to the individual modules. Guidelines are provided on how modifications can be achieved [HAM86]. Toolkits thus offer more flexibility than expert system shells.

The most flexible expert system developing environments are presented by Artificial Intelligence languages such as PROLOG, LISP and derivatives thereof. Such languages supply none of the components illustrated in Fig. 2.1, but offer flexible environments for their development.

2.3. Literature

Comprehensive expert system introductory texts are available [ALT84, JAC86, WAT86]. These texts include descriptions of a few of the systems which have become well known for being innovative in various areas of the field. [HAR85] provides a good review of expert systems with particular reference to business applications.

Possibly the best known expert system, MYCIN, was developed at Stanford University to assist physicians with advice on diagnosing and treating infectious diseases. Work on the project itself proceeded for many years and involved a number of now well known researchers in the field. The MYCIN project provided a great deal of the fundamentals for separate research and development. Most of the concepts developed in the P.R.O. Expert System Toolkit have their origins in this project. An extensive series of papers on the MYCIN and related projects is given by Buchanan and Shortliffe [BUC84].

A number of texts including [BAR84, JAC86, WAL86] provide comprehensive reviews of expert system building tools.

Two journals, Expert Systems and IEEE Expert, have appeared fairly recently. These journals concentrate on a wide range of topics, and much of their contents describe applications of expert systems in Britain and the U.S.A. respectively. The journal, Artificial Intelligence, documents much of the detailed research in the field of expert systems, and is more technical than the above two publications. A large amount of research conducted into user issues in expert systems, including the generation of explanations, is documented in the International Journal of Man Machine Studies.

Finally, much current research is initially presented in the proceedings of conferences or interest groups. The British Computer Societies Special Interest Group, for instance, holds annual conferences, the proceedings of which are made available [ES83, BRA85, MER85].

3. The P.R.O. Expert System Toolkit

This chapter is a brief summary given to introduce the P.R.O. Expert System Toolkit, and to highlight recent developments. For further detail consult [BRA86a, BRA86b, PUR87, RIO87].

3.1. Introduction

Bradshaw [BRA86b] recognized that P.R.O. offered greater flexibility than most commercially available small expert system shells, but was hesitant to refer to the system as a toolkit. With recent developments it has now become apparent that the system fulfills the criteria of a toolkit [RIO87].

The P.R.O. Expert System Toolkit currently offers a collection of modules which can be combined and modified as needed for any particular application. The main modules of the toolkit are the inference engine, the question-answering and explanation-generating facilities of the user interface, and the knowledge base. Subsections within these modules, such as that controlling uncertainty combination in the inference engine, may be easily modified. Additional components of the toolkit are described in [RIO87].

The P.R.O. Expert System Toolkit has been used in the development of three systems [DEA87, O'K86, PUR87], two of which are complete. These three systems make use of different combinations of toolkit modules.

3.2. Knowledge representation

The P.R.O. Expert System Toolkit makes use of production rules for knowledge representation. Individual rules are specified as Prolog clauses. Rules, therefore, follow the "*Conclude ... If ...*" Prolog rule format [CLO81], rather than the "*If ... Then ...*" format described above. The two formats are otherwise similar.

The conclusion section, or head, of each rule defines the value that can be associated with a variable if the conditions specified in the rule body are met. Each condition refers to a variable and a comparison that must be performed on that variable before the condition can succeed.

A rule can be defined to be a system goal, also called an hypothesis, making it a rule which will yield a final answer.

3.3. Inference engine

The inference mechanism is goal-directed and backward chaining. At the start of a consultation, the first system goal is chosen for evaluation. In attempting to derive a value for this goal, the conditions are evaluated separately. If the value of the variable designated in the condition is unknown, then the system checks whether the variable is specified in the body of a rule or by a question. In the former case, control then passes to the evaluation of this rule. This process is known as backward chaining. Backward chaining continues until the variable specified in a condition is represented by a question, which is asked of the user. The relevant conditions can then be evaluated, and in this manner the truth of the goal ascertained. The system moves on to subsequent goals if the current goal is not shown to be true.

There are two types of questions, character (yes/no/don't know) and numeric. The syntax of the conditions varies slightly according to the type of variable specified.

3.4. Uncertainty

The basic method of uncertainty specification and propagation through the rule network, was designed to be simple and computationally inexpensive. It was furthermore designed to be natural to the ecologist users.

Certainty factors, representing the degree of belief in a piece of information, are entered by the user in response to system questions. Additional measures, representing the importance of conditions and the strength of rules, are defined in the production rules by the knowledge engineer.

Certainty factors are then propagated through the network by computing condition contribution factors for each condition in a rule. These contribution factors are obtained by multiplying the certainty factor of the variable, (as obtained from the user) by the importance of the condition. The sum of these condition contribution factors multiplied by the strength

of the rule, is the certainty factor for the variable defined in the rule body. This certainty factor is then used in rules higher in the network, or returned as the goal certainty factor.

3.5. User interface

The routines of the user interface are responsible for all contact with the user. The routines control question phrasing and answering, and are responsible for explanation-generation.

The development of the explanation-generating facilities is described in the following chapters.

3.6. Systems developed using P.R.O.

The P.R.O. Expert System Toolkit was selected for the development of three small ecological expert systems, as it represented the most flexible expert system building tool available.

Direct knowledge acquisition techniques (described in section 2.1.1.1.) were used in the development of each knowledge base. These techniques were successful as there was close contact between experts and knowledge engineers during each project.

The River Conservation System, RCS, [O'K86] is used to determine the conservation status of rivers in Southern Africa. The system makes use of a variant of the usual P.R.O. inference engine, which has been designed to produce an index score (in the range 0 - 100) indicating the conservation status of the relevant river. The system has been in use on a trial basis for approximately one year. Requests for system modifications are currently being formulated.

The FISHFARMER aquaculture expert system [BRU86, DEA87], designed to advise on suitable species, sites and culture methods for aquaculture, or fish farming, was completed during the period of this research. The completed system has knowledge on fourteen fish species contained within approximately 80 production rules.

FISHFARMER underwent much testing by experts and potential users during its final stages of development. The system was most recently demonstrated at the International Alternative Life History Styles conference [ALH87]. FISHFARMER was well received, and it is believed to produce good results. The system is described in more detail in chapter 7 where the author's involvement is described.

Work on the PISCES expert system [PUR87] has reached an advanced stage of development although much of the detail in the knowledge base has yet to be decided upon. The system is designed to assist ecologists and legislators in deciding on the merits of introducing alien fish species into local waters. PISCES was also demonstrated at the International Alternative Life History Styles conference [ALH87].

The PISCES system has had various subsections of the inference engine, including that which controls uncertainty, modified.

4. The explanatory needs of expert system users

4.1. Introduction

As noted in chapter 2, all expert systems offer advice. The ability of systems to explain their reasoning, concerning this advice, is fundamental to their credibility. Explanation-generating facilities were not recognised as being of major importance in the early expert systems [CLA83]. Rather, explanatory routines were merely added to the end of existing systems as a final touch. An appreciable amount of research has recently been conducted into the provision of effective explanations. Much of this research has been aimed specifically at users, in an attempt to infer how they expect the results of any decision support system to be presented and justified. There is as yet, however, no well-articulated theory on what advice and explanation is required of such systems.

It is therefore necessary to conduct surveys aimed at ascertaining what explanatory features users require of expert systems. The following two sections describe the results of such surveys conducted by other researchers as well as those conducted during this research.

4.2. The results of other research

In attempting to build a theory on the type of advice and explanation that expert systems need to provide, researchers have studied naturally occurring consultations between human experts and "users" [KID85b]. In such consultations the human expert is expected to give correct answers, and to explain how these answers were arrived at, in a way that the "user" will understand.

Over the period of approximately one year, Kidd recorded phone-in consultations, where experts in domains such as electronic equipment trouble-shooting, health, cooking and spring cleaning were consulted by callers with particular problems. Kidd found that the "users" most often approached consultations with their own intentions and expectations. They sought firstly to verify the validity of these expectations. Once this was achieved, they were interested in having the advice of the consultation explained and supported so that they could leave with a fuller understanding of the relationships that existed in the problem.

Further studies have focused on determining what assistance users require of expert systems which lack adequate explanatory facilities [CLA83, CLA84]. These studies have shown that it is essential to provide justifications of the information contained in the rules of the knowledge base. This subject is covered in greater depth in chapters 5 and 6.

4.3. Local case studies

The three systems developed using P.R.O. became valuable case studies for local analyses of what type of explanation-generating facilities users request. A very wide spectrum of experts, users and potential users could be surveyed, as the three systems were at different stages of development.

User opinion, as regards explanation-generating facilities, was ascertained through :

- Contact with the user of the completed RCS system.
- Regular meeting during the development of the FISHFARMER and PISCES systems.
- The use of demonstration expert systems at conferences and meetings [ALH87].
- Questionnaires given out at conferences and sent to users (an example questionnaire is given in Appendix A).

The expert responsible for the development of the RCS system, who is also the main user and who remains in contact with all other users, has gained valuable experience on how the system has performed. He was, therefore, questioned directly. Explanatory facilities were found to be non-vital to the functioning of the RCS system. The expert could, however, provide insight into the sorts of problems that users have with the advice given by expert systems.

The author was involved in assisting the knowledge engineer of the FISHFARMER system with development of the knowledge base. A version of the system, without the explanation sub-system added, was demonstrated to potential users and experts [ALH87]. Their thoughts regarding explanatory facilities were noted through the use of questionnaires. The knowledge engineer assisted further in appraising the explanation-generating facilities during various

stages of their development.

The author was also involved in the design and initial review phases of the PISCES system. Meetings of all parties involved were held on a weekly basis during these phases. These meetings provided much of the input during the formative stages of the explanation sub-system development. It was particularly useful to have a system that was in its design stages for a case study, as ideas about how it should function were very varied.

The review phase, which included demonstrations of the system to experts and potential users, provided much input to subsequent stages of sub-system development.

Results of these meetings, questionnaires and demonstrations revealed that it would be valuable if post-consultatory explanation were available for analysing system results. Such explanatory information was needed, it was found, to show exactly how a system goal was validated, and which the important factors, or variables, were. This corresponds with the deductions of Kidd.

The following represent the main findings as regards post-consultatory explanation.

- It was seldom clear to the user how individual certainty factors had been used in deriving the overall confidence that the system had computed for a goal. Often a great deal of time was spent analysing results before full faith was placed in them. This defeats the point of having a computerised system.
- It was recognized that the use of certainties introduced subjectivity into the system and that the degree of subjectivity could be considerable in ecological systems where there was a lack of quantitative data. Analytical methods for observing the effect on the system conclusions of the answers given to questions, were felt to be very necessary.
- Linked to the above point was a need to identify areas where large amounts of uncertainty had lead to doubt about conclusions made.
- It was necessary to have rules justified in terms of facts known to the user.

- There was difficulty conceptualising the structure of the rule base.

4.4. Explanation sub-system guidelines

These studies determined the guidelines for the design of the explanation sub-system.

The guidelines were that the sub-system should provide comprehensive post-consultatory explanation regarding the conclusions of the system. This post-consultatory explanation should:

- Explain and justify the conclusions of a consultation to a user.
- Convey the structure and relationships of the knowledge base to the user.
- Assist the user in determining the sensitivity of the model to the various inputs.

The following chapter describes research issues in the topic of explanation-generation. The results of this research are used in meeting the above guidelines.

5. Explanation-generating facilities - research issues

5.1. Introduction

Research into the development of explanation-generating facilities began in the late 1970's during the MYCIN project. It was not, however, until 1983 that research papers covering a range of different aspects of explanation-generation appeared. Research in this field has intensified in the subsequent years.

Depending on the exact use of the expert system, and the stage of its development, explanatory routines are necessary to :

- Help knowledge engineers test and debug the system during development.
- Assure a sophisticated user that the system's reasoning is appropriate.
- Instruct a naive user about the reasoning and knowledge in the system.

As production rule-based programs are generally thought of as being able to capture an expert's reasoning steps accurately, it is believed that it should also be possible to provide reasonable explanations, as well as good advice, using this knowledge representation [CLA83].

The rest of this chapter describes recent and current research issues in the topic of explanation-generation. Although most of the research described below refers to large expert systems applied to various domains, the results can generally be adapted to the smaller ecological expert systems being developed locally.

5.2. Mycin

The first rule-based expert system to have the development of explanatory routines as a major issue, was MYCIN [SCO84]. Although the development of such routines was not a design issue at the outset of the MYCIN project, methodologies developed during the project have served as the basis for most systems which have explanatory facilities [BER87, CLA83, SAV87].

Initially, the MYCIN researchers became aware that displaying English translations of the rules used in the consultation would assist in explanation. This led to the development of full HOW and WHY traces through the rules. The function of the HOW trace was to show how a fact had been derived from the rules in the rule base. The WHY trace could show, in terms of the rules, why an answer to a question was needed. The success of MYCIN led to the development of GUIDON [CLA84], a system for use in tutorials with medical students. In GUIDON, explanations based on the MYCIN trace were to be used in educating the student users.

Major findings of both the MYCIN and GUIDON projects were that much of the information needed to provide good explanations, the so-called strategic information, was represented implicitly in the rules and therefore lost to the trace explanation facility [CLA83]. The information referred to is that which the knowledge engineer uses in constructing the associations in the rule base, but is not able to represent in rules. This information includes such details as the rule order and the justifications for particular rule associations.

5.3. Rule justification

A great deal of research has recently been conducted into methods of representing such strategic information explicitly [HAS84, CLA83, SWA83]. It is generally recognized that if a system is to justify its reasoning, then it is necessary to model additional levels of expertise, normally left out of the rules in performance-orientated expert systems.

Clancey [CLA83] shows that structures used in justifying rule associations often have to be represented in a form which is redundant to the inferencing mechanism. These structures are then only used by the explanation sub-system.

Swartout [SWA83, SWA87] argues that there are different types of knowledge that should form the basis of an expert system's rules. These are terminological knowledge, domain descriptive knowledge, and problem-solving knowledge. Swartout's approach, and that of the Explainable Expert Systems (EES) project, is for system builders and domain experts to collaborate in the construction of the high-level representations of this knowledge. An automatic programmer is then employed to derive system code from these knowledge representations. Traces of the rule derivation process are left behind and used by the explanation-generating facility to

justify steps taken by the system.

The EES project is not yet complete, but would appear to represent very significant research. Swartout admits, however, that *"much remains to be done before expert systems will be able to explain themselves as lucidly as human experts"* [SWA87].

In the third approach to representing strategic information, Hasling [HAS84] demonstrates how strategy can be made explicit by using meta-rules. Meta-rules, first suggested by Davis [DAV84], are constructed to directly represent the strategy that the domain expert would employ in dealing with such issues as the ordering of rules.

5.4. Analytical routines

In addition to explaining the actions of the system to the user, there is some interest in providing analytical routines, which can enable him to obtain answers to "what-if" type questions.

Coombs and Alty note that in expert systems developed to assist in decision-making tasks, *"there should be focus on elaborating the content relevant to problem-solving rather than in the solution itself"* [COO84]. The user, the authors state, should be able to answer questions including :

What would happen if X?

What are the critical factors in X?

The authors note that few attempts have been made to address such issues in the expert systems area.

It is to the more traditional decision support systems that one has to turn for ideas and to observe new developments in this area.

The answering of "what-if" type questions is known as sensitivity analysis in an Operations Research context. The role of sensitivity analysis has been given, [CLA86], as :

- To assist in understanding the correspondence between a model and its subject.

- To identify variables which have only marginal influence on elements of the model.
- To quantify the impact of variations in input to model output.
- To assess the limits to which input data may be altered without significantly departing from specified ranges of interest in the output data.

Sensitivity analysis can, therefore, be vital in identifying areas in a model where additional research is necessary.

Sensitivity analysis is frequently used in business planning applications. Researchers at the Intelligent Laboratory at Carnegie-Mellon University have developed a software product called ERGO (Explaining Results Generated by Others), which is an explanation-generating interface designed for use with financial planning software. The role of ERGO is to trace variables back to their roots, make comparisons and report on critical factors [JON87]. The system also has an extensive natural language capability. Work is currently in progress on developing an expert system shell within the IFPS/Plus financial planning software package which also interfaces with ERGO.

The ERGO research project represents a clear attempt to use the ideas of sensitivity analysis in expert systems.

5.5. Natural language, graphics and windowing

There is a trend in current explanation facilities to make better use of natural language, graphics and windowing.

The provision of full natural language interfaces for expert systems has recently been identified as a high priority research area. Such interfaces enable more readable traces to be built up and allow for proper access to the system's motivation and behaviour [SPA84].

High resolution graphics allow the system designer to develop facilities which can pictorially portray information otherwise difficult to explain to the user. There is evidence that people

deal with complex interactive systems by building up their own conceptual models of the system [KID83]. Moreover, a graphical representation of the rule network is often preferred to an alphanumeric explanation [AND86]. The realization of these facts has led to some interesting work.

In recently published research Mott and Brooke [MOT87] present an inference system, DTree, which is based around a graphical implementation. In the DTree system, rules are built up as graphical decision trees on the display unit. Use is made of advanced graphical features, pull-down windows and a mouse, in this process. Users can engage in consultations, once a number of rules have been entered, by selecting goals that they wish to investigate from a menu of all the goals. Rule evaluation then proceeds in the normal backward chaining fashion. Explanation of rule evaluation is achieved through displaying, in separate windows, the graphical decision tree and any text which may be associated with the rules by the knowledge engineer. Verbose traces are thus eliminated and text entered with the rules can be used primarily for rule justification.

Windowing, as exemplified in the above research, can allow for the simultaneous display of different information, such as a graphical and textual representation of the same section of the rule base, or a section of the rule trace and, in a separate window, the justification thereof.

5.6. Explanatory features in small expert system shells

Expert system shells which have been reviewed as part of this study are APES, EXSYS, MICRO-EXPERT, SAVOIR, SYNAPSE and TWAICE. SAVOIR and TWAICE are systems developed primarily for large computers, with scaled down versions available on micro-computers. Comprehensive reviews of three of these shells have been published [CAR87a, CAR87b, CAR87c].

The systems all offer some form of HOW and WHY traces for describing, in terms of the rules, how facts were derived, or why answers to questions are needed.

Traces, as exemplified by the following piece of text produced by the MICRO-EXPERT shell, can be poorly implemented and rigid.

"If not *alcohol may harm Peter* then *alcohol is recommended for Peter*" [BER87a].

The wording of such a trace is built up by the concatenation of strings of canned text (in italics) with text generated by the system. Nonsensical statements which confuse or irritate the user could be precluded if a few natural language templates were added into the system.

The best implemented trace is produced by the TWAICE system which allows for well worded traces to be built up with user supplied templates [SAV87].

The SAVOIR system relies to a large extent on canned text, the success of which depends on the knowledge engineer anticipating the needs of the potential user.

Only the SYNAPSE and TWAICE systems offer any analytical features. In SYNAPSE, the current amount of certainty attributed to the goal being checked is displayed at each rule evaluation. No arithmetic detail explaining how this amount of certainty was derived is given. Features for performing some sensitivity analysis are offered by TWAICE. These enable the user to observe the effect of different answers to questions on the system goals. The user is, however, not given any prior indication of which questions are critical.

Presentation of the explanatory information is poor in EXSYS and MICRO-EXPERT, with no, or poor, use made of graphics and windowing.

Surveys undertaken on systems in use [BER87, KID85a], show that most users believe there to be considerable room for improvement in the explanation facilities offered. Systems in use range from those which utilise only canned text for explanation, to those that use rule traces and finally to those which combine the two. The most common form of explanation remains the rule trace, although organisations are increasingly favouring combining text and traces, as a way of providing strategic information.

Berry also notes an increasing realization of the need for natural language and graphical facilities. She states that, "*Good explanation and good natural language facilities go together and require similar developments*", and "*Most organisations visited agree that expert systems are currently far too textual and that improved graphics will greatly enhance the user interface*" [BER87a].

5.7. Conclusion

If there is such a thing as a standard in explanation-generating facilities, then it would appear as if the rule trace holds this position. There are, however, problems inherent with the rule trace approach. These are that :

- Strategic information is not normally coded in the rules and is therefore not available to the trace.
- Information on the sensitivity of model parameters is not represented.
- Traces can be verbose, and even confusing due to bad wording.

Ideas on how to circumvent these shortcomings have been advanced. The following clear patterns are emerging :

- Separate knowledge structures are needed to represent strategic information.
- Analytical routines, which can represent model sensitivities, will play an important role in the success of future systems.
- Increased use of the "new technologies" of natural language, graphics and windowing is vital for enhancing the comprehension of system output.

6. The P.R.O. Explanation sub-system

6.1. Introduction

It was stated in chapter 1 that the aim of this research was to recommend and develop explanation-generating facilities for use in systems produced using the P.R.O. Expert System Toolkit. Guidelines for the design of these explanation-generating facilities were given in chapter 4. This chapter shows how the results of the research described in chapter 5 have been used to meet the guidelines.

From the descriptions of the research issues given in chapter 5, it should be clear that the implementation of a fully comprehensive explanation-generating facility for all applications of the P.R.O. toolkit would be a long-term evolutionary project, which could only be set in motion during this research.

This chapter discusses the features that have been implemented to date and makes recommendations, based on current research trends, for any subsequent work.

The P.R.O. inference mechanism has been devised to validate the truth, and in the process derive a value for a goal selected to be the current goal. The selection of the current goal occurs at the start of the consultation and the subsequent question and answer session performs the validation.

Explanatory routines, henceforth known as the explanation sub-system, have been designed so that the user can investigate how the value for the current goal was derived, if the goal was established to be true, or why the current goal was proved to be untrue. The sub-system is furthermore designed to provide support and justification for, as well as analysis of, the conclusions made.

The features of the sub-system supplement the question-time explanatory features implemented earlier [BRA86b].

This explanation sub-system is split into the rule trace and the analytical component, the sensitive question identifier (SQI).

6.2. The rule trace

As noted in chapter 5, rule traces are widely implemented in rule-based expert systems, although they are often of a poor standard in small systems. A major design issue was thus to develop a high quality rule trace.

The implementation of a basic rule trace, analogous to the HOW trace of MYCIN, is fairly simple if a dynamic stack of facts derived during the consultation is built up in working memory. All that is then needed to supplement this is access to the relationships within the rules in the knowledge base.

An example rule base segment used to show how traces can be built up is given in Fig. 6.1. Such rules could form part of a larger rule base used for determining whether fish species are likely to invade sensitive river areas, see [PUR87].

```

Goal Survive true If
    Water_temperature > = 20 and,      (I)
    Food_present true.                  (II)

Conclude Food_present true If
    Snails_present true.                (III)

Question Water_temperature.
Question Snails_present.

```

FIG. 6.1 Example rule base.

Suppose that during the process of a consultation the question `Water_temperature` was answered with a minimum of 18 and a maximum of 20, and the question `Snails_present` was answered yes with a certainty of 10. The system can thus show that the goal `Survive` is true and a post consultation rule trace, as in Fig. 6.2, could be displayed.

The Goal: Survive was shown true
with a certainty factor (Cf) of 9.

Because:

The Question:

Water_temperature was answered 18 - 20.

And the Rule:

Food_present was shown to be true with a Cf of 10.

The Rule: Food_present was shown true
with a Cf of 10.

Because:

The Question:

Snails_present was answered yes with a Cf of 10.

FIG. 6.2 A basic rule trace through the example rule base.

For a rule trace to represent a significant improvement over that displayed in Fig. 6.2, and those of the systems reviewed, the following points are significant.

- The trace should display strategic information which has been lost in the construction of the rules.
- The facility must convey to the user how the certainty factors that he supplied were utilized to derive the overall goal certainty.
- Text produced must be understandable, unambiguous and less rigidly presented than in the systems reviewed.
- The overall facility should be easy to use and understand.

We now show how these issues are addressed in the explanation sub-system.

6.2.1. Strategic information

It was observed in the previous chapter that many authors now recognize that a complete representation of all the knowledge in a particular domain will involve separate knowledge structures, and that considerable research is currently being conducted in this area. It is also significant to note that the topic is linked with that of knowledge acquisition - the area where there is the largest bottleneck in expert systems research [BER87b, BUC84, OLS87].

Rather than become too embroiled in this topic, the author contends that strategic information can adequately be represented by a string of canned text associated with each rule-condition link in the knowledge base. The knowledge engineer can in this way make contact with concepts which underlie causal associations and are familiar to the user. Steps taken by the system can thus be justified, and the order of the rules explained. This approach is based on the ideas of Clancey [CLA83] and facilities noted to be in use in Britain [BER87a].

Three rule-condition links exist in the rule base (I,II,III in Fig. 6.1). An example of the type of canned text that can be appended to condition (II) for justification of the causal relationship and rule order, is given below.

- (II) Once it has been determined that the species can survive in the river, it is necessary to determine whether a food source exists.

As far as recommendations for future development are concerned, it is the feeling of the author that the joint topic of knowledge acquisition, and the representation of knowledge in separate structures, should be an area for future research at Rhodes University. Such research will impact on methods of strategic explanation-generation.

6.2.2. Certainty combination

It was noted in chapter 4 that users of systems developed in P.R.O. often spend a great deal of time analysing how certainty factors have been propagated through the rule network. Showing the steps in the arithmetic combination of these factors is an easy amendment to the textual trace. Although simple, this facility can assist the knowledge engineer (in rule base debugging) as well as the user (in understanding the inner mechanisms of the system).

6.2.3. Natural language, graphics and windowing

Development of a comprehensive natural language interface was outside the realms of this project. The production of more intelligible textual traces has, however, been attained through the use of a small number of natural language templates. The use of these templates is complemented by the addition of a short piece of text to each rule and question. This text is then supplemented for the rule/question name in the trace.

Further natural language facilities, particularly routines that could allow users to phrase questions to ask of the system at the end of a consultation, would be useful. MYCIN [SCO84] has such a facility, although it is believed to be fairly elementary. Such facilities could, however, form the basis for more comprehensive routines, enabling the user to phrase "what-if" queries as in the ERGO project [JON87].

To assist the user in understanding the rule trace, a graphical portrayal of this trace can be viewed on the screen. Graphical representations often suffer from being too crowded with text. For this reason, only the unique hierarchical rule number, also used in the textual traces, is displayed in the graphical tree. The hierarchical rule number is designed to portray the depth and breadth of a condition in the reasoning tree. It is unfortunate that this graphical trace and the associated textual trace and justifications can not be viewed concurrently in separate windows. This limitation is caused by software constraints.

Improvements to this facility, allowing windowing and zooming will only become possible when better graphics capabilities, such as bit-mapped displays, become available on micro-computers.

6.2.4. Viewing the trace

Interaction with the sub-system is facilitated through the use of menus and windowing which allows for different parts of the textual trace to be listed concurrently.

Results of the trace can be obtained rule-by-rule on the screen or routed to a printer. When large rule bases are used, users are often scared off by lengthy traces. For this reason, printed traces can be started at any rule by making use of the screen-routed rule-by-rule trace until a specific start rule is located. A printed rule trace can then be obtained for the

section of the rule base which was used to prove the rule true.

Fig. 6.3 shows a trace through the example rule base, augmented with the above mentioned features. Such a trace would be obtained if the printer were selected for output. Screen-routed traces show the rule-condition links separately and allow users to move through the links as they wish. Fig. 6.4 shows the graphical representation of this rule trace.

The Goal: The species can survive in the environment
was shown true with a certainty factor (Cf) of 9.

Because:

The Question:

1. The temperature of the water was answered with a maximum value of 20 which is equal to the prescribed minimum of 20 with Cf 8.

If the species is to survive in the environment then it is essential that the water temperature is above or equal to 20 deg. C.
- refer to Fishtables 1.

And the Rule:

2. There is food present was shown true with Cf 10.

Once it has been determined that the species can survive in the environment, it is necessary to determine whether a food source exists.

Cf Goal = $0.5 * Cf \text{ of } Q \ 1. + 0.5 * Cf \text{ of } R \ 2.$

Rule 2. was shown true because :

The Question:

- 2.1. Are there snails present in the river was answered yes with Cf 10.

The species can only survive if snails are present as a food source. - refer to Fishtables 23.

Cf Rule 2. = $1.0 * Cf \text{ of } Q \ 2.1.$

FIG. 6.3 A full rule trace through the example rule base.

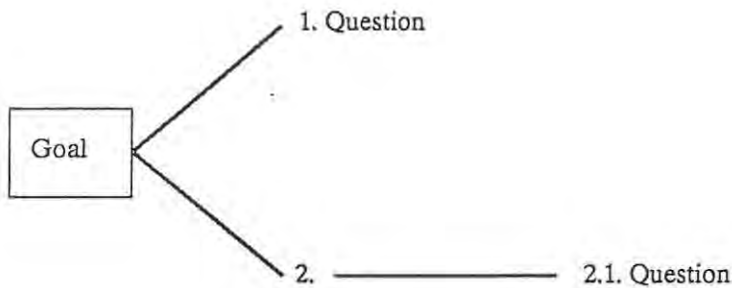


FIG. 6.4 A graphical representation of the rule trace.

6.3. The sensitive question identifier (SQI)

It was decided at the outset of the project that the development of a set of analytical routines was to be a major design goal. It was felt that there was need for such facilities in systems which incorporate uncertainty in their reasoning. Specifically, users wish to :

- Know the impact of certain answers to questions on the system goals.
- Pinpoint areas where a large amount of uncertainty in a rule evaluation has lead to a dubious result.

It was noted in the above chapter that there has been little previous research into the provision of this type of explanatory assistance in expert systems. Implemented features have therefore been based on the needs of local expert systems users, and the methods used in more traditional decision support systems [CLA86, JON87, TAH76]. There are presently three components to the SQI.

The first component consists of routines which list the questions used in deriving a value for the current goal. As noted earlier, the current goal can be shown to be true with a certain value and confidence ascribed to it, or proved untrue.

If the current goal is proved untrue, then a listing of those questions whose answers lead

directly to this is given. The questions are listed with the answers that are needed for the goal's validity to be reversed. The user can thus determine whether, with more research into the value ascribed to the question, the goal's truth could be verified.

Returning to the example given (Fig. 6.1), suppose that the question **The temperature of the water** was answered with a minimum of 15 and a maximum of 18. The inference engine can then no longer validate the truth of the current goal. A listing as in Fig. 6.5 is produced by the explanation sub-system.

The current goal : The species can survive in the environment
cannot be proved true.

For it to be proved true, the following questions have to answered as :

<u>Question</u>	<u>Answer</u>
The temperature of the water	maximum value of greater than 20

FIG 6.5 Listing produced when a goal is proved untrue.

If the current goal is, on the other hand, shown to be true, then routines are invoked which list the questions used in deriving a value for the goal, their answers and the extent to which the amount of certainty ascribed to them affected the result (see Fig. 6.6). These routines have been found to be extremely helpful in debugging rule bases and in ascertaining which the most important questions are.

The second set of routines are used to identify areas where the result of a numeric evaluation has been put in doubt owing to the presence of uncertainty. In the example (Fig. 6.1) the question **The temperature of the water** was answered with a minimum value of 18 and a maximum of 20. The relevant condition (I) in the initial rule states that the temperature has to be greater than or equal to 20. As there is uncertainty in the answer given (value 18-20), there is a possibility that the true answer is less than 20, and thus the result of this condition has to be flagged. Such a question is highlighted to the user as being sensitive (see Fig. 6.6), indicating that more care has to be taken in answering it if the relevant condition

is to succeed with full confidence.

<u>Question</u>	<u>Answer</u>	<u>Effect(%)</u>
The temperature of the water	18 - 20	44 - S
Are there snails in the water	yes	56

FIG. 6.6 Questions used, their numeric effects and sensitivity. Sensitive questions are flagged with an S.

A final set of SQI routines enable the user to re-answer a question and note the effect of the new answer on the system goals. The routines allow the user to obtain a report on the various answers given to questions and their respective effects on the goals. The user can thus build up an understanding of the sensitivity of questions. The effect of these routines is thus to perform sensitivity analyses, as described in chapter 5, on the model parameters.

As an example, consider that the question **The temperature of the water** is to re-answered and the effect noted. The question is chosen to be re-answered as it has been identified as a sensitive question (Fig. 6.6), and different measurements are available. If the question is re-answered with a minimum value of 18 and a maximum of 22, and with a minimum value of 17 and a maximum of 19, then the listing given in Fig. 6.7 is produced by the explanation sub-system.

Sensitivity of question :

<u>Answer</u>		<u>Goal and certainty</u>	
18 - 20		The species can survive in the environment	9
18 - 22		The species can survive in the environment	8.5
17 - 19		--- No goal is true ---	

FIG 6.7 Listing produced by routines of the SQI when a question is re-answered.

These routines are especially useful, as demonstrated above, when a sensitive question is identified, or when there is disagreement about the answer to a question. The above example is trivial, but in more complex models, where there are a number of interactions and dependencies, it can become vital to note the effect of different answers.

Only the second set of SQI routines are dependent on the method of uncertainty propagation employed. The routines described in this research represent one possible model for the identification of sensitive questions. The other SQI routines are independent of methods of uncertainty propagation, and are therefore directly portable to other systems.

6.4. Summary

The routines developed for the P.R.O. explanation sub-system can be split into those of the rule trace, and those of the sensitive question identifier (SQI).

The rule trace has been designed to be more comprehensive and understandable than in the expert system shells reviewed. Most of the features implemented have been suggested and described in recent research.

The SQI routines have been devised in response to users' requirements, and demonstrate what can be achieved in the provision of analytical routines in small expert systems. There has been little prior research in the area of providing the user with routines to answer "what-if" type questions, although the value of having such facilities has been recognised. It is, therefore, believed that there will be much further activity in this area.

As highlighted above, the sub-system has been developed in response to the requirements of ecologists who are using the P.R.O. Expert System Toolkit for expert system development. The routines have been interfaced with one completed ecological expert system. This system, and the benefits provided by the explanation sub-system, are described in the next chapter.

7. Applications

This chapter serves as a brief description of the current applications of the P.R.O. Expert System Toolkit and the explanation sub-system.

The major system into which the explanation sub-system has been introduced is the FISHFARMER aquaculture system, which has recently been completed at Rhodes University [BRA86, BRU86, DEA87, RIO87].

The INVEST demonstration expert system is a small financial expert system developed specifically to demonstrate the features exhibited by the P.R.O. Expert System Toolkit as a whole, and the explanation sub-system in particular. INVEST was developed as the FISHFARMER system was too large and specific for demonstrating the features of the toolkit to a wide range of potential users. The domain selected for the INVEST system was chosen to be simple and understandable by users.

7.1. FISHFARMER

Aquaculture has been defined as the controlled cultivation and harvest of aquatic organisms for commercial utilization [DEA87]. Aquaculture has a healthy future in Southern Africa, but has not yet developed to its full potential. One of the main constraints on aquaculture advancement is the lack of available expertise. FISHFARMER has, therefore, been developed to :

- Aid in the assessment of the aquaculture potential of various fish species in relation to potential sites of introduction and culture methods.
- Collect and collate all relevant data pertaining to aquaculture in Southern Africa.

The knowledge base has information on fourteen fish species, and consultations are designed to provide an assessment of the aquaculture potential of any of these species, given the conditions of the potential site. Many of the species included are not being cultured commercially in South Africa, but are anticipated to be considered in the future. The

knowledge base thus represents a systematic collection of the available data, which can easily be accessed and modified as needed.

It is envisaged that the system will be widely used by Nature Conservation Officers, fish farming consultants, researchers, and anyone interested in assessing the aquaculture potential of their circumstances.

So that the system can provide useful assistance to these users, it is important that it be user friendly and that it can :

- Explain why a particular question is being asked.
- Explain technical terms and procedures associated with questions.
- Explain the rationale of its reasoning concerning the advice given [DEA87].

The FISHFARMER knowledge base has been under development for a period of approximately two years. Prototype systems have been demonstrated at regular intervals during development. Response to the system has been very favourable.

The explanation sub-system was incorporated into the FISHFARMER expert system after completion of the knowledge base. It is felt, by those involved in the FISHFARMER system, that the incorporation of the full set of routines making up the explanation sub-system renders the results of the system more acceptable. The explanatory routines will be particularly useful when :

- The sensitivities of critical variables are to be observed.
- Knowledge contained in the rules is to be investigated as new data becomes available, or as new species are to be considered for aquaculture. The routines can, in this context, provide an easy interface for the examination of the dependencies and inter-relationships contained within the rules.

7.2. INVEST

The INVEST demonstration expert system is designed to provide guidance to users who wish to make investment decisions. Decisions suggested by the system are based on current and expected South African business conditions. The system does not take the user's personal financial position into account.

A consultation with the demonstration system takes the form of a guided step-by-step session. During a consultation one of five conclusions can be made by the system. These conclusions, or systems goals, are :

- Invest in long term fixed rate deposits.
- Invest in short term fixed rate deposits.
- Invest in property.
- Invest in consumer items.
- Invest in industrial stocks.

To validate the truth of a particular goal, the system asks the user to supply financial and business condition indices.

The demonstration system has been designed to automatically supply answers to system questions, at the same time explaining the steps that it performs.

A brief description on how to get the INVEST system operating is given in Appendix B.

8. Computational aspects

8.1. Introduction

As the aim of this research was to develop explanation-generating facilities for use in systems produced with the P.R.O. Expert System Toolkit, it was necessary that the routines implemented were fully compatible with the toolkit. Development was thus restricted to the micro-computer environment. Furthermore, because P.R.O. had been successfully used in the development of small expert systems [BRA86b, BRU86, O'K86, RIO87], it was desirable that as few changes as possible were made to the existing system.

The original routines of the P.R.O. Expert System Toolkit were developed in Turbo Prolog [BRA86b] and there appeared, at the commencement of this research, to be no reason for changing this implementation environment.

8.2. Input to the explanatory routines

In chapter 3 the backward chaining inference mechanism employed by the P.R.O. inference engine was described. Systems such as P.R.O. derive knowledge about facts in the process of backward chaining. These facts are derived by asking questions and evaluating conditions specified in the rules of the knowledge base.

The inference engine is responsible for building up and maintaining these known facts during the consultation. The main role of explanation-generating facilities, as perceived in this research, is to display these known facts and to provide routines for analysing the relationships between them. This should be done in a form that can provide insight into the mechanisms of the system.

As noted in chapter 6, all that is needed for the implementation of a rule trace similar to the HOW trace of MYCIN, is access to the facts derived during the consultation, supplemented by information on the relationships within the rules of the knowledge base. This information also provides the basis of the routines of the SQI.

In the P.R.O. Expert System Toolkit, facts derived during the consultation (answers to

questions and results of evaluations) are stored as separate Prolog database clauses [BOR86, BRA86b, CLO81]. The certainty factor associated with each fact is stored along with its answer in these clauses.

For the relationships within the rules of the knowledge base to be accessible to the explanation sub-system, it is necessary that these relationships are represented as Prolog database facts, in addition to their representation as rules for use by the inference engine. This duplication of information is necessary as P.R.O. rules, when encountered, begin the inference process.

Compilation of these relationships into separate facts is a process that need be performed when the knowledge base has been completed, and repeated only when the rules of the knowledge base are modified. This function is, for this reason, performed by a system pre-processor which operates separate to the main P.R.O. consultation system. The output of the pre-processor is read in by the P.R.O. toolkit at the start of any consultation. Having this compilation as a separate step assists in keeping the computational load of the main system as low as possible.

The structures outlined above are the fundamental inputs necessary for the production of rule traces and the implementation of the first two sets of SQI routines. The only additional structure needed by the routines of the rule trace is the justification structure. The third component of the SQI relies largely on the routines of the inference engine.

8.2.1. Structures representing rule justification

The author proposed in section 6.2.2. above, that strategic information used to justify rule associations can adequately be represented by a string of canned text associated with each rule-condition link in the knowledge base. This is achieved by appending justification text to each of the conditions in a P.R.O. knowledge base.

The syntax for these condition types is given in Fig 8.1.

```

checkres( cond_name1, cond_name2, relative importance, answer, justification)

checkvalue( cond_name1, cond_name2, relative importance, bound of answer,
            type of comparison, answer, justification)

```

FIG. 8.1 The modified checkres and checkvalue condition predicates.

The arguments of these predicates are as defined by Bradshaw [BRA86b], apart from the final justification argument. This final argument, in both predicates, is incorporated so that a string of text can be associated with each condition for justification of the rule-condition association.

8.2.2. The result predicate

The `result` database predicate, as described by Bradshaw, is used to store the answers and certainty factors of questions and numeric evaluations. The syntax of the predicate is given in Fig. 8.2.

```

result( name1, name2, certainty factor, answer, truth status flag)

```

FIG. 8.2 The result database predicate.

The `name1` and `name2` arguments of the `result` predicate follow the variable-naming convention employed throughout the P.R.O. toolkit. The certainty factor and answer are as specified by the user or computed by the system when evaluations are performed. The final argument contains an indication of the truth of the result. Answers to questions are always true, but the results of condition evaluations can be true or false.

8.2.3. The rule1cres and rule2cval predicates

The relationships within the rules of the knowledge base are kept in two database predicates, the syntax of which is given in Fig. 8.3.

```

rule1cres(  name1, name2, rule answer, cond_name1, cond_name2,
            relative importance, answer, justification)

rule2cval(  name1, name2, rule answer, cond_name1, cond_name2,
            relative importance, bound of answer, type of comparison,
            answer, justification)

```

FIG. 8.3 The rule1cres and rule2cval database predicates.

These database predicates are compiled by the system pre-processor from the rules of the knowledge base. The arguments of the predicates are derived directly from the rules. The `rule1cres` predicate is formed from the concatenation of the arguments in a rule body and an associated `checkres` condition. The `rule2cval` predicate is similarly formed from the concatenation of the arguments in a rule body and a `checkvalue` condition.

8.3. Combining these structures into an explanation-generating facility

8.3.1. The rule trace

The routines of the rule trace are only invoked once the current goal has been proved true. All clauses of the `result` predicate with a truth status flag set to true are used to prove a goal true. Merely displaying these true facts is not deemed as being adequate.

These true facts are ordered into a rule trace using the relationships defined by the `rule1cres` and `rule2cval` database predicates. This principle is used whether the trace is displayed on the screen or the printer or whether a graphical representation of the trace is viewed.

All additional information needed for the production of justifications, and details on certainty combination, is contained in the `rule1cres` and `rule2cval` predicates.

8.3.2. The routines of the sensitive question identifier (SQI)

The principle underlying the first two sets of routines making up the SQI is similar to that described above. To derive the details on effects and sensitivities of answers given, all the

steps making up the rule trace have to be taken. However, with the SQI, the only facts displayed to the user are those that were derived from questions.

The third set of routines completing the SQI depend, to a large extent, on the routines of the inference engine for inferring new facts once a question is re-answered.

8.4. The P.R.O. Expert System Toolkit

The P.R.O. Expert System Toolkit has been designed in a modular form, which enables the system developer to use only the routines which are needed [RIO87]. This toolkit approach recognizes that expert system developers have varying requirements [JAC86, WAT86].

The largest single component of the P.R.O. Expert System Toolkit, the user interface module, constitutes approximately 70% of the toolkit source code with no knowledge base included, or 45% of the code with the inclusion of the FISHFARMER knowledge base. This relationship is consistent with the results of other surveys. Berry [BER87b] describes the results of a survey of expert systems in use in Britain, which shows that the user interface was the largest component of the systems surveyed with *"44% (of the total code) involved in user input and output"*.

The routines of the P.R.O. user interface can, in turn, be split into those of the question and answer session, and those of the explanation sub-system. The code of the explanation sub-system makes up 80% of the total user interface code.

8.4.1. The development of an expert system using the toolkit

It is the recommendation of the author that those wishing to use the P.R.O. Expert System Toolkit for the first time, begin their investigations by becoming familiar with the demonstration INVEST system. INVEST, described in detail in Appendix B, provides potential users with a guided illustration of all the features in the P.R.O. Expert System Toolkit.

A copy of the INVEST knowledge base (INVEST.KNB) is supplied on the system diskette. This knowledge base should be printed out, so that the steps that the inference engine takes through the rules can be followed.

The author then recommends that small prototypical knowledge bases are developed in the domain of interest. This process of prototyping is extensively used in expert systems development [BAR84, HAY84]. The knowledge engineer can, during this process, determine which of the modules offered by the toolkit are required in the system.

Knowledge engineers might find further modifications to the routines of the toolkit necessary. The system code has been designed to be flexible so that any such modifications are simple to perform.

8.4.2. Extensions to the toolkit

It was noted above that knowledge acquisition is cited as being the current major bottleneck in the production of expert systems.

Although knowledge acquisition tools have not been used in the implementation of any of the systems developed using the P.R.O. Expert System Toolkit, it is felt that future development of both the FISHFARMER and the PISCES systems would benefit from having such facilities.

Limitations thrown up by both the micro-computer environment and the particular software used were encountered during development of the explanation sub-system and the subsequent incorporation of the routines into the FISHFARMER system. Further development of the P.R.O. Expert System Toolkit could be limited by these constraints.

A knowledge acquisition module would be separate from the main routines of the toolkit responsible for the consultation, and would therefore not result in any machine memory difficulties.

A number of standard Prolog systems, which offer the advanced features of Turbo Prolog, are now available. It is believed that these systems could offer better performance. The advantages of converting the toolkit to one of these systems should be investigated before any further development is undertaken.

9. Conclusions

Effective explanation-generating routines, as components of the user interface, are fundamental to the credibility of the results of any advice-giving expert system.

Explanation-generating facilities should be able to :

- Explain and justify the conclusions of a consultation to a user.
- Convey the structure and relationships of the knowledge base to the user.
- Assist the user in determining the sensitivity of the model to the various inputs.

The user, in this context, can range from a first time user seeking to understand how a conclusion was made by the system, to a knowledge engineer wishing to have easy access to the relationships contained within the rules of the knowledge base.

The documentation presented here has described the development of an explanation sub-system for the P.R.O. Expert System Toolkit. The toolkit operates in a micro-computer environment. Development of the explanation sub-system has involved adapting the results of recent research in the area of explanation generation to the requirements of ecologists who use the P.R.O. Expert System Toolkit for the development of small expert systems.

The first two design goals given above have been met in the explanation sub-system by providing traces through the rules of the knowledge base that were used in the consultation. Recent research has given pointers as to how rule traces can be augmented to provide for more effective explanation. Many useful ideas gleaned from studying this research have been incorporated in the rule trace of the P.R.O. explanation sub-system. This sub-system provides more comprehensive explanation than any of the commercially available small expert system shells reviewed.

Very little prior research on the implementation of routines to meet the third design goal was noted. It was felt, however, that such facilities were especially important in systems which incorporate uncertainty in their reasoning. A collection of routines which meet this design

goal have been included as a major component of the sub-system.

The explanation sub-system has been interfaced with the FISHFARMER ecological expert system which has recently been completed. It is believed that the explanatory routines will make a large contribution to the acceptability of the advice that the system provides. The sub-system will also be used to provide easy access to the knowledge contained in the rules of the knowledge base. This facility will be especially useful when system revisions become necessary.

The development of a full explanatory capability is an evolutionary process, with research, development and user feedback occurring concurrently. The next step in this development is for the users of P.R.O. based systems, which incorporate the explanation sub-system, to highlight areas where system reasoning is not yet transparent, or where further explanatory or analytical features are needed.

It is the opinion of the author that further research into the provision of explanation-generation facilities for expert systems should be conducted in parallel with research in the areas of knowledge acquisition and representation. Only when the results of research in these areas becomes integrated, will it be possible for expert systems that explain themselves as lucidly as human experts to be developed.

Bibliography

- [ALH87] Conference on Alternative Life History Styles for Fishes and Other Organisms, Grahamstown, June 1987.
- [AND86] Andriole, S.J., Graphic Equivalence, Graphic Explanations, and Embedded Process Modelling for Enhanced User-System Interaction, *IEEE Trans. on Systems, Man and Cybernetics*, 16, No. 6, 919-926, 1986.
- [BAR84] Barber, E.O., Expert Systems Survey, *Proc. Management Briefing on Expert Systems for Informatics*, London, 1984.
- [BER87a] Berry, D.C. and Broadbent, D.E., Expert Systems and the Man-machine Interface, Part Two : The User Interface, *Expert Systems*, 4, No. 1., 18-27, 1987.
- [BER87b] Berry, D.C., The Problem of Implicit Knowledge, *Expert Systems*, 4, No. 3, 144-151, 1987.
- [BOR86] *Turbo Prolog Manual*, Borland International Inc., California, 1986.
- [BRA85] Bramer, M., *Research and Development in Expert Systems*, (Ed.), Proc. Fourth Technical conference of the British Computer Society Specialist Group on Expert Systems, Cambridge University Press, Cambridge, 1985.
- [BRA86a] Bradshaw, J., The P.R.O. Expert System Shell, *Proc. First Computer Science Research Students Conference*, Stutterheim, 1986.
- [BRA86b] Bradshaw, J., The P.R.O. Expert System Shell, (M.Sc. Thesis), *Tech. Doc. 86/6*, Department of Computer Science, Rhodes University, Grahamstown, 1986.
- [BRU86] Bruton, M., Dean, N. and Bradshaw, J., An Expert System for Acquaculture, *Proc. Acquaculture '86 Conference*, CSIR, Pretoria, 1986.
- [BUC84] Buchanan, B.G. and Shortliffe, E.H., *Rule-based Expert Systems*, Addison-Wesley, Mass, 1984.

- [CAR87a] Carden, K.J. and Purchase, H.C., An Evaluation of EXSYS, *Tech. Doc. 87/19*, Department of Computer Science, Rhodes University, Grahamstown, 1987.
- [CAR87b] Carden, K.J. and Purchase, H.C., An Evaluation of SYNAPSE, *Tech. Doc. 87/20*, Department of Computer Science, Rhodes University, Grahamstown, 1987.
- [CAR87c] Carden, K.J. and Purchase, H.C., An Evaluation of MICRO-EXPERT, *Tech. Doc. 87/23*, Department of Computer Science, Rhodes University, Grahamstown, 1987.
- [CLA83] Clancey, W.J., The Epistemology of a Rule-based Expert System - a Framework for Explanation, *Artificial Intelligence*, 20, 215-251, 1983.
- [CLA84] Clancey, W.J., Extensions to Rules for Explanation and Tutoring, In: Buchanan, B.G. and Shortliffe, E.H. (Eds.), *Rule-based Expert Systems*, Addison-Wesley, Mass, 1984.
- [CLA86] Clarke, K. and Cochrane, K.L., A Sensitivity Analysis of the Hartebeestpoort Dam Ecosystem Model, *Proc. O.R.S.S.A. Conference*, University of Cape Town, 30-41, 1986.
- [CLO81] Clocksin, W. and Mellish, C., *Programming in PROLOG*. Springer-Verlag, Heidelberg, 1981.
- [COO84] Coombs, M. and Alty, J., Expert Systems - an Alternative Paradigm, In: Coombs, M. (Ed.), *Developments in Expert Systems*, Academic Press, London, 135-157, 1984
- [DAV84] Davis, R. and Buchanan, B.G., Meta-level Knowledge, In: Buchanan, B.G. and Shortliffe, E.H. (Eds.), *Rule-based Expert Systems*, Addison-Wesley, Mass., 507-530, 1984.
- [DEA87] Dean, N., FISHFARMER - An aquaculture expert system, *Honours project*, Department of Ichthyology and Fisheries Science, Rhodes University, 1987.
- [ES83] *Expert Systems '83.*, Proc. Third Technical conference of the British Computer Society Specialist Group on Expert Systems, Cambridge, December, 1983.

- [HAM86] Hammond, P. and Sergot, S., *APES Augmented Prolog for Expert Systems - User Manual*, Logic Based Systems, 40 Beaumont Ave, Richmond, Surrey, TW9 2HE, England, 1986.
- [HAR85] Harmon, P. and King, D., *Expert Systems - Artificial Intelligence in Business*, John Wiley & sons, New York, 1985.
- [HAS84] Hasling, D.W., Clancey, W.J. and Rennels, G., Strategic Explanations for a Diagnostic Consultation System, In: Coombs, M. (Ed.), *Developments in Expert Systems*, Academic Press, London, 117-133, 1984.
- [HAY84] Hayes-Roth, F., The Knowledge-Based Expert System : A Tutorial, *Computer*, 11-28, Sept. 1984.
- [JAC84] Jackson, P. (1984) A Flexible Toolkit for building Expert Systems, *Research paper no. 245*, Department of Artificial Intelligence, University of Edinburgh.
- [JAC86] Jackson, P., *Introduction to Expert Systems*, Addison-Wesley, Wokingham, England, 1986.
- [JON87] Jones, K., Artificial Intelligence - victory of mind over matter, *Computing S.A.*, 21-22, September 21, 1987.
- [KID83] Kidd, A.L. and Cooper, M.B., Man-Machine Interface for an Expert System, In: *Expert Systems '83*, Cambridge University Press, Cambridge, 1983.
- [KID85a] Kidd, A.L. and Cooper, M.B., Man-Machine Interface issues in the construction and use of an Expert System, *Int. J. Man Machine Studies*, 22, 91-102, 1985.
- [KID85b] Kidd, A.L., What do users ask? - Some thoughts on diagnostic advice, In: Merry M., (Ed.), *Expert Systems '85*, Cambridge University Press, Cambridge, 1985.
- [LES85] Lesmo, L., Saitta, L. and Torasso, P., Evidence Combination in Expert Systems, *Int. J. Man Machine Studies*, 22, 307-326, 1985.

- [MER85] Merry, M., *Expert Systems '85*, (Ed.), Proc. Fifth technical conference of the British Computer Society Specialist Group on Expert Systems, Cambridge University Press, Cambridge, 1985.
- [MOT87] Mott, P. and Brooke, S., A Graphical Inference Mechanism, *Expert Systems*, 4, No. 2, 107-117, 1987.
- [NIL86] Nillson, N., Probabilistic Logic, *Artificial Intelligence*, 28, 77-87, 1986.
- [O'K86] O'Keefe, J., Danilewitz, D. and Bradshaw, J., The River Conservation System - an automated 'knowledge assistant' to help in determining the conservation status of South African rivers, *Proc. Conference on Freshwater Wetlands and Wildlife*, Charleston, U.S.A., 1986.
- [OLS87] Olson, J.R. and Reuter, H.H., Extracting Expertise from experts : Methods of knowledge acquisition, *Expert Systems*, 4, No. 3, 152-168, 1987.
- [PUR87] Purchase, H.C., P.R.O. version 2, *Tech. Doc. 87/26*, Department of Computer Science, Rhodes University, Grahamstown, 1987.
- [RIO87] Riordan, D., The Development of a Toolkit for Building Expert Systems, *Tech. Doc. 87/28*, Department of Computer Science, Rhodes University, Grahamstown, 1987.
- [SAV87] Savory, S.E., *Intermediate level features of TWAICE; Nixdorf TWAICE course notes*, Nixdorf, West-Germany, 1987.
- [SCO84] Scott, A.L., Clancey, W.J., Davis, R. and Shortliffe, E.H., Methods for Generating Explanations, In: Buchanan, B.G. and Shortliffe, E.H. (Eds.), *Rule-based Expert Systems*, Addison-Wesley, Mass., 338-362, 1984.
- [STA84] Starfield, A., Adams, S. and Bleloch, A., A Small Expert System Shell and its Applications, University of Witwatersrand, 1984.
- [SPA84] Sparck-Jones, K., Natural Language Interfaces for Expert Systems : An Introductory note, In: Bramer, M.A. (Ed.), *Research and Development in Expert Systems*, Cambridge University Press, Cambridge, 85-94, 1984.

- [SWA83] Swartout, W.R., XPLAIN: A system for Creating and Explaining Expert Consulting Programs, *Artificial Intelligence*, 21, 285-325, 1983.
- [SWA87] Swartout, W.R. and Smoliar, S.W., On making expert systems more like experts, *Expert Systems*, 4, No. 3, 196-207, 1987.
- [TAH86] Taha, H.A., *Operations Research : an Introduction*, MacMillan Publishing Co. Inc., New York, 1986.
- [WHI84] White, A.P., Inference Defficiencies in Rule-based Expert Systems, In: Bramer, M.A. (Ed.) *Research and Development in Expert Systems*, Cambridge University Press, Cambridge, 39-50, 1984.
- [WAL86] Walker, T.C. and Miller, R.K., *Expert Systems 1986: An assessment of technology and applications*, SEAI technical publications, 1986.
- [WAT86] Waterman, D., *A Guide to Expert Systems*, Addison-Wesley, Reading, Massachusetts, 1986.

Appendix A

It is important to note that the demonstration system referred to in the following questionnaire was constructed before completion of work on the explanation sub-system. The explanatory information referred to represents a prototypical subset of the full explanation sub-system described in chapter 6.

Explanation in Expert Systems

Questionnaire

It is acknowledged that expert system explanation facilities are useful during :

- Development and testing of a system; to help in debugging.
- Day-to-day use of the system; so that system reasoning and results can be explained to users.

The possible forms that an explanation facility can take are very varied and dependent on factors such as :

- Who the system is going to be used by.
- Whether or not uncertainty is used.

In the demonstration system most of the explanatory information provided is available after

the system has shown one of the goals to be true. Information can be summarised into the following categories :

- Transparency of the line of reasoning taken by the system.
This is displayed in overlapping windows during the consultation.
- Traces through the rules used to show the goal true. These are augmented by :
Justifications supplied by the knowledge engineer and, information on how user supplied certainties were combined to achieve an overall goal certainty.
These traces are given on the screen or on the printer.
- Numeric information quantifying which questions had the greatest effect on the goal.
- A graphical representation of the network of rules used.

I would be grateful if you would complete the following questionnaire and return it to me.

1. Is there a need for an explanation facility in the expert system that you are involved with?

Yes/No

2. Would the demonstrated features represent a useful addition in this system?

Yes/No

If yes, would they be most useful in

debugging/day-to-day/both

If no, are there any such features that you think might be useful?

.....

3. Is the output of this explanation facility easy to understand?

Yes/No

If no, do you have any ideas about improvements?

.....

4. Which features do you think potentially have the most relevance to your system?

Overlapping windows
Textual trace with justification
Information on certainty combination
Numerical effect of answers
Graphical rule network

5. Do you think there is a need for more explanatory information to be available at question time, or at any other time during the consultation?

Yes/No

Comments

6. Traces and graphical representations can be difficult to follow in large systems. Do you see this as a problem?

Yes/No

7. The present system demonstrates the numerical effect of a given answer, but gives no "what-if" type information - e.g. what if a small change were made to answer X. Do you think such a feature would be useful in your system?

Yes/No

Comments

8. Do you have any other comments?

.....

.....

Appendix B

The INVEST demonstration expert system - a guide to its use.

1. Getting started

The INVEST demonstration expert system is designed to provide a quick illustration of the features offered by the P.R.O. Expert System Toolkit focussing on the facilities of the explanation sub-system. The system operates as an automatic step-by-step guide through the facilities of the system.

A copy of INVEST is supplied on a floppy disk with this documentation. There are seven files on the diskette. These are :

INVEST.EXE	the directly executable system file.
INVEST.KNB	the knowledge base.
INVEST.SVD	a representation of the knowledge base needed by the system.
INVEST.ASO	as above.
INVEST1.REP	first automatic reply file.
INVEST2.REP	second automatic reply file.
INVEST3.REP	third automatic reply file.

To engage in a consultation with the system, insert the INVEST diskette into a disk drive of an IBM or compatible micro-computer operating under the DOS operating system, and with at least 640 K memory. It is also advisable that the computer be fitted with an enhanced graphics card and monitor, and that a printer is connected to the computer.

As there is insufficient space on the diskette for the operating system, it is essential that the computer be "booted-up" with DOS, and that the DOS prompt is set for addressing the disk drive into which the INVEST diskette has been inserted.

Type in INVEST to load the system from the diskette.

After a while, introductory text is provided on the screen. To proceed, after reading this text, press SPACE BAR.

The system then prompts :

Do you want a demonstration? (y/n) :

To which you should answer y (lower case).

A horizontally placed menu, in which the three reply files are listed, is then displayed. These reply files are text files containing responses to the subsequent system prompts. To invoke the first demonstration select **INVEST1.REP** if a printer is connected to your computer, or otherwise select **INVEST2.REP**. Selection of the correct file is achieved by moving the cursor, an inverse video bar, to the desired name using the arrow keys, and then pressing the **ENTER** key.

There follows a description of how the demonstration system operates. All that you, the user, need do from now on is press the **SPACE BAR** once you have read and understood the information displayed on the current screen.

The rest of this documentation is provided for reference purposes, and need only be read if more details on system operation are required.

This first demonstration does not cover all the features of the P.R.O. Expert System Toolkit and explanation sub-system. Further features are demonstrated by selecting the third automatic reply file **INVEST3.REP**, which can be done as a separate exercise. This demonstration is described below in section 6.

2. System prompts

After selection of a reply file, the next system prompt relates to the retrieving of previously saved results. No results file is to be used, and the system retrieves an answer of **n** (no) from the reply file.

The main consultation then begins with a listing of the available system goals, one of which must be chosen to be the current goal for the system to validate. The goal **INVEST IN A LONG TERM FIXED RATE DEPOSIT** is selected to be the current goal, the truth of which the system will attempt to validate.

The system then proceeds through a question and answer session, and following on this, invokes the routines of the explanation sub-system.

3. The question and answer session

Questions, the answers of which are needed to determine the truth of the current goal, are asked during this session. As before, answers to these questions are automatically supplied from the reply file, and a brief explanation is given.

Both character (yes/no/don't know) and numeric question are asked in this demonstration. Certainty factors are entered directly with character answers, and are derived by the system, from the difference between the minimum and maximum values, in the case of numeric answers.

After the answers to five questions have been obtained, the system can verify that the goal **INVEST IN A LONG TERM FIXED RATE DEPOSIT** is true. This fact is displayed before the system proceeds to invoke the routines of the explanation sub-system.

4. The explanation sub-system

Access to the routines of the explanation sub-system is controlled from a menu. This demonstration proceeds firstly through the routines of the rule trace, and then through those of the SQL.

4.1 The rule trace

The function of the rule trace is to list the rules and conditions that were used to show the current system goal to be true. Consider the following set of example rules which are used to explain how the rule trace operates :

```
Conclude goal X
  If A and,
  If B.
```

Conclude A
If C and,
If D.

Conclude B
If E.

If the goal X is proved to be true, then the trace can list that X was found to be true because A and B were true. A was in turn found to be true because C and D were true. B was found to be true because E was true.

4.1.1. The screen based rule trace

The first routines invoked during this demonstration display the rule trace on the screen. Rule-condition links are displayed separately one-at-a-time.

The following rule trace options are selected in the demonstration.

Firstly **r** (for rule trace) is selected from the main menu. The first goal-condition link (corresponding to the link - X was proved true because A was true - in the above example) is displayed. Control is then passed to an intermediate menu, from which one can scroll through the rule-condition links and obtain justifications thereof.

When **h** (for how the condition of the current rule-condition link was proved true) is selected from the intermediate menu, routines are invoked to show how the rule A was proved. The first rule-condition link displayed shows that rule A was proved true because C was found to be true.

When **m** (for more rule-condition links) is selected, then further rule-condition links at the current level are displayed. The rule-condition link displayed on the screen then shows that rule A was proved true because D was also found to be true.

Justification of a rule-condition link is requested when **j** (for justify) is selected from the intermediate menu.

Control is passed back to the main menu when q (quit) is selected from the intermediate menu.

4.1.2. The printed rule trace

These routines are only selected in the demonstration, if INVEST1.REP was selected, implying that a printer is connected up to your computer.

Selection of p from the main menu, results in a complete rule trace being routed to the printer. This trace contains all rule-condition links and justifications thereof.

4.1.3 The graphical rule trace

Selection of g from the main menu results in a graphical representation of the complete rule trace being displayed on the screen.

Graphical traces often suffer from being too crowded with text. For this reason, only the unique hierarchical rule number, associated with the rule-condition links in the printed rule trace, is displayed to represent the specific links in this graphical tree. Scrolling facilities exist for viewing graphical trees larger than the one displayed in this demonstration.

This concludes the demonstration of the routines of the rule trace.

4.2 The SQI

The analytical routines of the sensitive question identifier (SQI) are selected from the main explanation sub-system menu.

When a (for answers) is selected, all conditions that are questions (have no rules associated with them) are listed, with the answers attributed to them, and the extent to which the amount of certainty ascribed to them affected the goal.

Sensitive questions are flagged in this listing. It can be noted that one of the answers was found to be sensitive. This implies that the result of a numeric evaluation using this answer was doubtful due to the size of the answer range specified.

In this particular instance, there is a condition in the knowledge base which states that :

The projected inflation rate for 12 months time must be greater than 14.

As the question was answered as 13 - 15, there is uncertainty about the result of the evaluation, and the question is flagged as being sensitive.

When s (for sensitivity analysis) is selected, routines are invoked that allow for the re-answering of one particular question and the displaying of the effect of this on the system goals. A question may be chosen for re-answering if it is identified as a sensitive question, or if varying answers to the question are available from different sources.

In this demonstration, the projected inflation rate question is to be re-answered as it was identified as sensitive, and varying projections have been given in authoritative publications. The effect of these different scenarios is to be tested.

Identification of the question to re-answer is achieved using the variable names from the knowledge base.

In this demonstration, the question is re-answered three times, before the results of this sensitivity analysis are displayed.

The results show that with a slight rise in the expected inflation rate, the best investment option changes from investing in long term fixed deposits, to investing in short term deposits.

5. Exiting from the system

The explanation sub-system is exited from after the sensitivity analysis section. Control is then passed to the inference engine, Two further prompts :

Do you want to save the results? (y/n)

and

Do you want to use the system again? (y/n)

are answered n (no) from the reply files. The INVEST system is thus exited.

6. INVEST3.REP

A further demonstration system is available. This demonstration illustrates features not viewed in the first demonstration.

To execute this demonstration, re-load the system by typing **INVEST** at the DOS prompt, and select **INVEST3.REP** from the reply file menu.

In this demonstration, a different system goal, **INVEST IN PROPERTY** is selected to be the current goal. Four questions are asked by the system, and answers are retrieved, as before, from the reply file. Before answering the fourth question, an explanation of the terminology in the question is requested by replying **e** to the question when it is initially phrased.

After answering the four questions the system deduces that the current goal cannot be proved true, and a listing of the questions which resulted in the goal being invalidated are given by the explanation sub-system. In this demonstration, only one answer given is responsible for the current goal being invalidated. This question is listed along with the answer that would be needed for the truth of the current goal was to be reversed. The user now has two options. He can either observe whether the answers supplied can lead to other system goals being proved true. He can, alternatively, re-answer the particular question, if this is appropriate. This demonstration will illustrate the first option.

As no further explanatory routines are appropriate at this stage, control is passed back to the inference engine. The system leaving prompts are displayed on the screen. This time we select not to leave the system, and to retain the answers already supplied.

When the system goals are listed again, the fourth goal **INVEST IN CONSUMER ITEMS** is selected to be the current goal. The system now attempts to verify the truth of this goal using the previous answers.

This current goal is found to be true with these answers, and the routines of the explanation sub-system are invoked.

6.1 The explanation sub-system

In this demonstration, the first set of routines of the SQI are selected by choosing **a** from the main explanation sub-system menu. This results in a listing of questions answered along with the answers attributed to them, and the extent to which the amount of certainty ascribed to the answers affected the goal.

From this listing it can be noted that one of the questions, **WHAT IS THE B.A. RATE (%)** has a far larger effect on the current goal than the other questions. The reason for this can be observed by viewing the rule trace, which is invoked by selecting **r** from the main menu. The subsequent selection of **m** (for more) from the intermediate menu, illustrates that the question affected the goal directly. Selecting **j** (for justification), allows the user to examine the rationale behind this decision.

Inspection of this rule trace could leave a critical user believing that too much importance has been placed on this one question. It is thus illustrated in this demonstration, how the explanation sub-system can be used for the purposes of knowledge base debugging during the stages of knowledge base development.

7. Further consultations with INVEST

These two demonstrations have been designed to illustrate most of the features available in the P.R.O. Expert System Toolkit, and explanation sub-system, in particular.

The INVEST expert system can be consulted without requesting a demonstration reply file to be utilized. The user is then responsible for answering all the system prompts.