

**SECOND GENERATION KNOWLEDGE BASED SYSTEMS  
IN HABITAT EVALUATION**

**MARK CAIN**

**THESIS SUBMITTED IN PARTIAL FULFILMENT  
OF THE REQUIREMENTS OF DE MONTFORT UNIVERSITY  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY**

**JUNE 1993**

SECOND GENERATION KNOWLEDGE BASED SYSTEMS  
IN HABITAT EVALUATION

BY  
MARK CAIN

ABSTRACT

Many expert, or knowledge-based, systems have been constructed in the domain of ecology, several of which are concerned with habitat evaluation. However, these systems have been geared to solving particular problems, with little regard paid to the underlying relationships that exist within a biological system. The implementation of problem-solving methods with little regard to understanding the more primary knowledge of a problem area is referred to in the literature as 'shallow', whilst the representation and utilisation of knowledge of a more fundamental kind is termed 'deep'.

This thesis contains the details of a body of research exploring issues that arise from the refinement of traditional expert systems methodologies and theory via the incorporation of depth, along with enhancements in the sophistication of the methods of reasoning (and subsequent effects on the mechanisms of communication between human and computer), and the handling of uncertainty.

The approach used to address this research incorporates two distinct aspects. Firstly, the literature of 'depth', expert systems in ecology, uncertainty, and control of reasoning and related user interface issues are critically reviewed, and where inadequacies exist, proposals for improvements are made. Secondly, practical work has taken place involving the construction of two knowledge based systems, one 'traditional', and the other a second generation system. Both systems are primarily geared to the problem of evaluating a pond site with respect to its suitability for the great crested newt (*Triturus cristatus*).

This research indicates that it is possible to build a second-generation knowledge-based system in the domain of ecology, and that construction of the second generation system required a magnitude of effort similar to the first-generation system. In addition, it shows that, despite using different architectures and reasoning strategies, such systems may be judged as equally acceptable by end-users, and of similar accuracy in their conclusions. The research also offers guidance concerning the organisation and utilisation of deep knowledge within an expert systems framework, in both ecology and in other domains that have a similar concept-rich nature.

**CONTENTS**

CHAPTER 1	PAGE
INTRODUCTION: EXPERT SYSTEMS AND ECOLOGY.....	1
1.1 Expert Systems.....	1
1.1.1 The Limitations of Existing Expert Systems.....	2
1.2 Ecology.....	3
1.2.1 The Conceptual Basis of Ecology.....	4
1.3 The Focus of this Study.....	5
CHAPTER 2	
ARTIFICIAL INTELLIGENCE AND ECOLOGY.....	7
2.1 Quantitative and Qualitative Approaches to Ecology..	7
2.2 Ecological Applications of Knowledge Based Systems.	10
2.3 Habitat Evaluation.....	14
2.4 The Parameters of Ecological Reasoning.....	16
2.5 Discussion.....	17
CHAPTER 3	
SECOND GENERATION EXPERT SYSTEMS AND DEEP KNOWLEDGE....	18
3.1 Second Generation Knowledge Based Systems.....	18
3.2 Definitions of Deep Knowledge.....	19
3.3 Common Sense Knowledge and Depth.....	20
3.4 Temporality and Spatiality.....	21
3.5 Benefits and Limitations of Deep Systems.....	21
3.6 Methods and Approaches to Addressing Deep Knowledge.....	22
3.6.1 Deep Knowledge in the Domain of Physics.....	23
3.6.2 Deep Knowledge and the Representation of Causality.....	25
3.6.3 Deep Knowledge and Mathematical Formalisms.....	28
3.7 The Integration of Different Approaches to Depth...	29
3.8 Deep Knowledge and Second Generation Expert Systems.....	30

CHAPTER 4	
THE REPRESENTATION OF DEPTH IN ECOLOGY.....	31
4.1 SBF Approaches to Depth and Ecological Knowledge...	31
4.2 A Causal Approach to Representing Ecological Knowledge.....	33
4.3 The Representation of Depth in Ecology.....	34
4.3.1 Concepts.....	34
4.3.2 Paradigms.....	35
4.4 The Components of Reasoning in Habitat Evaluation within this Study.....	36
4.5 Possible Approaches to the Representation of Paradigm-Based Knowledge.....	38
4.5.1 The Representation of Knowledge in Second Generation Knowledge Based Systems.....	38
4.5.2 The Knowledge Representation used in PERSEUS.....	41
 CHAPTER 5	
THE REPRESENTATION AND HANDLING OF UNCERTAINTY.....	44
5.1 Introduction: The Ubiquity of Uncertainty in Reasoning.....	44
5.2 Means of Addressing Uncertainty.....	45
5.3 Desiderata for Reasoning under Uncertainty.....	46
5.4 Methods for Reasoning under Uncertainty.....	47
5.4.1 Numerical Methods.....	47
Probabilistic Methods.....	48
Certainty Factors.....	52
Dempster-Shafer Theory.....	54
Fuzzy Logic.....	56
Other Numeric Approaches to Uncertainty	57
5.4.2 Non-numerical Methods.....	58
Nonmonotonic and default logics.....	58
Cohen's Theory of Endorsements.....	60
Reasoned Assumptions.....	62
Fox's Semantic System.....	62
5.5 Quantitative and Qualitative Approaches to Uncertainty: Discussion.....	64
5.6 The Approach to Uncertainty in Paradigm-Based Reasoning.....	67
5.6.1 The Qualitative Conviction Calculus: 'Convictions' as the Units of Uncertainty.....	68

CHAPTER 6	
KNOWLEDGE COMPILATION AND MANIPULATION.....	72
6.1 Introduction.....	72
6.2 Reasoning with Knowledge.....	72
6.3 Control as a Distinct Type of Knowledge.....	74
6.3.1 Implicit Control in Rule-based Systems.....	75
6.3.2 Explicit Control in Rule-based Systems.....	75
6.4 Knowledge Compilation.....	76
6.4.1 Knowledge Compilation and Depth.....	77
6.4.2 Knowledge Compilation as a Cognitive Process.....	78
6.4.3 Knowledge Compilation in Machines.....	79
6.4.4 The Potential Benefits of Knowledge Compilation in Machines.....	79
6.5 Compilation within the Qualitative Conviction Calculus.....	81
6.5.1 Constructing Chains of Reasoning within the QCC..	84
6.6 Consideration of Evidence within the Qualitative Conviction Calculus.....	87
6.6.1 Identifying an Outstanding Value.....	87
6.6.2 The Prominence of a Value.....	87
6.6.3 Collating Supporting Evidence in the QCC: Knowledge Renaissance.....	88
6.6.4 The Qualitative Conviction Calculus as Directed Search.....	89
6.7 Control within the Qualitative Conviction Calculus.	90
6.8 A Comparison of the Qualitative Conviction Calculus and other Control/Uncertainty Formalisms.....	90
6.9 Information Assertion and Revocation in PERSEUS....	91
6.10 Characteristics of the Qualitative Conviction Calculus.....	91

CHAPTER 7	
THE DEVELOPMENT OF FIRST-GENERATION AND SECOND- GENERATION KNOWLEDGE-BASED SYSTEMS: A CASE STUDY.....	92
7.1 Introduction.....	92
7.2 Habitat Evaluation as a Suitable Focus for this Study.....	92
7.2.1 Is the Use of Expert Systems in Pond Evaluation Possible?.....	93
7.2.2 Is the Use of Expert Systems in Pond Evaluation Justified?.....	93
7.2.3 Is the Use of Expert Systems in Pond Evaluation Appropriate?.....	93
7.3 The Ecology of the Great Crested Newt.....	94
7.3.1 Attributes that influence the success of the Great Crested Newt.....	95
7.4 The Knowledge Engineering Process within this Study.....	96
7.4.1 Pre-development Requirements.....	96
Target Users.....	97
Expert Involvement.....	97
Hardware Specifications.....	98
Selection of Software Tools.....	98
The Pre-Development Requirements for PERSEUS.....	99
7.4.2 Knowledge Acquisition.....	101
7.4.3 The Specification and Design of HEX and TRITON..	108
7.4.4 The Implementation of HEX and TRITON.....	109
7.4.5 Comparison of the Overall Construction of HEX and TRITON.....	110
7.5 Difference/Similarities in the Functionality of HEX and TRITON.....	111
7.6 Commentary on the Development of First- and Second-Generation Knowledge-Based Systems.....	113

CHAPTER 8	
EVALUATION OF FIRST- AND SECOND-GENERATION	
KNOWLEDGE-BASED SYSTEMS WITHIN THE PRESENT RESEARCH...	114
8.1 Introduction.....	114
8.2 The Requirement for Evaluation.....	114
8.3 The Parameters of Evaluation.....	115
8.3.1 Avoidance of Bias.....	116
8.3.2 Accuracy of HEX and TRITON.....	117
8.3.3 The User Interface of HEX and TRITON.....	117
8.3.4 The Coverage of HEX and TRITON.....	117
8.3.5 The Reliability of the HEX and TRITON Software..	118
8.3.6 Attendant Materials used in conjunction with HEX and TRITON.....	118
8.4 Participating Groups in the Evaluation Process....	118
8.5 Data Collection in the Evaluation Process.....	119
8.5.1 Evaluation Experiments.....	120
8.6 Evaluation Details.....	121
8.6.1 Assessment of Accuracy of HEX and TRITON.....	121
8.6.2 Evaluation of the User Interfaces of HEX and TRITON.....	121
8.6.3 Evaluation of Explanations within HEX and TRITON.....	122
8.6.4 Evaluation of Attendant Materials used in Conjunction with HEX and TRITON.....	122
8.7 Summary.....	122



CHAPTER 9	
ANALYSIS OF EVALUATION DATA.....	124
9.1 Introduction.....	124
9.2 Comparison of User Opinions.....	124
9.3 Questionnaire-Based Evaluation of User Opinions...	125
9.3.1 User Interface.....	125
9.3.2 Conclusions and Explanations.....	128
Accuracy of Conclusions and Convictions.....	128
Explanation Facilities.....	129
Utility of Information.....	133
Educational Merit.....	134
9.3.3 Reasoning/Questioning Strategy.....	134
Robustness.....	135
Questioning Strategy.....	136
9.3.4 Overall Ratings of HEX and TRITON.....	137
9.4 Comparison of Real Pond Data and the Conclusions of the Domain Expert, HEX and TRITON.....	139
9.5 Commentary on the Findings of the Evaluation.....	143
9.5.1 Depth.....	144
9.5.2 Uncertainty.....	144
9.5.3 Control.....	144
9.5.4 User Interface.....	145
9.6 Summary.....	146
 CHAPTER 10	
DISCUSSION.....	147
10.1 Introduction.....	147
10.2 Aims of the Research.....	147
10.3 Methods Employed.....	148
10.4 Conclusions.....	150
10.5 Implications.....	152
10.6 Future Work.....	154
10.7 Summary.....	155
 REFERENCES.....	156

APPENDICES

APPENDIX A	Summary of Evaluation Data
APPENDIX B	The Development of the PERSEUS Toolkit
APPENDIX C	Reference Guide for the PERSEUS Toolkit
APPENDIX D	User Guide for HEX
APPENDIX E	User Guide for TRITON
APPENDIX F	The Pond Assessment Form used with HEX and TRITON
APPENDIX G	Evaluation Questionnaires

## LIST OF FIGURES

NUMBER	FIGURE	PAGE
1	The Basic Architecture of an Expert System	2
2	An Architecture for Interacting Knowledge Types	29
3	An Example of a Parallel Certainty Inference	48
4	Depth-first and Breadth-first Search	73
5	Increasing Levels of Detail Underlying Compilation	86
6	Overall Rating of Student and Expert Evaluators for HEX and TRITON	138

## LIST OF TABLES

NUMBER	TABLE	PAGE
1	A Comparison of AI and Numeric Methods in Ecology	9
2	Application of Expert Systems in Ecology - A Suggested Categorisation	11
3	Parameters considered by Community Ecologists	16
4	Desiderata for an Ideal Uncertainty Calculus	47
5	Objections to Probabilistic Methods of Handling Uncertainty	51
6	Stages in Decision Making in Fox's Scheme	63
7	A Comparison of Numeric and Non-Numeric Approaches to Uncertainty	65
8	The Combination of Inheritance Terms with Facts and Relationships	83
9	The Combination of Relationships within the Qualitative Conviction Calculus	84
10	Attributes that Influence the Success of <i>Triturus cristatus</i> within a Pond	96
11	Considerations preceding development of an Expert System	97
12	Facilities of Second-Generation Knowledge-Based Systems Required in the Present Research	100
13	Potential Problems in the interaction between Knowledge Engineer and Expert	101
14	A Section from the Transcript of an Early Interview	103
15	Shade 'Index' derived from Knowledge Acquisition Sessions	106
16	The Pattern of Knowledge Acquisition pertaining to HEX	107
17	The Pattern of Knowledge Acquisition pertaining to TRITON	108
18	Comparison of the Construction Processes of HEX and TRITON	111
19	The Parameters of Evaluation in an Expert System	116

## LIST OF TABLES

NUMBER	TABLE	PAGE
20	Participating Groups within the Evaluation Process	118
21	Summary of Evaluation Details	123
22	Summary of User Interface Data	127
23	Comparison of Ratings Derived from User Interface 'Scores'	128
24	Data concerning the Accuracy of Conclusions	129
25	Data of Experts' Opinions of Accuracy of TRITON-generated Convictions	129
26	Results of Data of Explanations given during Consultation	130
27	Results of Data of Explanations given at Conclusion	131
28	Results of Data of 'Default' Explanations - Criteria of Comprehensibility, and Preference	132
29	Data Summary concerning the Utility of Information given	133
30	Data concerning the Education Merit of HEX and TRITON	134
31	Results of Data concerning Robustness of HEX and TRITON	135
32	Results of Data concerning the Questioning Strategies of HEX and TRITON	136
33	Overall Rating of HEX and TRITON by Student and Expert Evaluators	137
34	Summary of Data concerning the Comparison of Real Data with the Conclusions of the Domain Expert, HEX, and TRITON	140
35	A Comparison of Conclusions/Certainties of the Domain Expert and TRITON	143
36	Number of Questions Asked, and Time Taken, by HEX and TRITON when used to Evaluate 100 actual pond sites	145
37	The Findings of this Research	150

## ACKNOWLEDGEMENTS

I wish to express my wholehearted thanks to Professor Derek Teather for his support and guidance throughout the course of this work. Thanks also to Mr Robert John for his continued support. I would also like to thank the body which funded this research, the Scientific and Engineering Research Council, and the numerous individuals that aided in the evaluation of the systems developed within this research.

Special thanks to Dr Rob Oldham at the Dept of Applied Biology, De Montfort University, for his time and co-operation, without whom this work would not have been possible.

# CHAPTER ONE

## INTRODUCTION: EXPERT SYSTEMS AND ECOLOGY

### 1.1 Expert Systems

**Expert, or knowledge based, systems (ES/KBS) emerged in the mid-1970's as a branch of Artificial Intelligence (AI) (Shortliffe, 1976). These systems were concerned with addressing tasks and problems that in human beings would require "knowledge", typically gained over a long period of exposure to the task. Knowledge can be defined in this context as 'information that allows an expert to make decisions' (Parsaye and Chignell, 1988). The construction of early, and many current, expert systems has been oriented to practical ends, using a small part of the total knowledge held by a human expert. This small body of knowledge, called heuristic knowledge, is specifically geared to problem-solving in highly-constrained contexts, and contains little explicit formalisation of the other underlying relations that also exist within the domain, and the minds of experts (Price and Lee, 1988). This emphasis on practicality has made expert systems one of the more popular and exploitable areas of AI.**

The essential components of expertise have been recognised as (1) knowledge and (2) some means of using knowledge, typically called **inference** (Forsyth, 1989). The architecture of a basic expert system includes a facility for storing the knowledge, the **knowledge base**, a means of reasoning with the knowledge, the **inference engine**, and in most cases, a suitable mechanism for communication between human and machine, the **user interface** (Forsyth, 1989). Figure 1 illustrates this design.

A variety of tools/computer languages are used to build expert systems, a common one being a **shell**, a dedicated package that provides suitable means to represent and order knowledge, an inferencing mechanism, and tools to design user interfaces.

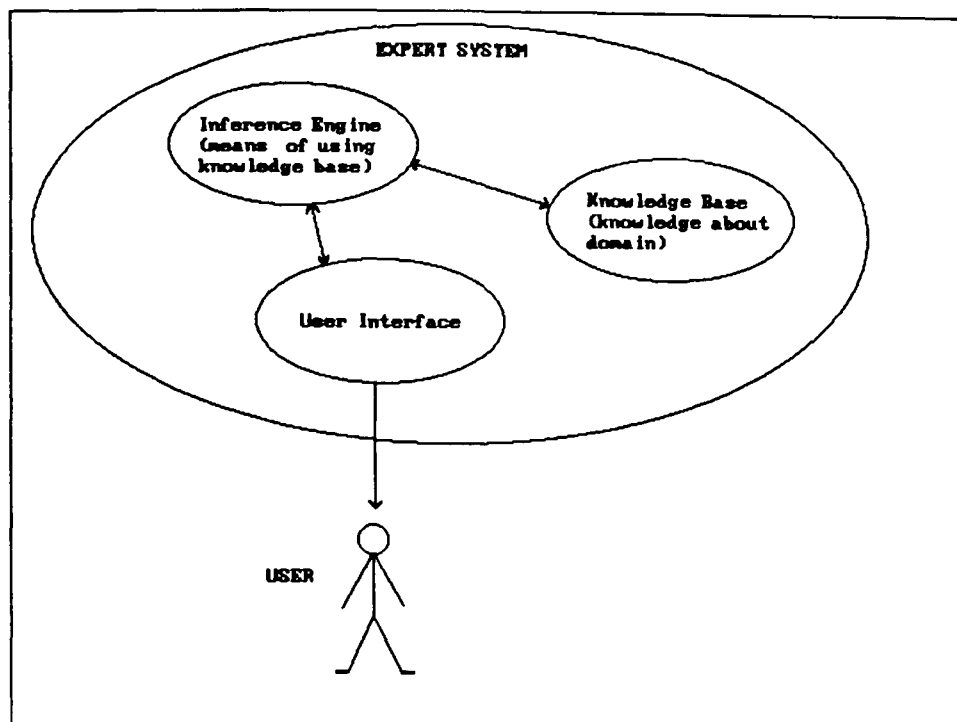


Figure 1: The Basic Architecture of an Expert System

### 1.1.1 The Limitations of Existing Expert Systems

The practical nature of expert systems has typically led their builders (**knowledge engineers**) to focus on implementing knowledge that is concerned with the task at hand only (ie heuristic knowledge). This has caused expert systems to suffer from a number of limitations, such as being usable in only the more common cases, being 'brittle' ie having a rapid decline in ability as cases move away from the norm, not being able to reason in any but the most superficial of levels, and not being able to explain its own reasoning sufficiently to users (De Kleer and Brown, 1983).

A typical example of heuristic knowledge is '*if a car's lights cannot be turned on, then the battery is flat*', a relationship that may be used by both a human mechanic and an expert system that emulates the expertise of a car mechanic. However, such a system does not contain a representation of, and therefore does not 'understand', the underlying cause-effect relations of electronic circuitry, and the structural/functional relations that exist within a motor car in the way that a human mechanic does (Price and Lee, 1988). These levels of heuristic and more fundamental knowledge have been referred to as **shallow** and **deep** respectively (Hart, 1982). Other terms for these types of knowledge include 'high-road' and 'low-road' programs respectively (Michie, 1982). Apart from lack of depth, most present expert systems to date suffer from other well-defined limitations. The handling of uncertainty, whilst addressed in present expert systems, is a contentious area, with a variety of approaches used, and no clearly superior method (Bhatnagar



and Kanal, 1986). This is further discussed in Chapter 5. Other limitations in present expert systems include; problems in control and suitable representation of knowledge (discussed in Chapter 6), causing problems such as the inability of many expert systems to be able to distinguish between cases they can solve from cases they cannot (Keravnou and Washbrook, 1989), and; having knowledge bases that are not typically reusable for different tasks within the same domain (Price and Lee, 1988). There are also clear shortcomings in the user interface of many systems, such as inflexibility with respect to both the knowledge present within the system, and in the proper matching of the user interface to the abilities and knowledge of the user (Keravnou and Washbrook, 1989). Systems that suffer from lack of depth, and the other mentioned shortcomings, are typically called **first generation**, whilst systems that address some or all of these shortcomings are referred to as **second generation** (Keravnou and Washbrook, 1989).

## 1.2 Ecology

The word **ecology** was first coined by Haeckel (1866), who defined it as '*the domestic side of organic life*'. Krebs (1985) defines ecology more formally as:

... the scientific study of the interactions that determine the distribution and abundance of organisms.

There are several other biological fields which interact with, and may come under the broad umbrella of, ecology. These include genetics, physiology, behaviour, taxonomy and evolution (Pianka, 1988).

Ecological study can have a number of different emphases (Krebs, 1985). **Descriptive** ecology is mainly natural history, and proceeds by describing various aspects of the natural world, such as ecosystems (eg rain forests), the behaviour of organisms, and the relationships between organism, ecosystem and nonliving world, and is concerned with the details of living systems. This corresponds with much of a lay person's understanding of what ecology involves. **Functional** ecology is oriented towards understanding the relationships that are present within the ecological realm. This type of ecology tends to deal with populations and communities as they currently exist, and is concerned with the measurement and quantification of interactions, processes and organisational parameters that occur. **Evolutionary** ecology is similar to functional ecology, but where functional ecology is concerned with responses to immediate changes in communities and populations (and how particular mechanisms are occurring), evolutionary ecology is involved with understanding how populations and communities have adjusted to accommodate

changing conditions, often through geological time (eg why natural selection has favoured certain changes).

Ecology is also involved with several different levels of living systems. At one end of the spectrum, evolutionary aspects means that ecologists may have to address organisms at the individual and genetic level. Behaviour focuses the attention to the organism level, whilst functional ecology concentrates on higher-level biological systems, namely populations, communities and ecosystems.

### 1.2.1 The Conceptual Basis of Ecology

Ecology may be considered a **soft science**, rather than a **hard one** such as physics, meaning that many of its important conceptual structures cannot yet be adequately defined in a mathematical way. The situation that exists in ecology, with the broad concepts underlying ecology having no definite basis in mathematics, apparently rankles some ecologists (Egler, 1986, refers to this as '*physics envy*'). This can be seen in the early literature of the application of such techniques as systems analysis and mathematical simulation in ecology. One such work, by Patten (1971), states in the first line of its preface:

This is a book of ecology in transition from a 'soft' science, synecology, to a 'hard' science, systems ecology...

Patten (1971) goes on to note that the techniques used in synecology (the study of groups of organisms in relation to their environment) are statistical and analytical, and adds that the paradigm of synecology has tended to be '*quantify and clarify*', thus lending it the air of a hard science. The systems ecology approach, involving the use of mathematical models to understand ecological processes, offers some benefits, but has by no means become a basis for the science of ecology. The synecological (analytic) approach is still dominant in ecological science. As Maynard Smith (1974) states:

Ecology is still a branch of science in which it is better to rely on the judgement of an experienced practitioner than on the predictions of a theorist.

In other words, ecology is a science where empiricism tends to dominate over theoretical considerations. The reasons why non-mathematically based concepts remain the foundations of ecology are further explored in Chapters 2 and 4.

### 1.3 The Focus of this Study

The vast majority of working expert systems built to date can be classed as first generation, with inadequacies based in any or all of the following realms; depth, uncertainty, human-machine interaction, inflexibility in the representation of knowledge, and control.

Various approaches to implementing depth in expert systems have emerged, and these are critically reviewed and compared in Chapter 3. The applicability of the existing approaches to depth in the field of ecology, and specifically habitat evaluation, are critically reviewed in Chapter 4. The inadequacies in these approaches are also discussed in Chapter 4, along with proposals for the implementation of depth in a system geared to habitat evaluation.

The techniques of handling uncertainty in expert systems, their relative advantages and limitations, and their potential use in this project are critically evaluated in Chapter 5, and are followed by a proposal for a suitable approach to handling uncertainty, relative to the configuration of ecological depth proposed in Chapter 4.

The problems of first generation systems (other than depth) are expanded upon in Chapter 3, and further reviewed in Chapters 5 and 6. Proposals for addressing such problems within this research are also made in Chapters 5 and 6.

The focus of this thesis is a comparison of first and second generation systems, in the chosen domain of habitat evaluation. The means of addressing these issues is via the building and evaluation of two working expert systems. One is a "traditional", first generation expert system, called HEX (Habitat Evaluation eXpert system), whilst the other is a second-generation system, called TRITON. The construction of TRITON was preceded by the construction of a second generation knowledge-based system development shell called PERSEUS (Paradigm-based and Experiential Reasoning System using Ecological UnderStanding), which was designed and constructed specifically for the current research.

The two systems (HEX and TRITON) share the same main objective: To be able to classify a pond as suitable or unsuitable to support the great crested newt (*Triturus cristatus*). Each system gathers the evidence needed to make an assessment via a question-answer session with the user. As well as performing assessment of a pond habitat, it was felt that each system should be able to explain why it asks certain questions, to clarify why it has reached certain conclusions, and to supply background information relating to the crested newt, features of the pond habitat, and other organisms. Such abilities make the systems not simply able to perform habitat evaluation, but available also to be used as educational tools for both students and habitat managers. The specifications set for this project include the availability

of suitable computing software and hardware, with some provision for further tools/machines required, and a time scale (fixed by funding) of 3 years involving a single expert and knowledge engineer.

Comparison of the two systems involves a review of the necessary methods used in building each system (in terms of the usual stages of knowledge engineering), along with the benefits and costs of building each system (Chapter 7). This is followed by an explanation of the criteria used in evaluation (Chapter 8), and an analysis of the evaluation of the two systems by two methods; commentary by novice and expert users, via such aspects as output, usability, and reasoning strategy, and; comparison of actual pond data with the conclusions of the domain expert and each system about suitability of the ponds for crested newts (Chapter 9).

A conjecture of this thesis is that, using the case study of habitat evaluation, the addition of deep knowledge and other features extends the capabilities of ecological expert systems, addressing many of the limitations already mentioned.

## CHAPTER TWO

### ARTIFICIAL INTELLIGENCE AND ECOLOGY

Within ecology and related disciplines, research and implementation of artificial intelligence technology has a small, but significant, presence. The nature of work in this cross-disciplinary area has many facets, ranging from problem-solving to educationally-oriented knowledge-based systems.

This chapter begins by considering the differences between traditional methods of data handling and modelling (ie quantitative methods), and methods typically associated with artificial intelligence approaches in these areas (ie 'qualitative' approaches). This leads into a review of existing applications of artificial intelligence methods in ecological domains. There is then a brief examination of some of the existing approaches to habitat evaluation (including knowledge-based systems), and their utility in the practical evaluation of habitats. Of particular interest is the way in which ecologists consider habitats in an informal or unquantified way. Consideration of such processes leads into a section containing proposals about how human ecologists reason about ecosystems, initiating a discussion that continues throughout the rest of this thesis. This discussion concerns the nature and focus of "ecological" reasoning within both the minds of human experts, and reasoning machines. The chapter concludes with a description of the limitations of present approaches to habitat evaluation, and suggests a means of overcoming these limitations.

#### 2.1 Quantitative and Qualitative Approaches to Ecology

Numeric techniques (ie mathematical and statistical) are widely-used and fundamental tools of modern ecology (eg Maynard Smith, 1974; Krebs, 1985). The wide acceptance of numeric (particularly statistical) techniques in ecological science results from their proven utility, and also their widespread use and success in science generally. Graham (1989) notes that the present preoccupation with numeric methods in science derives from '*the prevailing empiricist climate in the philosophy of science...*'.

The basic formalism in numeric methods is the equation, which, although elegant and powerful, has the limitation of only being able to treat modelled elements as numerical entities. Alternatively, knowledge-based systems methodologies allow modellers to manage elements in a different way, using symbolic relationships and manipulations. This permits modellers to address a non-numeric body of knowledge; **qualitative** knowledge, including definitions, cause-

effect relationships, and interpretation of data given to the system. Such representations may be considered similar to those naturally used by humans in reasoning, and therefore may be more accessible and instinctive to use than mathematical approaches.

Ecology (and science generally) has tended to ignore the potential of formally utilising qualitative information. Starfield and Bleloch (1983) note that:

efforts to quantify have perhaps deflected attention from the need to live with and exploit qualitative data...

The natural and meaningful content of qualitative statements provides a contrast to quantitative approaches; as Skellam (1972) states, '*Mathematical statements...are almost void of empirical content*'. The knowledge-based systems paradigm may provide a means of organising and examining ecological knowledge in novel ways, a process that Rykiel (1989) refers to as '*ecological reasoning*'.

Starfield and Bleloch (1986) have identified areas of ecology where a qualitative approach is more suitable than a numeric, including those where there is a lack of quantitative data for suitable numeric analysis and modelling, areas where resource constraints disallow the resource-intensive process of quantitative data collection, and areas where qualitative approaches are more pertinent than quantitative. There are limitations to AI approaches, however. AI methods do not provide the precision of numerical methods. The output of AI systems is usually qualitative, and as such loses the resolution that quantitative methods provide (Price and Lee, 1988). Mathematical models/approaches are usually easily tested, whilst the ambiguous resolution of an AI model may be harder to test and validate.

Table 1 provides a comparison of the relative benefits and limitations of AI and numerical approaches.

Several workers have suggested the integration of AI and numeric approaches may result in computer systems that maximise the advantages of both, whilst minimising their limitations (Rykiel, 1989; Stone and Schaub, 1990). The importance of integrating quantitative and qualitative approaches has also been recognised in other fields, such as psychology (Henwood and Pidgeon, 1992), accompanied by the insight that quantitative and qualitative information are not as distinct as they may appear. Henwood and Pidgeon (1992) suggest that:

... quantification is but one manifestation of the common practice of deriving coherent, mobile and combinable inscriptions in science. By this argument qualitative and quantitative research procedures are but different forms of the analytic practice of re-representation in science, in that both seek to arrange and rearrange the complexities of 'raw' data.

These authors go on to note the rarity in any scientific program of the *exclusive* use of either method.

Artificial Intelligence	Numerical Methods (mathematical and statistical)
1. Rare, and not well understood by large groups of people.	Ubiquitous, and well-understood by large numbers of people (particularly statistical methods).
2. Lack of Resolution	Precise.
3. Harder to validate (due to lack of resolution)	Easy to validate or invalidate.
4. Contains empirical content.	Absent of empirical content. (ie equations are independent of application's content, with the exception of certain assumptions made about the data)
5. Units of representation usually cannot be treated abstractly (as it contains empirical content).	Fundamental unit of representation (the equation) is elegant, and can be treated abstractly.
6. Allows treatment of qualitative and quantitative data.	Equation limited to numerical data.
7. AI may supplement specialist expertise.	Mathematical models require specialist expertise.
8. Certain areas of ecology best suited to AI approaches (eg interpretation of data).	Certain areas of ecology best suited to mathematical approaches (eg data analysis, optimal foraging).
9. Can be self-interpreting.	Requires intermediate analysis for interpretation.
10. Detailed numeric data, typically 'expensive' to acquire, is not necessary.	Detailed numeric data is necessary.

Table 1: A Comparison of AI and Numeric Methods in Ecology

Specific areas of ecology have been identified where qualitative methods may be more suitably employed than quantitative methods. For example, some authors suggest that the qualitative nature of knowledge-based systems may facilitate a more

suitable way to study the behaviour of organisms than numeric methods (Loehle, 1987; Folse *et al.*, 1989). One common representation in knowledge-based systems is the 'production' rule, usually of the form '*if A, then B*'. Such a representation is highly suited to modelling animal behaviour. Loehle (1987) discusses an example of seagulls raising chicks, considering time and energy expenditures. Example rules may be '*IF (Demand of food from chicks) THEN (forage ...)*', and '*IF (Inclement weather) THEN (protect chick)*'. Such a representation is both intuitively and realistically a better match of the behaviours an organism may exhibit under various conditions than a mathematical approach. Loehle (1987) also notes that AI approaches may easily accommodate conflict resolution devices to aid in the selection of which rules to use given conditions that may involve several rules (eg the chicks are demanding, and the weather is inclement).

Additionally, qualitative methods are a suitable means to model contingent behaviours within a complex system (ie behaviours that only occur given previous, possibly unpredictable, conditions). Such 'event-driven' behaviour is common in the ecological realm. For example, a bee colony swarms given certain conditions of food availability and its own size, rather than the time of day or year.

A further advantage of AI methods over numeric methods is that detailed and comprehensive numerical data is not necessary. Guariso and Werthner (1989) note that, whereas in quantitative modelling one often has to know the exact values of many parameters, this is often unnecessary in qualitative models.

Overall, qualitative methods may be viewed as a means of interpreting and understanding ecological data that supplements and complements numeric approaches. Considered thus, they extend the range of tools available to the ecologist, rather than supplanting existing tools.

Within ecological knowledge-based systems, a small but significant part of the total research concerns the evaluation of habitats. Starfield and Bleloch (1986) suggest that habitat evaluation, which often relies on informal, qualitative reasoning by an expert, is an area that is particularly suited to qualitative methods.

## 2.2 Ecological Applications of Knowledge Based Systems

Within ecological and related domains, there are a variety of AI-based applications. The work within this cross-disciplinary area is varied, and a suggested categorisation of this work is given in Table 2, whilst a full bibliography of this area (using this categorisation) is given in Cain (1993).



<p>1. Modelling/Simulation of:</p> <ul style="list-style-type: none"> <li>- Animal Behaviour.</li> <li>- Contingent ('event-driven') behaviour in individuals/populations/ecosystems.</li> <li>- Ecosystem Dynamics.</li> </ul> <p>2. Aids in the construction of quantitative/qualitative models.</p> <p>3. Facilitators in the "development" of ecological science.</p> <p>4. Tools for education/knowledge dissemination.</p> <p>5. Evaluation/Management of:</p> <ul style="list-style-type: none"> <li>- crops</li> <li>- pests and disease</li> <li>- livestock</li> <li>- wildlife</li> <li>- habitats</li> </ul>
--

Table 2: Application of Expert Systems in Ecology - A Suggested Categorisation

The vast majority of work involving ecological knowledge-based systems has been of a practical nature, typically concerning the analysis of suitable variables to suggest some final goal (ie a diagnostic approach). This work includes agriculturally-based systems involved in the management of crops, livestock, pests and diseases, to the more purely ecological domains of wildlife and habitat management. The potential use of knowledge-based systems technology in such areas has long been recognised (Starfield and Bleloch, 1983).

It is worth noting that the majority of systems described in this chapter are first-generation, typically shallow and using rule-based representations (see Section 4.5.1). Of the many systems documented in Cain (1993), only one specifically uses a non-rule-based representation for knowledge, the RANGECON system (an aid in American range management), which utilises frame-based knowledge representations (Ekblad *et al.*, 1991).

The practical nature of knowledge-based systems has led many builders of ecologically-oriented systems to centre their work on machines that are affordable to individual users, typically microcomputers, with relatively few using more expensive equipment, including UNIX-based workstations (Stone *et al.*, 1986), VAX-based machines (Roach *et al.*, 1985; Thatch and Schneider, 1988), and mainframe

computers (Rice *et al.*, 1989). Despite the fact that the majority of builders have utilised inexpensive equipment that can be accessed by potential users, there is an inherent problem in using such equipment for ecological applications. It is usually not feasible for machines to be used *in situ* ie out of doors (Roach *et al.*, 1985). This required the development of a questionnaire for recording information required by HEX/TRITON *in situ*. A copy of this questionnaire can be found in Appendix F.

Within the area of modelling and simulation, several workers have suggested that knowledge-based systems methods are well suited to modelling animal behaviour, and the behaviour of 'event-driven' systems. Other suggested uses of AI methods in modelling include the study of ecosystem dynamics. For example, Camara *et al.* (1987) have developed qualitative methods for studying the dynamics of energy transfer down food chains, whilst Loehle (1987) suggests the use of AI techniques to model the spatial aspects of ecological systems.

The use of knowledge-based systems methods to construct aids that enable ecologists to build quantitative and qualitative models, is exemplified in the work of researchers that have developed "ECO"/"EcoLogic" (eg Uschold *et al.*, 1989). ECO and EcoLogic are tools that enable users to construct ecological simulation models. There are several problems that prevent many ecologists from using mathematical modelling as a scientific tool. Such modelling is typically a difficult process without specialist skills, there is little standardisation of modelling approaches, and modelling parameters and relationships tend to be widely scattered throughout the literature. The ECO/EcoLogic program aims to bypass these problems, by using the information provided by ecologists in their own terms to aid in the model construction process.

The use of AI and knowledge-based systems in the "development" of ecological science relates to the nature of AI methodologies. AI may provide a useful means to clearly define the ideas that go to form the science of ecology, and utilise these ideas rationally. Rykiel (1989) suggests that AI technologies may be useful for the development of ecological theory in at least three ways; the organisation of computer-compatible knowledge bases, incorporating both qualitative and quantitative knowledge; rapid assessment of assumptions, hypotheses and other ideas in a theoretical context; and the determination of consequences and logical consistency of long and complicated reasoning paths. In addition, the task of building an knowledge-based system often highlights weaknesses in the knowledge about the domain concerned, and may help to target the direction of research. This has been found to be the case within the present work.

Additionally, Rykiel (1989) notes that qualitative methods are a useful intermediate stage between lack of understanding and formal quantification. He notes that there is little point in waiting for ecology to become primarily quantitative

whilst ignoring usable qualitative knowledge. Thus, qualitative methods can act as a stepping stone from incomprehension to formal quantification.

Several researchers focus on the use of knowledge-based systems methods as a means to disseminate knowledge. For example, Colfer *et al.* (1989) have developed a system (FARMSYS) for use in instruction about farming methods.

However, most of the work in ecological knowledge-based systems research involves the consideration of some task that requires management and/or evaluation. Areas of development include systems for crop management (eg apples: Roach *et al.*, 1985), pest management (eg brown planthopper infestation of rice: Holt *et al.*, 1990), disease control (eg cereal diseases: Jones, 1988), and wildlife management (eg management of deer: McNay *et al.*, 1987).

Habitat evaluation/management is also well-represented within the literature of ecological knowledge-based systems. Such systems include those for the management of rangelands using fire (Davis *et al.*, 1985), the management of estuarine habitats (Starfield *et al.*, 1989), and systems used in forestry management (Rauscher *et al.*, 1986; Rice *et al.*, 1989). The use of knowledge-based systems in habitat evaluation is further discussed in Section 2.3.

Overall, an important item that emerges from the literature on ecological knowledge-based systems is the apparent lack of systems that are actually in use. The literature contains numerous papers reporting encouraging results in early trials, with systems at the stage of final testing (eg Roach *et al.*, 1985). Other reports document systems that have reached the stage where testing is about to commence (Ekblad *et al.*, 1991). A more unsatisfactory state of affairs is the exclusion of any details of testing and actual use, other than identification of potential users (Hokans, 1984). Having identified this problem, it must be noted that some systems exist that are in actual use (with apparent success). Norton (1987) reports on several agriculturally-orientated systems, developed at Imperial College, that are in present (though limited) use in various parts of the world, ranging from applications concerned with cotton pest management in Southern Africa, to stored grain pest control in the UK.

Overall, the main body of work concerning the interaction of AI and ecological/environmental practice indicates an academic enthusiasm for the research area, with few of the systems discussed actually being used. This seeming lack of real-world applications may be an artifact of early reporting that is not followed with more detailed findings once testing has taken place, or the length of time and degree of effort that proper testing requires causing a lag between initial work reported and proper use of real-time systems. It is a general comment in knowledge-based systems literature that there is a high drop-off between the initiation and completion of knowledge-based systems projects (eg Keyes, 1989). This may be attributed to

several problems; unsuitability of the task at hand; inexperience of the knowledge engineer; interpersonal problems occurring between expert and knowledge engineer; decline in enthusiasm as project continues; and the lack of formal methods for knowledge-based system development (Kreutzer and McKenzie, 1991).

### 2.3 Habitat Evaluation

The focus of the present research is in considering the use of knowledge-based methods in the domain of habitat evaluation. It is therefore appropriate to consider the variety of methods (including knowledge-based systems) that are used in this area.

A very general method of evaluating habitats in ecology is via the use of a *species diversity index*. Several variations of this index exist, and involve the measurement of individual species within a habitat, along with the relative abundance of each (Krebs, 1985). Other more specific means of assessing habitats are available, typically involving the recording of species/features present and using a numeric scale (eg combining 'ratings' given to each species and habitat feature) to rank a habitat. Such methods are often generated by groups or organisations for local use, and tend to be either not empirically tested and/or not accepted as a universal method for habitat assessment. An example of one such development is the use of the 'habitat suitability index' or HSI (US Fisheries and Wildlife Service, 1981). HSI's integrate the concepts of habitat and carrying capacity for a given species within that habitat. These can be derived using various means, including word ranking (a site may be classed as "excellent", "good", "poor", etc), and the measurement of various numeric coefficients of the habitat (eg percentage of tree cover). This approach has been used to assess habitats in terms of such species as the red-spotted newt (Sousa, 1985), and the northern Pintail (Howard and Kantrud, 1986). It should be noted that this approach is only commonly used within the US Fisheries and Wildlife Service, with the exception of a modified version of the HSI method, used by Jeffcote (1991).

Other means of evaluating habitats include those relating to conventional ecological simulation and modelling. These include systems analysis approaches to ecosystems (Weigert, 1975), multivariate analyses (Shugart, 1984), conceptual classification (Gertner and Guan, 1990), and conventional decision support systems geared to habitat assessment (Guariso and Werthner, 1989). Again, the use of these techniques is limited, as they are based upon a requirement for (limited) specialist knowledge. An exception to this is the use of multivariate analysis, which is a relatively popular method of evaluating habitats. However, this technique is

statistical, and suffers from disadvantages associated with numeric/statistical methods, including the requirement for large, accurate data sets, and specialist skills in implementation and interpretation.

A number of knowledge based systems have been constructed that address the evaluation and/or management of specific habitats, some of which have been briefly mentioned in Section 2.2. For example, Ekblad *et al.* (1991) have developed a system geared to evaluation of range management practices, rather than management itself. On a more general scale, Loehle and Osteen (1990) devised a system that can be used in environmental impact assessment. Of some pertinence to the present work are systems that evaluate habitats in terms of suitability for a single species/taxonomic group. An example of such a system is that developed by Buech *et al.* (1990), which evaluates forest habitats in terms of their suitability for deer.

Several knowledge-based systems are being successfully used in the evaluation and management of habitats, and are listed in Cain (1993). It should be noted that all of these systems are shallow, only containing the heuristic knowledge required to do the task at hand. A number of authors have suggested the need to address the fundamental concepts that underlie ecology in knowledge-based systems (eg Coulson, Folse *et al.*, 1987), whilst other authors have proposed theoretical frameworks within which to handle ecological concepts. Davey and Stockwell (1991) offer the most complete theoretical framework of this type to date, addressing the specific concept of 'niche'. However, the work of Davey and Stockwell (1991) concerns itself with a very small subset of ecological theory, and does not readily extend to the entire area of conceptual ecology that deals with habitat evaluation (ie community ecology). These authors recognise the need to incorporate ecological 'concepts' into the reasoning processes that underlie ecological knowledge-based systems.

The first requirement of utilising such concepts is perhaps to identify these concepts. In reviewing ecological literature generally (eg Krebs, 1985), and in personal conversation with a number of experienced field ecologists, it is apparent that habitats are often discussed, and reasoned about, using community parameters, or **macrodescriptors** (Pianka, 1988). The use of a species diversity index may be considered a quantitative form of one such macrodescriptor. The use of macrodescriptors in habitat evaluation is further discussed in the next section.

## 2.4 The Parameters of Ecological Reasoning

As already noted, ecologists often reason about habitats using **macrodescriptors** (Pianka, 1988). Krebs (1985) has listed five traditional characteristics of communities that have been measured and studied, and these given in Table 3, along with other parameters that may also be considered by field ecologists (Begon *et al.*, 1986; Pianka, 1988).

Field ecologists and habitat managers often do not measure all (or any) of these parameters explicitly, but set their own qualitative status to each. Such qualitative "processing" is well suited to the knowledge-based systems paradigm, and corresponds to a set of ecological "first principles" with respect to habitat evaluation. The use of ecological macrodescriptors as a frame upon which to implement ecological deep knowledge is carefully considered in Chapter 4.

<ol style="list-style-type: none"><li>1. <b>Species diversity:</b> An index of the number of species within a habitat and the relative abundance of these species.</li><li>2. <b>Growth form and structure:</b> Relating the vegetation complement to habitat classification.</li><li>3. <b>Dominance:</b> The recording of which species exert a major controlling influence upon a habitat.</li><li>4. <b>Relative abundance of species.</b></li><li>5. <b>Trophic structure:</b> The feeding relations within a habitat</li><li>6. <b>Succession:</b> The "life cycle" of a habitat, occurring as the habitat gravitates towards a climax (stable) state.</li><li>7. <b>Productivity:</b> Rate of increase of biomass per unit time.</li><li>8. <b>Spatial heterogeneity/Structural diversity of the vegetation.</b></li><li>9. <b>Stability:</b> The tendency of a community to return to its original state given perturbation.</li><li>10. <b>Resilience:</b> The speed with which a community returns to its former state given perturbation.</li><li>11. <b>Resistance:</b> The ability of the community to avoid displacement.</li><li>12. <b>Cycling of nutrients within the habitat.</b></li><li>13. <b>Biomass:</b> The mass/energy content of organisms within a habitat or area.</li></ol>
--

Table 3: Parameters considered by Community Ecologists  
(after Krebs, 1985; Begon *et al.*, 1986; Pianka, 1988)

## 2.5 Discussion

In this chapter, the benefits and limitations of quantitative methods have been compared with those of qualitative methods. Whilst numeric approaches are precise, widespread, but data-hungry, qualitative approaches are less precise but often easier to use, requiring less intensive methods of data collection. Productive numeric approaches typically involve the use of sophisticated means of addressing fundamental processes that are occurring within an ecological system, whilst current knowledge-based systems are often heuristic simplifications, not geared to specifically address the more primary aspects of ecological processes. It is noted in Section 2.3 that various authors have suggested a means of addressing the fundamental aspects of ecological processes may be via the development of conceptually-based models in ecology (ie Davey and Stockwell, 1991; Coulson, Folse *et al.*, 1987), but very little practical work has been reported on such systems in the ecological domain. A proposal of this research is that the appropriate representation and implementation of 'concepts' within ecological knowledge-based systems are required to make such a system 'deep'. The nature of 'depth' in AI is discussed in detail in Chapter 3, whilst the means to implement such knowledge in ecological applications is proposed and discussed in Chapter 4.

## CHAPTER THREE

### SECOND GENERATION EXPERT SYSTEMS AND DEEP KNOWLEDGE

#### 3.1 Second Generation Knowledge Based Systems

The recognition that the shallowness, architecture, and user interface of first generation expert systems were limitations of the technology originated in the late 1970's (Clancey, 1979). Clancey (1979) noted that rule-based systems were too weak a representation to use for educational purposes. Other workers identified the need for better explanation facilities for users (eg Swartout, 1981). It was apparent that first generation systems became unwieldy very quickly in unusual circumstances, were not able to recognise cases where their heuristics were inappropriate, and were constrained to a very small subset of tasks otherwise done by an expert (Keravnou and Washbrook, 1989). This lack of functionality makes a first generation system inadequate in both the tasks it can acceptably perform, and in its interaction with users. Both of these facilities are recognised as requirements for second generation systems (eg Keravnou and Washbrook, 1989).

Keravnou and Washbrook (1989) have defined several of the criteria of second-generation systems. These include the addition of depth to expert systems (discussed in this chapter), as well as the inclusion of more sophisticated methods of handling knowledge within a system than presently exist, and improvements in human-computer interaction. More sophisticated and explicit ways to handle uncertainty are also identified as key to improving existing expert systems (Buchanan and Smith, 1989; Mamdani and Efstathiou, 1985), and this issue will be discussed in Chapter 5.

This chapter considers the issue of depth, benefits that may result from its inclusion into a knowledge-based system, and current methods available for representing deep knowledge.



### 3.2 Definitions of Deep Knowledge

Several workers have ventured definitions of depth. Steels (1990) defines deep knowledge as that which makes explicit the models of the domain and the inference calculus that operates over these models. Klein and Finin (1987) recognise a possible continuum of depth in the definition:

Consider two models of expertise  $M$  and  $M'$ . We will say that  $M'$  is deeper than  $M$  if there exists some implicit knowledge in  $M$  that is explicitly represented or computed in  $M'$ .

Washbrook and Keravnou (1990) point out that this definition is relative and not absolute. Two comparable systems may be 'deep' (in terms of containing active causal, structural, or other representations) but one may be more shallow than the other (or both may contain shallow parts that the other stores more deeply). The terms shallow and deep are neither quantitative nor relative in the way that the definition of Klein and Finin suggests, but are in fact classificatory. A shallow system can best be defined by what it does not represent - deep knowledge. A deep system is one that contains an active representation (ie available for computation and manipulation) of the components and processes within a domain. Bylander (1990) notes that there are two aspects to deep knowledge, that of the representation scheme, and that of the reasoning method. He defines each aspect in the following way:

A representation is 'deep' with respect to a kind of phenomena if the representation describes the properties and relationships by which the phenomena interact.

A reasoning method is 'deep' with respect to a kind of phenomena if the method reasons based on how the phenomena interact.

Although deep knowledge is not explicitly represented in shallow expert systems, shallow knowledge is itself selected and grouped instances of deep knowledge (Steels, 1990). Steels (1987b) points out that this shallow, heuristic knowledge allows shortcuts, bypassing cause-effect and other relations, and sacrificing the clarity of relationship for speed. An expert can and does use heuristics but is able to utilise other relations when necessary (Anderson, 1983). Additionally, an expert has some idea of the relative importance of each relation, and can therefore disregard those that are trivial for any particular task at hand.

### 3.3 Common Sense Knowledge and Depth

One particular problem recognised in first generation systems is their lack of knowledge about the real world, typically referred to as "general knowledge", "consensus reality knowledge", and most commonly as "common sense" (Lenat and Feigenbaum, 1991). Each human being has a vast store of general knowledge, that is rarely talked about, but commonly used. Examples include; water flows downhill; doing work requires energy; people live for a single, contiguous, finite interval of time; and so on. Expert systems typically do not contain such knowledge, and it is not taught formally to humans. Kuipers (1979) notes that common sense is usually applied and acquired with little concentrated effort, and it allows the possessor to meet everyday demands involving physical, spatial, temporal, and social aspects. The lack of effort that goes hand-in-hand with common sense leads Kuipers (1979) to refer to it as working under an *opportunistic* mode, requiring little attention or energy from the user.

The interrelation of common sense and deep knowledge is complex - the two areas are heavily integrated. Knowledge that is common sense in a human being may be considered "deep knowledge" when present within an expert system, as that knowledge allows the system to reason about the interactions of domain phenomena at a fundamental level. The common sense statement that "fish cannot survive in a pond that has completely dried up" may be a necessary component of the deep knowledge of a system performing pond evaluation, but it is a superfluous (and probably counterproductive) piece of knowledge to include in a system built to diagnose liver complaints. Within any expert system, the deep knowledge required is made up of a tiny fraction of consensus knowledge (eg a pond is a water body), as well as more specialised knowledge about non-common aspects of the domain (eg ecosystem productivity is typically restricted by a specific limiting factor).

The requirement to accommodate a large body of common sense knowledge into expert systems, over and above that necessary to perform deep reasoning in a strictly defined domain, has been termed the **Breadth Hypothesis** by Lenat and Feigenbaum (1991). An ambitious proposal to embody such knowledge within a working system, called CYC, is currently underway (Guha and Lenat, 1990). However, to embody the consensus knowledge of humanity is well outside the scope of this study. Many previous projects have failed because the breadth of knowledge specified for the system has been too wide - many authors consider that a defining criteria of an expert system is that the domain it addresses must be both small and well-defined (eg Parsaye and Chignell, 1988).

### 3.4 Temporality and Spatiality

Other aspects of reasoning that are closely related to both depth and common sense reasoning are temporal and spatial reasoning. Virtually all reasoning about the real world contains some aspect of time and/or space. The complexity of both of these types of reasoning, along with the existence of well-developed lines of research (Allen, 1984; Cohn, 1989; Forbus *et al.*, 1987), means that they have not been explicitly addressed within this research. It is likely, however, that temporality and spatiality are implicitly represented in all expert systems, just as they are implicitly present in much human reasoning. Loehle (1987) notes that spatiality is implicit in many ecological ideas, such as migration and dispersal. Concepts of distance, position and arrangement in space are central to human reasoning.

Temporal reasoning can be handled in two ways; the first is by dividing the domain into appropriate 'time-slices', noting what each parameter of the domain does within that slice, and assuming that what occurs is constant for any given number of time slices (Allen, 1981). The other extreme is to recognise that events dictate the action of following events and processes, and is essentially an 'event-driven', or contingent, way to consider temporality (Attarwala and Basden, 1985). Temporality has a direct relationship with many ecological ideas. For example, all ecological communities are subject to succession, the tendency of a habitat towards a stable and predictable state. Such a process is essentially 'event-driven'. Event-driven temporality is closely related to contingent causality. When a phrase such as 'A causes B' is used, it is implicitly understood that B does not occur prior to A. Temporal representations have been investigated by several workers via temporal logics (Allen, 1981; McDermott, 1982).

### 3.5 Benefits and Limitations of Deep Systems

Deep systems may provide a solution to some of the problems that occur with shallow systems. Deep systems should have a gradual degradation with more and more unusual cases, be able to handle more complex problems, be able to both reason and explain itself from first principles when required, have a knowledge base that is reusable, and by virtue of this last point be available to perform several tasks within the same domain (Keravnou and Washbrook, 1989). Indeed, it is possible that some complex tasks may only be successfully performed by deep approaches (Guida, 1986): Travé-Massuyès (1992) notes that shallow knowledge is typically derived from *a priori* known situations. Other fields may be so new as to not have any experts, and deep systems may be a useful way to generate shallow, problem-

solving knowledge. Becker *et al.* (1989) have suggested the generation of shallow knowledge from deep as a means of bypassing the need for the acquisition of heuristic knowledge from a practising expert.

However, the finer grain of knowledge associated with a deep system may possibly require more input of time and energy by the knowledge engineer in developing the system in all its aspects, including knowledge acquisition, representation, and validation, a point further examined in Chapter 7. Cohn (1985) notes that one of the biggest problems in developing deep systems is that of 'conceptualisation'. In other words, what are the most efficient and effective ways to express the relationships between the objects or elements in a domain? Many domains do have a set of concepts and concept-relations already developed that may be exploitable in order to overcome problems of conceptualisation. In ecology, for example, there are concepts relating to community activity (eg productivity, stability), interspecies relations (eg competition, predation), and individual-level aspects (eg behaviour patterns, adaptation to local conditions).

Further problems may occur because systems reasoning from first principles are likely to take longer to reach a conclusion than those utilising heuristic or shallow representations (Price and Lee, 1988). This may be offset by correct integration of compatible shallow and deep knowledge within a system (Hart, 1982; Steels, 1987a), or via processes that generate shallow knowledge (ie heuristics) from deep knowledge, a process called **compilation** (Bylander, Smith *et al.*, 1988). Compilation is more properly discussed in Chapter 6.

Coiera (1992) notes that the approach to depth in a knowledge-based system depends on the problem-solving needs it addresses, and notes that '*a battery can be seen as a set of chemical reactions or as a supplier of electromotive force*', suggesting the focus of depth is both domain and task dependent. Shallower reasoning is typically more efficient than deep, and usually easier to implement, but may typically be less accurate and robust than a deep system. However, it must be kept in mind that "*even the deepest models are abstractions of reality*" (Bylander, 1990).

### 3.6 Methods and Approaches to Addressing Deep Knowledge

In AI, three schools of work may be identified that address the issue of deep knowledge. The first two are concerned with the formal representation of processes, events, and entities, using qualitative techniques (Kuipers, 1987). The first of these is concerned with the representation of activities and systems constrained by physical laws, and the other is based on the representation of cause-effect relations in a

variety of fields (Kuipers, 1987). The third school of thought concerns the use of mathematical methods as a means to represent deep knowledge (Hart, 1982). These three approaches will now be considered in detail.

### 3.6.1 Deep Knowledge in the Domain of Physics

Research into methods for qualitatively representing the deep knowledge of systems constrained by physical laws began in the late 1970's. Hayes (1979) proposed an approach to the encoding of physical laws in a qualitative way within computing systems, called **naive physics**. Hayes' work has been more theoretic in nature than many other researchers in the field, and his naive physics approach has been concerned with how people reason about everyday occurrences that are constrained by the laws of physics (such as throwing a ball). The majority of workers, whilst paying heed to the theories of Hayes, have tackled the problem from a more practical angle, being concerned with how such information relates to the knowledge of experts (engineers and physicists), and how to efficiently and effectively draw conclusions from that body of knowledge (Forbus, 1985). This approach has typically been called **qualitative physics** (Kuipers, 1987). Such work facilitates the identification of the core knowledge that underlies physical intuition, an area that is arguably the 'first principles' of a wide range of domains concerned with systems constrained by physical laws (such as locks, motor cars and coffee machines). Within these domains, such knowledge can be considered 'deep', using the definitions stated in Section 3.2. The representation and use of such knowledge allows reasoning when heuristic knowledge fails, and can provide explanation and justification for its conclusions and line of reasoning.

Qualitative physics has been implemented in a number of ways. Its main targets have been in the replication, and understanding, of the **structure**, **behaviour**, and **function** of physical systems, and how these different levels of expression interact. Chandrasekaran (1991) refers to such approaches as **SBF** systems, whilst other authors have referred to these approaches as '**qualitative kinematics**' (eg Faltings, 1992). Structure is a description of the physical entity that is being modelled. Function is the goal of the entities modelled within the system, typically from a human perspective (Keuneke, 1991), whilst behaviour is the overall activity of the entity. Whilst superficially similar, functionality and behaviour operate at a different level of description. For example, the behaviour of the hour hand on a clock can be described in terms of rotation on the clock face, whilst its function is to indicate the hour to any observer (Bobrow, 1985). Its structure is typically the physical attributes of the hand, depending on what is necessary to model. This may

include such details as dimensions, density, connection to other entities, and so on.

The interaction of structure and behaviour is a key area of research in the field of qualitative physics (eg Chandrasekaran and Milne, 1985; Kuipers, 1984). The inferred derivation of behaviour from structure has been called **envisionment** (De Kleer and Brown, 1981). Envisionment aids in encoding, and ultimately understanding, the multiplicity of processes that occur in complex devices, and identifies strong associational links that exist between components within these devices (Chandrasekaran, Bylander *et al.*, 1985).

Within the literature of qualitative physics, there are three main approaches to addressing physical processes (Bredeweg and Wielinga, 1988; Cohn, 1989). The first is the **constraint-centred** approach, in which a system is described by a homogeneous set of constraints (eg Kuipers, 1986). A model consists of a number of parameters, with clearly specified relationships between them.

The second approach is the **component-centred**, or device-centred, one; here a system is modelled by creating discrete components, which are then connected explicitly. De Kleer and Brown (1984) typify such work. Within this approach three types of constituent are identified; *materials*, *components*, and *conduits*. Materials are manipulated by components, and conduits transport materials between components, but do not influence them. For example, in an electrical system, the electrical current would be the material, a light bulb socket would be the component, and wire would be the conduit. De Kleer and Brown (1984) are particularly interested in generating an approach to qualitative physics that will provide explanation, and prediction. One main aspect of this approach is the creation of components from pre-formed libraries of components/devices. Such a generic approach allows easy creation of specific devices. Additionally, it allows adherence to the "no function in structure" principle advocated by De Kleer and Brown (1984). This means that the builder of such qualitative models should not be influenced by the function of the device; its behaviour should rely only upon its structure, not on local and possibly idiosyncratic criteria of functionality.

The third, and most sophisticated, approach is the **process-centred** one, which, like the component-centred approach, explicitly models discrete components, adding a dimension by allowing the modelling of processes that may act upon components, and temporal change. Forbus (1984) is a principal worker in developing this approach, called **Qualitative Process Theory (QPT)**. QPT allows the expression of processes that act on and relate objects. For example, the pressure of a gas in a closed container will increase if the amount of energy (ie heat) in the gas increases.

All of the previous approaches to qualitative physics use models with components that may have continuous values within a discrete range of qualitative

descriptions (eg voltage within a device is continuous in the sense that there is an indefinable number of distinct voltage values within a specified range). There exist other qualitative systems where state variables have discrete values. Such physical systems include digital circuitry, and the modelling of such systems have been accomplished by Davis (1984), who used multiple structural criteria, such as topological, thermal, and electromagnetic, to derive behaviour from structure in a digital circuit troubleshooting system.

The different approaches within qualitative physics reflect the level of modelling. The constraint-centred approach is concerned with generating a qualitative calculus that may be easily abstracted to different uses in much the same way as a mathematical calculus. The process-centred approach provides a closer and more thorough analogy to real processes, components, and events, but is harder to implement and utilise. The approaches of component-centred and process-centred methods concentrate on modelling systems in terms of entities within the system, and the interaction of these entities. This satisfies the definition of depth given by Bylander (1990), as considered in Section 3.2.

Recent work has focussed on the integration of these approaches. De Kleer (1993) has suggested an integration of different approaches to qualitative physics, by the development of a common language for describing the physical world.

Various systems have been built based upon the SBF approach to depth. Examples not mentioned already include knowledge-based systems for the diagnosis in electrical devices (Genesereth, 1984; Ng, 1991), power plant failures (Herbert and Williams, 1987), and the maintenance of a gun turret (Whitehead and Roach, 1990). The utility of this approach to an ecologically-oriented knowledge based system is critically reviewed in Chapter 4.

### 3.6.2 Deep Knowledge and the Representation of Causality

Causal reasoning is a key constituent to human thinking; it has been referred to as the "*cement of the universe*" (Shoham, 1985). The idea of causality, or cause-effect relations or processes, is multifaceted and context-dependent. A cause can be thought of as some process, event or entity that has an effect. For example, the presence of predaceous fish in a pond will "cause" the site to become unsuitable for a viable crested newt colony, as the fish are likely to eat all of the larvae.

Causality is also dependent upon the frame of reference. In an ecological domain, the causal expression concerning fish predation and pond unsuitability may be acceptable as such; but at a more fundamental level of biological understanding,

such as the physiological, there is no relevant causal relation expressed. At the physiological level, reference would be made to the death of the larvae by dismemberment (in the jaws of larger fish), suffocation, or digestion. It should be noted, however, that it may be possible to infer the causality from one level to another, ie from the physiological to the ecological level. For example, it is possible to infer the diet of a mammal from its dentition. However, it is not always possible to infer all higher-level behaviour from lower, due to emergent features. One could not easily predict the composition and inherent interactions of a pond community by knowing the physiological details of each species, for example. These issues will be discussed further in Chapter 4.

There are several advantages to using causal relations as the basis for a deep system. These include derivation of shallow knowledge from deep (eg it is straightforward to see, if A causes B, and B causes C, that A has some effect on C), and the subsequent availability of the deep structures for explanation purposes (Chandrasekaran and Mittal, 1983b). Most human experts explain their lines of reasoning in causal terms, and an emulation of this approach to explanation appears a viable one for expert systems (Shoham, 1985). Other reasons for using a causal approach include its ability to be used in different problem solving tasks within the same domain (Console *et al.*, 1989).

Causal knowledge is typically represented as a network of nodes. The nodes represent a state, event or object, whilst the links represent a variety of causal relations. Rieger and Grinberg (1977) suggested an extensive representation scheme of causality, proposing a remarkably coherent set of causal relations, including; expressing continuous and discrete causality (eg effects of long-term competition between two species, the predation of one organism by another), indirect causality (where the intermediate elements are hard to express, unknown, or otherwise not possible to articulate), antagonistic interactions, multiple causality between objects, and threshold causality (eg water boils at 100 degrees Celsius).

Many of the earliest systems that can be called deep are causally-based. Early systems implemented 'causal association networks', and include CASNET (Weiss *et al.*, 1978), ABEL (Patil *et al.*, 1981), and CADUCEUS (Pople, 1982). Examples of more modern causal networks include a causal simulation system, CAUSIM (Fu, 1991). Relations within such networks are purely causal, and their interface with the user is via associational objects, called 'observations'. The values of these observations are derived from the user, used to set a status for appropriate objects within the causal association network, allowing others to be instantiated from these. The construction of such causal nets presupposes the existence of a knowledge base of potential states, events, actions, and cause-effect relations.



A further aspect of some causal systems is the incorporation of 'concepts' into the knowledge base. Chandrasekaran and Mittal (1983a) have developed a group of medical expert systems, collectively called MDX, that contains various medical concepts, such as particular organs (eg liver, heart), diagnostic tests, deformities, and diseases (eg infectious hepatitis, cholestasis due to biliary stones, etc). Such concepts are tagged with empirical details about how to relate various kinds of *findings*, such as symptoms, lab data, historical information, etc. The Oxford System of Medicine uses a more sophisticated approach to handling such concepts, using higher-level relationships, such as 'A disease requires treatment' to control the reasoning process (O'Neil, Glowinski and Fox, 1989). Horn (1991) has developed a similar approach, using 'generic' concepts, such as 'manifestations' (ie symptoms and signs), 'disease descriptions', and 'diagnostic procedures'. The use of concepts in deep reasoning is discussed further in Chapter 4.

Chandrasekaran and Mittal (1983b) argue that causal networks are inadequate as they contain no explicit information about the structural and behavioural assumptions that underlie the causal links. It is a general criticism of systems that perform causal reasoning that there is an absence of structural representation, and an inability to derive behaviour from structure where it is represented (Kuipers, 1987). However, these criticisms are typically levelled at the earliest work on causal representations. More recent work has integrated causal and structural knowledge. For example, in the medical domain, several groups have been concerned with the use of representations of a patient's pathophysiological state in the reasoning process (Console *et al.*, 1989; Horn, 1991; Patil *et al.*, 1981; Szolovits, 1985), and there exist systems that use these representations for educational purposes (Dugerdil and Guillod, 1990; Kunstaetter, 1987). Bylander, Smith *et al.* (1988) discuss the derivation of shallow knowledge from a structural/functional and causal model of the cardiovascular system, whilst other workers have built comparable systems to model the heart (Bratko, Mozetic *et al.*, 1988; Hunter *et al.*, 1991; Mozetic, 1990; Shibahara *et al.*, 1983; Wildman, 1992).

A causal approach to representing depth in an ecological knowledge base system is discussed in Chapter 4.

### 3.6.3 Deep Knowledge and Mathematical Formalisms

Although relatively little work has taken place in the area of representing depth mathematically, some workers have suggested that mathematical representations may be utilised as the basis of a deep knowledge system. Hart (1982) notes that the most detailed representation available to model a reservoir is a set of partial differential equations. Mathematical simulation has a longer history and more developed theoretical framework than those approaches already discussed (Patten, 1971), and has the advantages and problems that accrue with using the equation as the basic unit of relationship. It may be quick, precise, elegant, and powerful, but it is data-hungry, opaque in terms of causality, structure, behaviour, or other non-mathematic relationships, and it has no 'understanding' of the concepts being modelled. Mathematical simulations cannot explain themselves, and they have no explicit or accessible representation of first principle reasoning. Fox, Barber *et al.* (1980) comment that systems of formal, mathematical appearance are not immediately comprehensible to users. Bylander (1990) notes that before a quantitative representation is used, a variety of qualitative steps must occur, typically within a human mind: an understanding of the situation involved; the mapping of the situation onto a quantitative model; and the interpretation of results. Quantitative reasoning is therefore a supplement for other reasoning methods, rather than a replacement. Proper combination with qualitative processing may be a means of utilising quantitative processing for the representation of deep knowledge. Chandrasekaran, Smith *et al.* (1989) note that:

...while mathematical models are useful, they are maximally effective under the control of some other mental process, rather than standing alone.

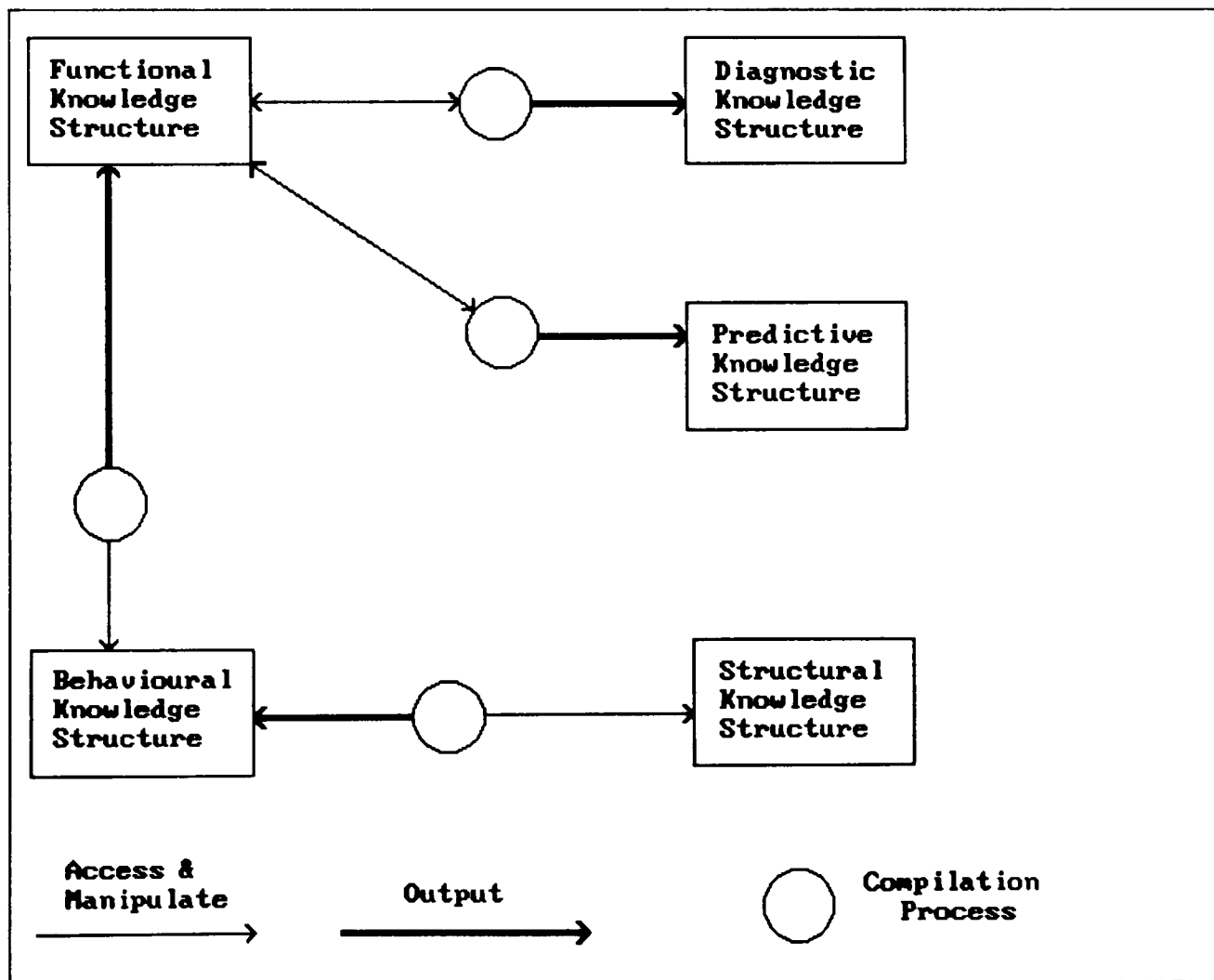
Conley and Sengupta (1989) have developed an approach that uses a mathematical model with a qualitative interpreter in the field of demography, whilst Kunz (1983) has developed a system to address renal physiology that combines causal relations and mathematical models. The most advanced integration of this kind to date appears to be the work documented by Long (1991), which integrates causal, SBF and numeric models into a single system addressing patient management under heart failure, with each model performing separate tasks.

### 3.7 The Integration of Different Approaches to Depth

Different models may be applicable to different domains and at different levels. Proper integration of different aspects of depth is likely to be a target of future research into deep systems. Steels (1990) speculates about such a situation, giving the example of a diagnostic system:

We might express a structural model describing part-whole relationships between components and subsystems, a causal model representing the cause-effect relationships between properties of components, a geometric model representing the spatial relations between the components, a functional or behavioural model representing how the function of the whole follows from the functioning of the parts, a fault model representing possible faults and components for each function that might be responsible for the fault, and an associational model relating observed properties with states of the system.

Minsky (1991) notes that the handling of different types of reasoning is likely to entail the use of different representations in a single architecture. Chandrasekaran, Smith *et al.* (1989) offer a diagrammatic representation of a potential architecture by which different types of knowledge may interact (Figure 2).



**Figure 2: An Architecture for Interacting Knowledge Types**  
(after Chandrasekaran, Smith *et al.*, 1989)

### 3.8 Deep Knowledge and Second Generation Expert Systems

The nature and utility of deep knowledge, in its various forms, has been discussed with respect to reasoning within this chapter, along with an introduction to the wider perspectives of second generation expert systems.

Deep knowledge is not a homogeneous entity that can be uniformly represented and handled. It consists of understanding a variety of different processes and interactions occurring within the real world, at the level of focus of the intelligent system (human or machine) involved. However, certain standard ways of viewing deep knowledge have arisen in the field of artificial intelligence, in particular structural/behavioural/functional, causal and (to a lesser extent) mathematical models. A task of the present research is to identify a suitable means to define and represent deep knowledge within the ecological domain, using existing or novel means (Chapter 4).

A further task of the present research is to address the wider issues of second generation expert systems in terms of a deep ecological expert system (ie uncertainty, control, and certain aspects of human-computer interaction). This involves a proper consideration of these areas with particular emphasis on ecological depth, and its representation.

Issues of uncertainty are addressed in Chapter 5, with specific stress on integrating the representation and use of uncertain knowledge and deep knowledge. Similarly, issues of control and the certain aspects of human-computer interaction (eg explanation and justification of knowledge used and conclusions reached) are addressed in Chapter 6, with respect to the formalisation and utilisation of deep knowledge and uncertainty.

## CHAPTER FOUR

### THE REPRESENTATION OF DEPTH IN ECOLOGY

The main approaches to addressing depth in knowledge-based systems are reviewed in Chapter 3. The suitability of each for the representation of ecological knowledge with respect to habitat evaluation will now be considered. It should be borne in mind that reasoning using the deep knowledge of a domain may be considered equivalent to reasoning from "first principles" (Abu-Hanna and Gold, 1990). The following sections examine the suitability of the various approaches discussed in Chapter 3 to reasoning using ecological "first principles", followed by a proposal for the use of an extended causal approach for this purpose.

#### 4.1 SBF Approaches to Depth and Ecological Knowledge

One of the main goals of the structural/functional/behavioural approaches to modelling depth has been the derivation of the behaviour of an entity from its structure (eg Chandrasekaran and Milne, 1985). This has worked well where the entities involved have been simple networks of interaction, such as man-made systems, or non-manufactured systems where activity is constrained solely (or mostly) by physical laws. In simple systems, knowledge of physical laws is such that the physical characteristics are easily measured or inferred. In more complex man-made systems, the structure has typically evolved with a fixed function or set of functions in mind, and has been designed and refined by human engineers. Such manufactured entities typically allow straightforward derivation of behaviour from structure. For example, given the structure of a head lamp, the components involved, and background knowledge in electrics, light reflection, and so on, it is possible to derive its full behaviour from its structure. However, are the components of ecological systems comparable to such structures?

There is a much cruder understanding about both structure and process in living systems than non-living. Though expanding, knowledge of the processes that occur at all levels of biological activity, from molecular to ecosystem-specific, is relatively poor, particularly in terms of the interaction between structure and activity of components. It is a widely-accepted theory in science that living organisms are patterned by evolution, operating via Darwinian selection, and not to some predetermined blueprint, as with man-made systems. Secondly, if it is supposed that an appropriate grain of structural knowledge is available, would the inference of activity from structure be suitable? It is possible to infer some aspects of grosser-level activity from organism structure at present. For example, the overall

physiology of a fish may permit an inference that it can only survive for any reasonable length of time in a liquid medium, and finer level physiological detail may allow further inferencing about the make-up of the medium (eg composed mainly of water, with certain minerals present, etc). However, a vast amount of knowledge would need to be handled to derive activity from structure, and this would be both highly inefficient, and require daunting efforts from the knowledge engineering team. Leading on from this, it must be noted that the properties of a living system are not derived solely from structure. In particular, **emergent** behaviour is found within biological systems. Begon *et al.* (1986) explain emergent behaviour using the following analogies:

A cake has emergent properties of texture and flavour that are not apparent simply from a survey of the ingredients. A sandy beach has emergent properties in the arrangement of sand grains and pebbles of different sizes that gives it pattern.

Whilst a beach is reasonably elaborate, how much more complex is a living system, with levels of activity ranging from the molecular, to the complex actions of higher animals? The development of **chaos theory** has well illustrated that even simple systems (of any type) can be highly unpredictable in their gross activity (Gleick, 1987). In discussing a structural/behavioural/functional approach, Kuipers (1987) notes:

The success of this approach depends strongly on the modular decomposition; a reasonable assumption in the case of designed structures such as digital circuits but less reasonable in, say, biology.

A further aspect of emergent behaviour is the nature of **progressive systems**. Living systems alter through time, under the auspices of natural selection, climatic change, and chance. Successive change through time, in any field where historical transition is considered, is a principle called **contingency** (Gould, 1989). Gould (1989) states:

A historical explanation does not rest on direct deductions from laws of nature, but on an unpredictable sequence of antecedent states, where any major change in any step of the sequence would have altered the final result. The final result is therefore dependent, or contingent, upon everything that came before.

Neither emergent nor contingent activity is easily or suitably handled by addressing structure (and therefore by the SBF approach). Such activities may be constrained by the structural/physical foundations of the system's components, but

are not predictable from these foundations.

One of the principle activities of ecological science is in exploring the relationships that occur within ecological systems in a functional way, measuring and quantifying interactions, processes and organisational parameters that occur in nature (see Section 1.2). This functional approach does not have its focus on the structural aspects of a system's components, although such knowledge may be of use to an ecologist. Ecologists use physiology and anatomy as a tool to discover more about an organism's interaction with the environment, but typically only as a means of cross-referencing details from other sources or techniques. In much the same way, the focus of a medical doctor in understanding and addressing disease is aimed at the processes that are occurring within the patient. Whilst comprehension of the structure of the disease organism may aid in the overall understanding of what is occurring, it is not a fundamental part of a doctor's reasoning about the activity and handling of the disease. Thus, ecologists do not use structure as a "first principle" of ecological thought, but rather as a (powerful) means to augment their understanding of system functionality.

#### 4.2 A Causal Approach to Representing Ecological Knowledge

Ecologists generally do not try to infer activity of a system from its structure, but observe the activity of ecological systems, try to recognise patterns in this activity, and theorise about what may instigate or constrain such patterns. This approach shares much with the methods of causal/medical systems, where the system being observed is the human body.

Functional ecology typically commences with the collection of empirical data. When data is analysed, and reveals empirical associations, scientists try to explain such associations via possible *cause-effect* relationships. Such data can involve emergent, contingent, or a number of other activities occurring within the ecological system (eg extinction rates, predator-prey interactions, etc). For example, contingency is evident in a theory of the evolution of life on earth, "punctuated equilibrium", which notes that the mass extinctions that have occurred in history have *caused* a large set of ecological niches to become available, into which new organisms may have evolved (Eldredge, 1987). Other examples of contingency include explanation of the behaviour and morphology of animals with respect to evolution, and the principles of colonisation of new habitats.

With emergent activity, the description of cause-effect relationships between system components has led to these components becoming ecological 'concepts', a formal set of expressions that are used by ecologists. These terms are the

macrodescriptors which ecologists use as templates of investigation and reasoning at the ecological community level, and include such terms as 'productivity', 'biomass', and 'species diversity' (Pianka, 1988).

### 4.3 The Representation of Depth in Ecology

It is a proposal of this thesis that the use of ecological macrodescriptors (derived from causal explanation of empirical associations) within a suitable reasoning system equates to the deep knowledge used by a community ecologist. Furthermore, these concepts are not treated as abstract entities, but are understood (possibly subconsciously) as nodes in a loose system of belief held by ecologists. Such systems of belief within scientific disciplines are called **paradigms** (Kuhn, 1970), and a title for the utilisation of this type of deep knowledge is proposed; **paradigm-based reasoning**. Paradigm-based reasoning may be considered a specific variant of the process of 'cognitive emulation' (ie emulation of a domain, rather than an individual), proposed by Slatter (1987b).

The use of macrodescriptors requires justification, however, and the following sections will discuss the nature of both concepts and paradigms.

#### 4.3.1 Concepts

'Concepts' refer to any discrete entities that are understood by human beings. Locke (1961) suggests that humans make use of mediating 'ideas' (concepts) that integrate reality and mental understanding. Words are equated to these 'ideas', and more general words (eg 'dog') typify more general ideas. Thus, an internalised concept of 'dog' allows us to identify and act accordingly towards real dogs. Concepts are therefore a means for human beings to understand and efficiently interact with the world. As Smith and Median (1981) state, '*Without concepts, mental life would be chaotic*'. The nature of concepts is a complex and controversial field, and full exploration of this subject is beyond the scope of this study. The subject is explored in Johnson-Laird (1983), and Smith and Median (1981).

The present research is concerned with the utility of concepts in human and machine reasoning. Wittgenstein (1953) set out the proposition that;

Concepts lead us to make investigations, are the expressions of our interest, and direct our interests.

Specialist fields, particularly those based in science, generate their own



concepts in trying to understand and analyse the processes occurring in their subject areas. Examples include the cell doctrine in biology, plate tectonics in geology, the germ theory of disease in medicine, and atomism in physics (Newell and Simon, 1976). Newell *et al.* (1976) note that such concepts are frequently qualitative in form, are seen everywhere in science, and that some of the greatest scientific discoveries can be found among them. Simon (1979) notes that such qualitative laws are sometimes later expanded into quantitative laws, sometimes not, but at any given instant, they constitute a substantial part of basic scientific knowledge.

Concepts in both everyday life and specialist fields are useful for a variety of reasons. Smith (1989) notes that a number of disciplines concerned with cognition consider concepts to be the basic constituents of thought and belief. The pivotal role of concepts relates to their major functional role in intelligent systems. Concepts promote 'cognitive economy' (Rosch, 1978) - where suitable categorisation and partitioning of entities in the world helps in gaining and applying as much information about the world using the least cognitive effort. To understand the concept 'cat', for example, is to be aware of the likely behaviour and qualities of any particular cat. A further function of concepts is their use as the building-blocks of experience; situations rarely occur exactly the same way twice, but general patterns do occur. Smith (1989) gives the example of a child associating a stove with a burn, and being able to relate the experience of a burn from one stove to all stoves. A third use of concepts are in perception and induction. Using concepts, a growling dog can be recognised *as* a dog, along with the hazards that growling suggests. Concepts also act as templates or guides in recognising patterns in otherwise noisy data, and aiding to focus attention onto the significant. Popper (1972) asserts that '*all knowledge is theory-impregnated, including our observations*'. Additionally, domain-specific concepts may act as a mental and verbal shorthand between non-novices in the domain, and the use of jargon is widespread in any specialist field. Such concepts are an efficient means of linking discrete items of experience/knowledge (Thompson, 1959).

#### 4.3.2 Paradigms

The term 'paradigm' was first used by Kuhn (1970) to define a set of concepts, legitimate problems, acceptable methods, and prototypical examples of successful practice within a given scientific tradition or 'school'. Newtonian physics and behavioural psychology are examples of such paradigms (Kreutzer *et al.*, 1991). The term paradigm also refers to individual concepts that link an entire field (Shapiro, 1986). An example of one such paradigmatic concept in ecology is that

of Darwinian evolution being the prime regulator of organism morphology and behaviour.

Paradigms are useful in providing a coherent, directing framework that suggest what questions require addressing, what experiments to perform, and what literature requires publication and reading (Kreutzer *et al.*, 1991). Paradigms make the scientific process more efficient by directing data collection, which is typically haphazard in a pre-paradigmatic subject. By acting as a focus for study, a paradigm also throws up anomalies that act as targets for research. In many cases, these anomalies are eventually resolved, often creating new anomalies. There often arises inconsistencies between reality and the paradigm, requiring the paradigm to be altered to accommodate reality. Such inconsistencies may be referred to as 'refutations' of the current paradigm. This view of scientific progress is referred to by Popper (1972) as the development of science by 'conjecture' and 'refutation'. Occasionally, anomalies are not resolved satisfactorily, and as their numbers increase, the validity of the paradigm is threatened. Often, such paradigms are discarded and replaced by new ones. In this case a scientific revolution has taken place. An example of such revolutions include the replacement of an earth-centred view of astronomy with the Copernican view, in which the planets are understood to revolve around the sun (Shapiro, 1986). Kuhn (1970) views the development of science as a series of discontinuous episodes, marking the rise and fall of paradigms.

A paradigm may be regarded as an internalised view of a subject-area, and as such be used as a template through which concepts may be reasoned about and coherently 'understood' (Sowa, 1984). Such internalised models may be referred to as **mental models**: such a model is a reflection of a personal belief system, acquired through observation, instruction, or inference (Norman, 1983). The use of mental models is ably expressed in the words of Craik (1943):

If an organism carries a "small-scale model" of external reality and of its possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of past events in dealing with the present and the future, and in every way to react in a much fuller, safer, and more competent manner to the emergencies which face it.

#### 4.4 The Components of Reasoning in Habitat Evaluation within this Study

Ecology is a science that deals with living systems at a number of different, though integrated, levels. This includes the genetic, individual, population, community, and ecosystem levels of study (Krebs, 1985). Each level of integration

involves a separate and distinct series of attributes, problems, and methods, each of which may be treated as a 'paradigm'. For example, populations have an attribute called *density*, that is meaningless at the organism level. Likewise, a community has a *species diversity*, an attribute without meaning at the population level. In general, an ecologist dealing with a particular level will seek explanatory mechanisms at lower levels, and biological significance at higher levels (Krebs, 1985).

In reasoning about a habitat (and its evaluation), ecologists use a discrete paradigm that is a subfield of ecological science, concerned with the ecosystem level of activity (ie the biotic community and its abiotic environment). This involves consideration of community-level macrodescriptors, such as productivity and species diversity. Ecologists appear to internally assess habitats via their community parameters, and direct their experimentation, analysis, and reasoning according to this conceptual framework. Also considered are elements of the non-living world that have causal effects on, and are possibly causally affected by, the biotic community. Such elements include light, chemicals, and climatic features.

The case study upon which this research focuses concerns pond habitats. The research is based on the construction of expert systems that perform evaluation of ponds in mainland Britain in terms of suitability for an endangered species, the great crested newt (*Triturus cristatus*). The motivations for using this subject as a basis for the research are twofold. Firstly, the subject has specific and easily identifiable boundaries, providing clear constraints to this work within the time scale of funding. Secondly, the crested newt is a species protected by law, and some formal means of identifying habitats that are likely to support this species has some use in legal and conservational decision-making. Two separate systems are used in the present study, one first generation, and the other second generation. The former system (HEX) is shallow, whilst the latter (TRITON) uses deep, paradigm-based reasoning.

The focus of paradigm-based reasoning in this research will be community-level macrodescriptors. Those found to be of significance in assessing the suitability of pond sites to support the crested newt include productivity, structural diversity, and species diversity. Other ecological interactions that require consideration are population-level activity, such as predation and competition, when they are of relevance to the success of the crested newt. Finally, more general observations that relate to pond suitability will be included in the reasoning system. These include the non-biotic parts of the ecosystem already mentioned (such as substrate, temperature, light intensity), as well as 'observations' that provide vital information about the status of the pond (either directly or via inference) but do not reasonably fit into the categories of community macrodescriptor, abiotic component, or species interaction. Examples of these include roads near to the pond, unpleasant odours, and the presence of species that indicate specific oxygen levels in the pond.

## 4.5 Possible Approaches to the Representation of Paradigm-Based Knowledge

The representation of paradigm-based knowledge centres on the emulation of knowledge about a domain, rather than the heuristic knowledge held by an individual expert. This latter knowledge is the typical focus of a first generation expert system. That is not to say first generation expert systems contain no domain-level knowledge: it is likely that even the most simple expert systems contain implicit knowledge of their domain. Slatter (1987a) notes that an emphasis on emulating domain expertise (ie conceptual, or paradigm-based knowledge) is present in several medical knowledge based systems, such as INTERNIST (Pople, 1982), PSYCO (Fox, Barber and Bardhan, 1980), and NEOMYCIN (Clancey and Letsinger, 1981).

A paradigm-based system should be able, by representing knowledge about the interrelations of concepts, and applying suitable inference methods, to reason using domain concepts. It could be used, for example, to infer likely values of productivity given details concerning species diversity, biomass, and so on. However, for a system to perform tasks or solve problems in the real world, it must have knowledge of observable or measurable attributes of the real world, and the interaction of these 'observations' with the concepts of the domain. This type of knowledge is typically found in experts (Kreutzer *et al.*, 1991). An example of such knowledge is to understand that heavy shading of a pond will reduce the intensity of light entering the pond, whilst low light intensity causes low pond productivity. An ecological knowledge-based system that incorporates first-principle concepts, and that is to be realistically used, must also contain knowledge based within the experience of an expert.

### 4.5.1 The Representation of Knowledge in Second Generation Knowledge Based Systems

The representation of knowledge is a fundamental aspect in the building of expert systems. There are a number of ways of formally representing knowledge, the most widespread forms being logic, network representations (including inheritance networks), frames, production systems (Baur and Pigford, 1990), and blackboard systems (Nii and Aiello, 1979). Other possible representations commonly mentioned in the literature, but not yet commonly used in expert systems are scripts and object-oriented approaches (Parsaye *et al.*, 1988). However, no single representation is ideally suited to the various types of knowledge that may be used with an expert system (Minsky, 1991).

There are several ideal characteristics associated with the representation of knowledge, such as; it should be able to define and manipulate constituents in both their own terms, and in terms of other constituents in the domain; it should allow the 'nature' of an object to change given its context within a domain; it should have a high **homomorphism** (ie it should be an intuitively good match to reality); and it should carry out these criteria effectively and efficiently (Luger and Stubblefield, 1989). Other potential criteria include; an ability to explain its own reasoning process; availability to proper manipulation within a computing framework; expressiveness (have a lack of ambiguity, be clear, and be uniform); be easy to use; have relevance to the task being performed; and be **declarative** (ie be independent of its use within a system) (Bench-Capon, 1991).

Many of these criteria, along with other characteristics, have been identified as necessary for the representation of knowledge in second generation systems. **Declarativeness** is stressed as a key factor in the representation of knowledge in second generation expert systems (Buchanan and Smith, 1989; Lenat and Feigenbaum, 1991). This means representing knowledge as stand-alone facts, such as "rain is water", "water is wet", and "wet clothes may cause the onset of pneumonia", rather than associational statements that are highly specific, such as "If you go out in the rain, then it may cause the onset of pneumonia". Embodying such knowledge in a use-independent form means that the methods used to address and manipulate that knowledge will be explicitly stated, aiding in clearly understanding the reasoning process, and in explanation and justification of questions (Buchanan and Smith, 1989). Lenat *et al.* (1991) call the need to represent knowledge declaratively the "*Explicit Knowledge Principle*". Such explicitness will also aid in making the knowledge present in a system reusable for other tasks (Buchanan and Smith, 1989). Some workers advocate that such explicitness is a better criteria of depth in knowledge based systems than causality (Washbrook and Keravnou, 1990). Representing paradigmic knowledge about a pond habitat using declaratively-based knowledge should allow the system to perform several tasks, such as pond evaluation, management, and educational assignments. Also, the addition of experiential knowledge about different habitats to an existing deep knowledge base may allow the system the potential to evaluate these new habitats, whilst requiring less time and resources to build than a similar system constructed from scratch.

The more declarative the knowledge, the easier it is to change without causing unforeseen effects to the rest of the system. Ease of change is referred to by Keravnou *et al.* (1989) as **extensibility**, or **maintainability**. This has long been touted as a virtue of production systems ie sets of "if-then" rules. There is, however, strong evidence that unstructured rule bases do not facilitate change, as their ordering and sharing of terms hide implicit control of the knowledge base (ie the

order in which knowledge should be used). Change may have unforeseen and undesirable effects in such knowledge bases (Keravnou *et al.*, 1989).

Buchanan and Smith (1989) argue **simplicity** and **uniformity** are desirable aspects of a knowledge representation. This aids in knowledge acquisition, expert understanding of terms used, in testing and evaluation, and possibly in user understanding.

It must be noted that a second generation knowledge representation must accommodate the many other aspects associated with second generation systems. These include consideration of uncertainty, control, explanation of knowledge used, and the representation and availability for manipulation of appropriate knowledge (eg causal, conceptual, structural, etc).

Within the literature of 'second generation' expert systems, a variety of knowledge representations are used, many in combination. For example, CENTAUR, a system used to interpret measurements gained from lung function, uses a frame-like representation in which production rules are used as means of handling activities within the system (Aikins, 1983). This system contains knowledge common to all diseases relating to lung function, and explicit knowledge of how to run a consultation, and interpret evidence. The Oxford System of Medicine, a system designed to embody the working knowledge of a general practitioner, uses terms expressed in first order predicate logic, and similarly separates knowledge of how to direct the various tasks it performs from more specific medical knowledge (O'Neil *et al.*, 1989). MDX, a system for diagnosing liver complaints, uses productions (written in the AI language LISP), and attempts to explicitly represent both the conceptual knowledge of the domain, and the problem-solving strategy of a physician (Chandrasekaran and Mittal, 1983a). In the SBF research area, several projects utilise a logic-based approach, typically via first-order predicate logic. Examples of this include DART (Genesereth, 1984) and HOIST (Whitehead and Roach, 1990). The use of first order predicate logic is prevalent in many recent systems that integrate structural and causal knowledge in the medical domain (Bratko, Mozetic *et al.*, 1988; Hunter *et al.*, 1991; Mozetic, 1990; Shibahara *et al.*, 1983; Wildman, 1992). The popularity of first order predicate logic (typically implemented via the AI language PROLOG) in second generation systems generally may be due to its ability to embody, integrate, and expand upon, all of the existing knowledge representations currently in common use. The ability to use several knowledge representations in conjunction within a single system is seen as fundamental to the progress of knowledge based systems (Minsky, 1991). The suitability of PROLOG to represent second generation knowledge is more fully discussed in the next section.

#### 4.5.2 The Knowledge Representation used in PERSEUS

The development of a second generation system shell took place as part of the present research, and is called PERSEUS (Paradigm-based and Experiential Reasoning System using Ecological UnderStanding). The construction of PERSEUS required explicit recognition of the criteria of a knowledge representation scheme suitable for the issues addressed by PERSEUS (PERSEUS was ultimately used to build the second generation knowledge-based system, TRITON). These are detailed in the rest of this section.

The knowledge representation scheme must be a suitably close match to the way concepts are held about the macrodescriptors of an ecosystem, and the experiential knowledge of an expert. This requires the incorporation of causality, and empirical association. Additionally, depiction of inheritance relationships is required (eg 'ponds are kinds of ecosystem'). The ecological concepts required by TRITON, and therefore needing to be representable within PERSEUS, have been identified in the process of building the shallow system, during knowledge acquisition sessions between knowledge engineer and expert. These concepts (eg productivity, species diversity) have been constantly used in explanation by the expert, were explicitly identified as ecological concepts during knowledge acquisition, and were further validated by analysis of a set of 4 degree-level ecological textbooks, examined for ecosystem- and community-level concepts (Begon *et al.*, 1986; Colinvaux, 1986; Krebs, 1985; Pianka, 1988). Expressions of relationship between such concepts in these texts are typically discussed in terms of unspecified ecosystems, so that generalisations can be made such as "*high structural diversity indicates high species diversity, in any given ecosystem*". Knowledge acquisition sessions generated information relating these concepts at the pond level, such as "*low species diversity in a pond (in some circumstances) indicates low productivity*". Combining these two levels of information requires the PERSEUS system to be able to identify ponds as ecosystems, using such terms as "*ponds are kinds of aquatic ecosystem*", and "*aquatic ecosystems are kinds of ecosystem*". General statements about aquatic ecosystems, or more general ecosystems, can be treated as information about ponds, when information at the pond level is not present. In this way, more specific information is used in favour of more general, but more general information is available when the specific is lacking. It is a focus of this work that human beings utilise the most specific information they have available, but will use more general information when necessary (Smith and Median, 1981). For example, a car mechanic may be able to repair models not previously encountered, by drawing on more generalised knowledge about car repair.

As well as inheritance structures to promote specific-prior-to-general reasoning, there is a requirement for some means of (i) expressing relationships about entities of the pond ecosystem, and (ii) consolidating information about these entities. Production ('if-then' rule) systems are a widely-used method of expressing relationships between objects. An example of such a rule is "if fish are present, then crested newts are absent". Within this rule, there is no explanation of the interaction between fish and crested newts, and the nature of the interaction (ie causal, indicative, etc) is not expressed. The basic formalism is still useful, however, and an extended and adjusted version of a production is used within PERSEUS. This production takes the form of:

*< Object1 > having < Attribute1 > that is < Value1 > < Relation > < Object2 > having < Attribute2 > that is < Value2 > .*

Examples include:

*A pond having a light intensity that is low causes this pond to have a productivity that is low.*

*A pond having a location that is northerly indicates the pond has a light intensity that is low.*

In the form of an 'if-then' rule, the latter example could be expressed:

*If pond location is northerly, then light intensity is low.*

It should be noted that this is a reduced version of the full 'extended' production (called a 'relationship') that is used in PERSEUS, and the fuller version is discussed more properly in Chapters 5 and 6. The total complement of 'relations' (eg 'causes', 'indicates', etc) are described in Chapter 6, along with their association to control.

The object-attribute-value triplet present within these extended productions is used as the means to consolidate facts about specific entities modelled in PERSEUS/TRITON. The object-attribute-value triplet is commonly used in the 'frame' representation (Forsyth, 1989). The object is the entity being addressed in the system (eg "pond", "light"). Each of these objects may have a number of attributes. For example, the "pond" is an ecosystem, and will have attached all of the macrodescriptors that go with an ecosystem (productivity, species diversity, etc). Organisms, such as crested newts, will have similarly appropriate attributes, such as 'presence'. Any one of these attributes may have a number of values (eg pond productivity may be high, medium, low, etc). Using the frame-type approach, it is natural to treat an object within the system as a coherent entity, leaving it with a high homomorphism.



The knowledge representation used in the PERSEUS system involves a hybrid of three common knowledge representations; inheritance structures, productions, and frames. The use of multiple representations is by no means unusual in expert systems, and there is a growing recognition that the use of several representations is necessary in developing complex systems (Minsky, 1991). The AI language PROLOG has been selected as the means to implement this hybrid representation. It is a language based on first-order predicate logic, which may be readily used to implement and integrate other common formalisms, including frames, rules, and inheritance structures (Muetzelfeldt *et al.*, 1989). PROLOG also allows enhancements and extensions to such representations that may be used in other aspects of the working knowledge based system (eg explanation), without altering their intrinsic application. It is possible, for example, to include such relations as 'causes' within a traditional production rule without altering its use within the system (eg IF carnivorous fish are present in a pond, THEN this *causes* crested newt larvae to be predated). It was hoped that inclusion of more detail (such as 'causes') may help to make the machine-generated explanations given to users less terse, more natural, and ultimately more understandable. Additionally, PROLOG supports a declarative style of programming readily, and is suitable for accommodating such issues as control and uncertainty in a (relatively) straightforward way. Such features make PROLOG highly suited to be a tool in this research.

## CHAPTER FIVE

### THE REPRESENTATION AND HANDLING OF UNCERTAINTY

#### 5.1 Introduction: The Ubiquity of Uncertainty in Reasoning

Human reasoning often takes place in circumstances where information is incomplete, imprecise or otherwise vague. Reasoning that incorporates such uncertainty is essential in virtually all problem areas (Buchanan and Smith, 1989), and Clark (1990) states that '*uncertainty is present in most tasks that require intelligent behaviour*'. In empirical disciplines, including physics, ecology, medicine, and engineering, having *all* of the data, and being completely certain of the accuracy of that data, is a rare occurrence (Buchanan and Smith, 1989). Making decisions in the real world usually relies on proper handling and balancing of information which contains uncertainties that cannot be realistically eliminated.

The commonness of uncertainty in human reasoning indicates the need to handle uncertain information in intelligent systems. Whilst methods exist for handling uncertainty within expert systems, and some of these methods are widely used, they are in the main simplistic, and subject to an increasing lack of confidence (Bhatnagar and Kanal, 1986). Mamdani and Efstathiou (1985) assert that:

...the current performance of expert systems is being limited by their capacity to cope with uncertainty.

Sources of uncertainty are both numerous and diverse. As well as the possibility for incorrect, ambiguous, insufficient, or otherwise inadequate data, methods to properly interpret this data may also generate uncertainty, particularly within the heuristic approaches used in contemporary expert systems (Cohen, 1987b). Heuristic knowledge is more likely to fail in a case that is unusual or novel, for example, and common failure is likely to promote a lack of users' confidence in a system (see Section 1.1.1).

In considering ecology, uncertainty may arise from the types of relationship that exist within ecological systems (typically non-deterministic, and often ambiguous), and from the qualitative nature of contemporary ecology. In expert systems generally, user input may be a source of uncertainty, particularly where answers tend to be subjective.

Given the ubiquity of uncertainty in human reasoning, workers in fields that address decision-making have developed a number of methods to handle uncertainty. These are considered in the following sections.

## 5.2 Means of Addressing Uncertainty

The management of uncertainty in first generation expert systems has typically used methods and representations that lack either a sound theory or clear semantics (Bonissone, 1987). A variety of methods for handling uncertainty in AI systems exist, each with associated advantages, limitations, and problems. A common way to address uncertainty in expert systems is to ignore it, assuming that the uncertainty is small, therefore treating all knowledge as categorically true (Buchanan and Smith, 1989). Another related approach is to use only certain or almost certain knowledge, though this severely limits the range of applications that may be built (Davey and Stockwell, 1991). If this approach is appropriately used, however, the method is simple and efficient (Buchanan and Smith, 1989).

Methods that do seek to explicitly represent and handle uncertainty fall into two groupings; schemes where uncertainty is expressed as a numerical value; and schemes where uncertainty is expressed by qualitative means. The numerically-orientated approach is by far the more commonly used of the two, and the variety of methods coming under this definition are discussed in Section 5.4.1. Uncertainty handling using qualitative methods is discussed in Section 5.4.2.

Recent work in AI has begun to treat uncertainty in a fundamentally different way to expressing uncertainty using some distinct (quantitative or qualitative) calculus. Two interrelated methods of addressing uncertainty, as a control problem (Clark, 1990), and a distinct type of knowledge (Fox, 1986b), have emerged. Clark (1988) asserts that;

the best work on uncertainty relegates the calculus to a detail, and solves the problem of uncertainty by control.

Cohen (1987b) proposes that uncertainty may be managed by proper *ordering* of problem solving actions and sequences of actions, and gives the example of a person needing to buy a birthday present and a box to put it in. This size of the box is indeterminate until a present is bought. Once the present is bought, there is a definite lower limit to the size that the box must be. Cohen (1987b) goes on to suggest that this example illustrates that uncertainty is often due to the *timing* of evidence, and such uncertainty can be minimised by proper control. Cohen (1987a) asserts that ordering strategies are common in **blackboard** systems (see Section 4.5.1).

Saffioti (1987) notes that a characteristic of AI research is its concern with representing and using knowledge in the most explicit form possible; the recent trend in treating uncertainty as a control problem is an outcome of this. Fox (1986b) is a

strong advocate of treating uncertainty as a distinct type of knowledge. Fox (1986b) notes that current expert systems do not handle uncertainty as a distinct type of knowledge, but as a algorithmic mechanism that is tagged onto knowledge representations. This concurs with the view of Cohen (1985), in suggesting current expert systems that address uncertainty have two parallel streams of processing, one performing logical inference, whilst the other performs numerical certainty propagation/combination (see Section 5.4.1, and Figure 3).

### 5.3 Desiderata for Reasoning under Uncertainty

Bhatnagar and Kanal (1986) note that in contemplating the various approaches used to handle uncertainty, three different considerations arise. These are; the representation of uncertain information; the combination of bodies of uncertain information; and finally the drawing of inferences using uncertain information. Bonissone (1987) suggests a set of requirements that need to be satisfied by an ideal formalism for representing, and inferencing under, uncertainty, incorporating proper evidence combination. These are listed as points 1 to 5 in Table 4. Bonissone (1990) and Bhatnagar and Kanal (1986) both suggest further desiderata for an uncertainty calculus, expressed in Table 4 (points 6-8).

- 1) It should address; the representation of uncertain information; the combination of bodies of uncertain information; and finally the drawing of inferences using uncertain information.
- 2) There should be explicit recognition that combination rules should not be based on global assumptions of evidence independence.
- 3) Combination rules should not assume the exhaustiveness and exclusiveness of the hypothesis.
- 4) There should be explicit representation of the amount of evidence for supporting and for refuting any given hypothesis, and there should be an explicit representation of the reasons for supporting and for refuting any given hypothesis.
- 5) The representation must be natural to parties involved in building and using the system.
- 6) The uncertainty calculus must be modular in order to support dynamic changes to the boundaries of the knowledge base.
- 7) The uncertainty calculus should be able to represent defaults that are compatible with the rest of the knowledge base where information may be missing, but be able to retract conclusions based on defaults or uncertain evidence when new evidence comes to light.
- 8) The uncertainty calculus should provide uncertain reasoning in real time.

Table 4: Desiderata for an Ideal Uncertainty Calculus

(after Bhatnagar and Kanal, 1986; Bonissone, 1987; Bonissone, 1990)

#### 5.4 Methods for Reasoning under Uncertainty

Current approaches to uncertainty assume all uncertainties may be treated in the same way, and this treatment is problem-independent (Mamdani and Efstathiou, 1985). The approaches to uncertainty can be divided into the expression of uncertainty in quantitative terms, or in qualitative terms.

##### 5.4.1 Numerical Methods

Cohen (1985) refers to quantitative methods as **parallel certainty inferences**. He notes that such approaches divide reasoning under uncertainty into two parallel

streams; the first is a stream of inferences about the domain; and the second is a stream of calculations of the credibilities of these inferences. Figure 3 illustrates the two parallel streams of reasoning. Cohen (1987b) notes that types of parallel certainty inferences include Bayesian methods, certainty factor methods, Dempster-Shafer calculi, and fuzzy logic (each of these will each be discussed in this section).

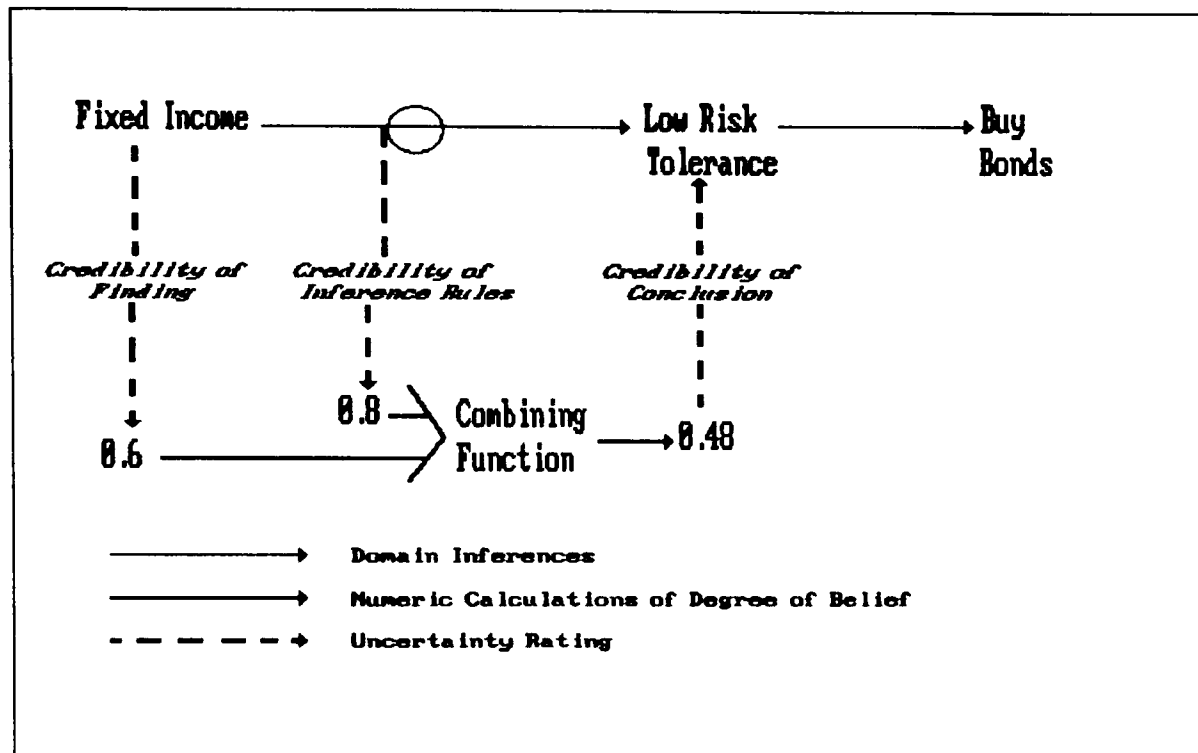


Figure 3: An Example of a Parallel Certainty Inference (after Cohen, 1987b)

### Probabilistic Methods

A common approach to addressing uncertainty in expert systems is by using **conditional probabilities**, or **Bayesian methods**. The conditional probability of  $x$  given  $y$  is simply the probability that  $x$  occurs given  $y$  has occurred (Jackson, 1990). For example, the probability that a pond is unsuitable for crested newts given that fish are present.

In Bayesian approaches, the conditional probability of  $x$  given  $y$ , expressed  $p(x | y)$ , is computed using the following formula:

$$(1) \quad p(x | y) = \frac{p(x \text{ and } y)}{p(y)} \quad (5.4.1-1)$$

Additionally, the joint probability of  $x$  and  $y$  is equal to the probability of  $y$  given  $x$  multiplied by the probability of  $x$ ;

$$(2) \quad p(x \text{ and } y) = p(y | x)p(x) \quad (5.4.1-2)$$

Using equations (1) and (2), it is possible to derive the simplest expression of **Bayes' Rule**:

$$(3) \quad p(x | y) = \frac{p(y | x)p(x)}{p(y)} \quad (5.4.1-3)$$

The probability of  $x$ ,  $p(x)$ , is referred to as the *prior probability* of  $x$ ; that is, prior to the discovery of  $y$ . The probability of  $x$  given  $y$  is referred to as the *posterior probability*. Using these equations, in particular (3), it is possible to rationally combine evidence to give probabilistic certainties. For example, the probability of crested newts occurring in a particular pond in the UK ( $p(\text{newt})$ ), given no further information, is about 0.11 (11%). The probability of fish being present in a pond ( $p(\text{fish})$ ) is about 0.2, whilst the occurrence of fish and crested newts in the same pond ( $p(\text{newt and fish})$ ) is (say) 0.0005. Using equation (1), we can derive the probability of crested newts in a pond, given fish presence (Note these figures are uncorroborated by real data, and are used only as an example):

$$\begin{aligned} p(\text{newts} | \text{fish}) &= \frac{p(\text{newts and fish})}{p(\text{fish})} \\ &= \frac{0.0005}{0.2} = 0.0025. \end{aligned}$$

The prior probability that newts are present in a pond is 0.11, but when further evidence is gathered, that fish are present, the posterior probability that newts are present becomes 0.0025.

Examples of successful expert systems that have used Bayesian methods and other probabilistic approaches include PROSPECTOR (Gashnig, 1982), used in mineral exploration, and a variety of medical expert systems concerned with diagnosis and decision support for medical practitioners (Spiegelhalter, 1987).

A range of Bayesian approaches to handling uncertainty exist, all of which belong to the school of probability theory. At the most specific and restricted level, Bayesian methods are used that rely on global assumptions of evidence independence (Bonissone, 1987). It is assumed that the components within such a system are independent of each other. This is the most common approach to using Bayesian methods (typically referred to as 'independent Bayes'), and there is an unfortunate trend for writers in AI literature to refer to 'independent' Bayesian methods as 'Bayesian methods' (eg Jackson, 1990); the former is in fact a subset of the latter. Other methods exist that do not rely on, or specifically address, assumptions of global independence (Lauritzen and Spiegelhalter, 1988; Pearl, 1986).

Whilst widely used, and a common way to address uncertainty in a variety of

expert system shells, such as LEONARDO (Creative Logic, 1987), probabilistic methods of handling uncertainty have been widely criticised in the literature.

Table 5 presents a listing of the various criticisms of Bayesian approaches that occur in the literature. Many of the criticisms listed in Table 5 have been comprehensively discussed in the literature. The first 6 objections have been addressed in a set of papers that have provided eloquent opposition (Cheeseman, 1985; Pearl, 1985; Spiegelhalter, 1986). Criticisms 7-9 are rejected in Henrion (1987). In this thesis, only arguments applying to the more pressing or relevant criticisms (in relation to the present research) are addressed.

A criticism often made of probabilistic methods is the use of human judgements as a means to generate numeric 'probabilities' undermines the rationale of the probabilistic approach. This claim results from a belief that probability is about long-run relative frequencies of events, and requires some function that assigns probabilities to every element within a system (Jackson, 1990). Such views are referred to as the *frequentist* view of probability (Olson, Willers *et al.*, 1990). A second school of statistical thought, the *personalist*, or *subjectivist*, holds that probabilities are not properties of the real world, but are subjective; a hallmark of this school is the willingness to accept human estimates of probability. This approach is championed by a number of workers, including Lindley (1987) and Spiegelhalter (1986). Tonn, Goeltz and Travis (1992) document some of the problems of generating probabilities from human experts.

The need for inordinately large amounts of data/judgements to generate a realistic system is another frequent criticism of probabilistic methods. Henrion (1987) answers this criticism by asserting that inordinate numbers of probabilistic judgements are not necessary where the inference structure created is a reasonable reflection of the way the expert thinks about the domain (ie the links present between components in the system mimic the mental model of the expert, and typically such links are well-organised, and relatively few).

Despite criticisms, probabilistic methods have been successful (eg Gashnig, 1982), and are commonly discussed as a prime means of addressing uncertainty in expert systems (eg Jackson, 1990). Extensions to 'traditional' Bayesian approaches exist (eg Pearl, 1986), and Spiegelhalter (1986) reviews such systems.

Henrion (1987) notes that advantages of probabilistic methods include axiomatic simplicity (eg it is easily imposed on existing knowledge representations, such as production rules), and it is a well-defined approach that provides formal means to incorporate empirical data. Lindley (1987) adds that other numerical methods, such as fuzzy logic and Dempster-Shafer approaches (discussed later in this section), are considerably more complicated concepts than probability, and goes on to claim that probability is a better way to handle uncertainty than existing



alternatives, despite its shortcomings. Lindley particularly focuses on the 'coherence' of probabilistic approaches relative to other methods of handling uncertainty.

Overall, probabilistic methods have proved to be a useful method of combining values of uncertainty, rather than a rigorous approach.

1. Probability requires vast amounts of *precise* data or unreasonable numbers of expert judgements.
2. It is unable to express ignorance, vagueness or certain other types of uncertainty.
3. It does not distinguish reasons for or against, or identify, sources of uncertainty.
4. The inference process is hard to explain.
5. It cannot express linguistic imprecision.
6. The independence assumptions used in 'independent Bayes' are unrealistic - it is very difficult to be certain that independence exists between different parts of a system. It may be obvious that a tyre puncture and a failed headlight in a car are independent entities, but can similar assumptions be made about more complex systems, such as the non-interaction of any two species in an ecosystem?
7. It is computationally intractable.
8. It is not how humans reason.
9. It does not make much difference what method you use, so the 'objectivity' of probabilistic methods is not essential.
10. The use of expert judgements, where it occurs, undermines the 'objectivity' of probabilistic methods, and doubts about consistency and comprehensiveness must occur.
11. It is difficult to modify a Bayesian-based set of values because of the dependencies between them. For example, the sum of probabilities for all possible hypotheses which display some evidence E must sum to 1.0. If we add a new hypothesis (eg identify a new disease with a symptom shared by many other diseases), the values used in the system may require recomputation.
12. A probability value reveals no information about its precision.
13. As probabilities are estimated from data frequencies, how are events with low frequency observations to be addressed?
14. The interpretation of the numeric value is unclear to users, as is the relative contribution of each piece of information given to the system.

Table 5: Objections to Probabilistic Methods of Handling Uncertainty

(after Clark, 1990; Frost, 1986; Henrion 1987)

## Certainty Factors

A second common approach to handling uncertainty quantitatively is by the use of *Certainty Factors*. This approach is also referred to in the literature as the use of *confidence factors* (Baur and Pigford, 1990), *Certainty Theory* (Shortliffe and Buchanan, 1975), *neo-calculist* methods (Olson, Wagner *et al.*, 1990), and as an extension of *Confirmation Theory* (Bonissone and Tong, 1985). Certainty factors (CF's) are a measure of the degree of belief in a fact or rule (Parsaye and Chignell, 1988).

The precise manner in which certainty factors should be used to calculate uncertainty is a subject of debate (Buchanan and Shortliffe, 1984), but a standard approach is common. The certainty factor is assigned to a fact or rule, usually in a predetermined range. This may be -1 to 1, 0 to 1, -100 to 100, or some similar range. Assuming for this case it is -1 to 1, then absolute truth is equivalent to 1, no valid inference is 0, whilst absolute falsehood is -1. The certainty factor of *A* is written  $CF(A)$ . Reasoning using certainty factors requires some means to find  $CF(A \text{ and } B)$  and  $CF(A \text{ or } B)$  in terms of  $CF(A)$  and  $CF(B)$ . A common way to do this is to adopt the weakest certainty factor in groups of 'AND' statements, and the strongest certainty factor in groups of 'OR' statements. Formally, this is expressed:

$$CF(A \text{ and } B) = \text{minimum}[CF(A), CF(B)]$$

$$CF(A \text{ or } B) = \text{maximum}[CF(A), CF(B)]$$

Typically, rules and/or facts are generated, with the certainty factor typically tagged at the end of the representation of the fact/rule. An example rule is;

'If (1) the stain of the organism is gram positive and  
(2) the morphology of the organism is coccus and  
(3) the growth conformation of the organism is chains  
then there is suggestive evidence (0.7) that the identity of the  
organism is streptococcus.' (Shortliffe and Buchanan, 1975)

Dan and Dudeck (1992) contains a full treatment of evidence combination in certainty theory.

Certainty factors were originally developed to address problems that occur in using Bayesian methods within a modular, rule-based approach (Shortliffe and Buchanan, 1975). Shortliffe and Buchanan (1975) were concerned with both the assumptions required to be made when using Bayesian methods, and in the cognitive complexity that follows from dealing with large numbers of conditional and prior probabilities. Additionally, they found that it was difficult to consistently assess subjective probabilities elicited from experts, and that the numbers given were different in character from probabilities (Heckerman and Shortliffe, 1992). The

development of certainty factors was part of work on the highly-influential MYCIN system, built to diagnose, and prescribe against, blood infections (Shortliffe, 1976). The use of certainty factors is widely discussed as a common and widespread means to address uncertainty within the literature (eg Jackson, 1990).

Several objections have been raised to the use of certainty factors. The original developers freely describe CF approaches as an approximation of Bayesian methods (Shortliffe and Buchanan, 1975), and as such, it may inherit many of the problems of Bayesian methods, other than those certainty factors were explicitly developed to answer. Spiegelhalter (1986) describes the certainty factor approach as an underspecified model; given a certainty factor for event *A* when event *B* has occurred, it has no means to address the 'certainty' of event *A* given the non-occurrence of (*B*). Other specific problems are associated with this formalism, such as it has no means for combining non-independent evidence other than chunking it into the same rule, and such a method is unsatisfactory for large bodies of knowledge. Clark (1990) notes that the certainty factors used in MYCIN occasionally address utility considerations; for example, higher certainty factors were assigned to rules with more serious consequences, so that such rules were favoured before others with less serious potential consequences. Clark (1990) also notes that, whilst such misuse may occur, it is not fundamental to the nature of the calculus, and the responsibility for this must lay with the builders.

Horvitz and Heckerman (1986) have criticised the *ad hoc* nature of this uncertainty calculus, noting that certainty factors are used as measures of change in belief, when they were actually elicited from experts as degrees of absolute belief. Heckerman and Shortliffe (1992), assert that '*the AI community has largely abandoned the use of CF's*' - perhaps due to associated problems.

Despite such criticisms and problems, the certainty factor approach has several recognised benefits. The appeal of the certainty factor formalism is that it provides a method that allows uncertainties to be quantified and combined in a formal (typically rule-based) calculus. Frost (1986) notes that using certainty values such as -1 to 1 lead users to clearly recognise the meaning of certain points in the scale, and that the combination of contradictory rules of the same certainty acts to simply cancel their effects out, making the approach intuitively sensible and straightforward. Olson, Wagner *et al.* (1990) note that the simple tagging of rules with certainty factors is both computationally tractable and appealing, and that a further reason for the appeal of certainty factors is its close association with a ubiquitous means to represent knowledge, the production rule. The modularity of the rule-certainty factor unit (able to be considered separate from other rules, and still make sense) contributes to its tractability as a computational method, and general appeal (Jackson, 1990).

## Dempster-Shafer Theory

A third approach to handling uncertainty quantitatively is the *Dempster-Shafer* method. The Dempster-Shafer theory of evidence was developed by Dempster (1967) and subsequently extended by Shafer (1976). It is also referred to as *Evidence Theory* (Shafer, 1976) and *Belief Theory* (Bonissone, 1987). Saffioti (1987) portrays Dempster-Shafer theory as a recent attempt to mathematically formalise an uncertainty calculus that addresses some of the adequacy problems of Bayesian and certainty factor approaches. In Dempster-Shafer theory, 'Belief functions' are used in place of probabilities or certainty factors, and these are used to put bounds on the assignment of probabilities to events, instead of having to specify the probabilities exactly. Methods for combining evidence and generating belief functions for these are also available within this theory.

Dempster-Shafer methods make an explicit distinction between uncertainty and ignorance (Frost, 1986). In probabilistic approaches, the sum of the set of hypotheses concerning some state must equal 1. Therefore a belief in some hypothesis,  $H$ , implies that one's remaining belief is committed to its negation, that is;

$$P(H) = 1 - P(\text{not}(H)) \quad (5.4.1-4)$$

or

$$P(H) + P(\text{not}(H)) = 1$$

For example, if one believes that the likelihood of crested newts being present in a pond is 0.11, then the implicit assumption is made that one believes the likelihood of newt absence is 0.89.

With Evidence Theory, a commitment of belief to some hypothesis ( $A$ ), does not force a commitment to the remaining belief in its negation (NB The belief function in hypothesis  $A$  is expressed  $Bel(A)$ ). This is expressed;

$$Bel(A) + Bel(\text{not}(A)) \leq 1 \quad (5.4.1-5)$$

or

$$Bel(A) + Bel(\text{not}(A)) + X = 1$$

where  $X$  is the measure of ignorance.

Like Bayesian and CF approaches, Dempster-Shafer theory relies on degrees of belief to represent uncertainty; unlike the two former methods, it allows attachment of beliefs to sets of hypotheses *explicitly* (eg {gastric cancer, gastric ulcer}) rather than hypotheses in isolation (Clark, 1990). Note that it is possible to tag uncertainties to 'sets' of hypotheses using the previous approaches, eg generate a  $P(\text{gastric cancer or gastric ulcer})$ , but such sets are considered as a single hypothesis

within the calculus, rather than being explicitly identified and treated as a set.

It is argued that the ability to address sets explicitly allows systems using Dempster-Shafer approaches to narrow down sets of hypotheses with the accumulation of evidence, rather than single hypotheses, a process more akin to diagnostic reasoning than Bayesian methods (Gordon and Shortliffe, 1984). It is also possible to construct different sets of hypotheses that share members. For example, an ecological set of pond predators may be {perch, crested newt, dragonfly larvae, great diving beetle larvae}, whilst an alternate set of pond insects could be {dragonfly larvae, great diving beetle larvae, mosquito larvae}. These sets may be used to reason about different aspects of ecological interest, such as predator-prey interaction, or insect biology. These sets (called *frames of discernment*) may additionally be combined to supersets, or fragmented to subsets, to create more 'frames of discernment'.

Several criticisms of Dempster-Shafer theory have been made within the literature. Bhatnagar *et al.* (1986) notes that the theory seems to lack effective decision making procedures, and that the presence of both ignorance and belief within the system means that both must be carried through any process of combination or inferencing. Bonissone (1987) notes that the theory suffers from '*computational complexity*' arising from various intricacies within the theory, such as the possible need to address all subsets and supersets of a given hypothesis set, and the combinatorial increase in interactions requiring computer processing that occur with the use of large sets of hypotheses.

Other problems occur in the 'normalisation' procedures used within Dempster-Shafer theory (Zadeh, 1984). Using these procedures, the most likely outcome is assigned the strongest value. However, this does not inform the user of whether the most likely event is a powerful possibility, or merely the best of an unlikely bunch.

Despite shortcomings, Dempster-Shafer theory offers several advantages over other numeric approaches to uncertainty. It is able to represent ignorance, and to deal with non-exhaustiveness by using a catch-all 'any other hypothesis' set. In dealing with sets as a unit of uncertainty, invalid assumptions of probabilistic equality between possible members are not made when no information is known, as may occur with Bayesian methods (Jackson, 1990).

Overall, Dempster-Shafer theory is a relatively novel approach to addressing and handling uncertainty. Saffioti (1987) considers that Dempster-Shafer and Bayesian methods are not alternatives, but belong to a family of probabilistic theories for handling uncertainty, and their use should be determined by their suitability for the application at hand. Bayesian methods are more computationally tractable, and typically more convenient. Dempster-Shafer theory, on the other hand, addresses ignorance explicitly.

## Fuzzy Logic

Whilst Bayesian methods, certainty theory and Dempster-Shafer theory may be considered a family of similar approaches to uncertainty (Saffioti, 1987), *Fuzzy Logic* offers a significantly different means to handle uncertainty by addressing one of the omissions of probabilistic methods; lexical imprecision, or elasticity (Zadeh, 1978). In classic logics, any entity can be described as either a member or non-member of a set. In fuzzy logic, an object may be a member of a set '*to some degree*' (Jones, 1989). This approach is associated with lexically imprecise terms such as 'tall', 'slow', or 'good looking'. Such terms are in very common use in language and thought. The calculus of fuzzy logic involves the mapping of qualitative terms into sets of values, typically vectors, and reasoning involves manipulation of these numbers. Despite its emphasis on qualitative values, fuzzy logic remains an essentially numeric technique. However, fuzzy sets offer a means to address vagueness as numerical values within a rational, coherent framework of expression, evidence combination, and inferencing (Jones, 1989).

Workers that have documented this approach identify several types of lexically imprecise terms. Zadeh (1986) describes linguistic concepts that have such imprecision, such as 'short' and 'busy', as *fuzzy predicates*. *Fuzzy hedges* are words that can be added to fuzzy predicates (and other fuzzy terms) to extend their utility; examples include 'very', 'extremely', 'fairly' (Zadeh, 1983a). *Fuzzy quantifiers*, such as 'occasionally', 'sometimes', 'very often', can be placed within sentences as indefinite measures of uncertainty (Jones, 1989). An example of a sentence that contains all three types of fuzzy terms is '*In some places, crested newts are extremely abundant*'. Zadeh (1986) identifies other types of fuzzy terms including *fuzzy probabilities* (eg likely), *fuzzy possibilities* (eg quite possibly), and *fuzzy truth values* (eg almost true, mostly untrue).

There are several criticisms of fuzzy logic. Baur and Pigford (1990) assert that the representation of ambiguous terms is very complex and difficult. The transition of qualitative terms to quantitative values is likely to be fraught with difficulties, particularly in maintaining objectivity and standard methods. Bonissone (1987) argues that fuzzy approaches do not offer any improvement on probability theory. Bhatnagar *et al.* (1986) note that, whilst fuzzy approaches may be very useful for formalising such terms as 'tall', it is not useful for representing the approximate height of a person. Thus, the theory seems more appropriate for representation of loosely defined concepts rather than uncertain information *per se*.

Despite these criticisms, Bonissone and Tong (1985) note that the use of fuzzy quantifiers is a more natural way to express degrees of implication than scalar (eg certainty factor) methods. Zadeh (1983b) claims that the imprecise language that

characterises much expert knowledge is a justification for the use of fuzzy reasoning. Zadeh (1986) further argues that it is *this* kind of complex uncertainty that probability theory and its variants are unable to address, and goes on to say that the mapping of realistic numeric values of probability to lexically imprecise *and* lexically precise expressions is an undefined process. However, it should be noted that the same argument can be made about assigning numeric values to fuzzy sets.

Recent reports in the literature (Cox, 1992) have discussed the successful use of fuzzy processors in Japanese technology, used in the guidance of trains, elevators and cars. Jones (1989) best summarises the potential of fuzzy logic:

Fuzzy set theory offers a tool which recognises the imprecision with which we label and discuss many concepts, and thus offers a way of dealing with linguistically defined objects. It thus offers a way of developing rules which have the expressive power to represent vagueness in a very natural way.

### **Other Numeric Approaches to Uncertainty**

Whilst the four numerical approaches to uncertainty discussed above are commonly debated in the literature, a variety of other approaches to handling uncertainty using (pseudo)numeric means have been suggested by workers.

*Evidential Reasoning* (Garvey, Lowrance and Fischler, 1981) may be considered a variant on Dempster-Shafer theory, emphasising proper management of uncertain knowledge from different sources (Bonissone, 1987). In this approach, the likelihood of a proposition is specified at its lower limit by the attached degree of belief, and from above by its degree of plausibility (Bonissone and Tong, 1985). However, it is difficult to see what benefits this approach may have over traditional Dempster-Shafer methods.

*Evidence Space* is an unusual approach, using two-dimensional space as the measure of uncertainty (Rollinger, 1983). The Cartesian (x,y) coordinates correspond to evidence for and evidence against a proposition. The four extremes of the 'uncertainty space' bounded are (0,0) for complete ignorance, (1,0) for absolute certainty in the proposition, (0,1) for absolute refutation of the proposition, and (1,1) for maximal conflicting evidence. However, this method is limited because Rollinger (1983) does not suggest a means of combining evidence, propagating uncertainty through chains of inference, or ways to perform other necessary operations.

Bundy (1984) proposes the use of *incidence calculus* as a means to overcome some of the problems of Bayesian (and other numerical) approaches, using methods established within probability theory. The fundamental unit of reasoning under this calculus is an 'incidence', ie a situation or interpretation that may occur. Whilst the separate probabilities of two statements does not determine the probability of conjunction or disjunction, Bundy (1984) suggests it is possible to get an approximate value for the probability of conjunction. This conjunction is termed an 'incidence' (Sheridan, 1991). Bundy (1984) argues that other numeric techniques are insufficient, as they do not truly capture the properties of probabilistic reasoning (ie they are not 'objective'). However, the scant details given about this approach appear to be computationally complex. It must also be noted that this approach was not fully developed at its introduction, and little refinement has occurred since, even by its originator (Bundy, 1984).

#### 5.4.2 Non-numerical Methods

Whilst numerical approaches to uncertainty are commonly used, many workers argue against such methods as an exclusive way to address uncertainty. Fox (1986a) asserts that the handling of uncertainty is of prime importance in realistic decision-making computer systems, and that the statistical concept of uncertainty may be considered as one of many possible approaches. Non-numeric methods for addressing uncertainty are discussed in the following section.

#### **Nonmonotonic and default logics**

*Logic* has been a common tool of knowledge representation throughout the history of AI (Saffioti, 1987). However, classical logic lacks facilities for describing how to revise a formal theory when new information causes inconsistencies to arise. In classical logic, once a proposition is taken as true, it cannot be revoked, despite further information that may be inconsistent with the rest of the knowledge base. This property, where addition of axioms will always tend to increase or maintain the total number of propositions considered to be true, is referred to as **monotonicity** (Clark, 1990). Overcoming such limitations has typically involved extending or modifying classical logic (Saffioti, 1987). This has led to the development of a variety of innovative logics, including **nonmonotonic** and **default** logics. Such logics are able to use statements of the form '*in the absence of contrary information, X is true*'. It is argued that much human knowledge tends to be held in this form,



eg '*typically, birds fly*'. It is a reasonable assumption that if *X* is a bird, it is likely to be able to fly, unless further information comes to light eg *X* turns out to be an ostrich. One notable framework for using nonmonotonic reasoning has been proposed by Doyle (1979), and is the **Truth Maintenance System (TMS)**. The TMS has nodes representing propositions, and connected to these propositions are justifications. Such justifications may consist of further propositions or justifications. If new information causes inconsistencies within the framework, the TMS is able to revise itself to restore consistency, and informs the reasoner about these revisions. Default values are typically overridden by any actual values given by a user, or values inferred from user-generated information (Bhatnagar *et al.*, 1986). Using such a strategy makes the TMS, and nonmonotonic/default logics generally, useful for reasoning with incomplete information.

There are several problems inherent in the nonmonotonic/default logic approach. For example, inconsistencies may occur when two default rules are contradictory. An example may be:

*Assume fish are present in a pond when one or more anglers are present.*

*Assume fish are absent if the pond has dried up completely, but subsequently refilled, within the last 6 months.*

Consider a situation where a status for fish is unknown, an angler is present, and the pond has dried up and subsequently refilled in the last six months. There is no formal non-arbitrary means to deal with this, and higher level reasoning is required (Clark, 1990).

Bonissone (1987) notes other limitations of nonmonotonic logics. Propositions are considered to be true or false, and no degrees of credibility/belief are permitted, so that nonmonotonic approaches are unable to use partial information in the reasoning process (ie information that is equivocal eg *the cat may have been grey*). This criticism has been rejoined by the argument that nonmonotonic arguments were not developed to deal with partial information, but instead to deal with incomplete information (eg knowing or not knowing for definite the colour of the cat) (Clark, 1990). Rich (1983) adds further criticism by noting that an ability to handle default information, touted as a virtue of nonmonotonic logics, is possible using numerical methods.

**Modal** logics are another means of handling uncertainty (Saffioti, 1987). Modal logics extend classical logic by distinguishing between what is necessarily true and what is possibly true. Necessary truth admits no degree of certainty, it is either true or false, whilst possible truth is not absolute. Mamdani *et al.* (1985) suggest that there is a '*kind of uncertainty*' in the use of necessary and possible truth. Such an approach, however, seems less sophisticated and of less use in handling uncertainty than nonmonotonic/default logics. Variants on modal logics exist, where

the explicit representation of necessity and possibility is replaced by necessity and 'belief' in epistemic logic, 'ought to be true' and 'permitted to be true' in deontic logic, addition of language tense (eg past, present, future) in tense logic, and so on. Multi-valued logic allows a variety of values to be attached to a proposition other than 'true' and 'false' (Saffioti, 1987). Such logics are reviewed in Mamdani *et al.* (1985).

Fuzzy logic is also cited as a type of extended logic that addresses uncertainty within the literature (Mamdani *et al.*, 1985). However, reasoning under fuzzy logic is fundamentally number-based, and it therefore cannot be properly considered as a non-numeric means of addressing uncertainty.

Overall, reasoning using logic offers a coherent way to reason about the real world in a fundamentally different way from numeric methods. Whilst the logics mentioned here may not be able to handle partial information, it may be argued that such approaches are a closer model of human reasoning under uncertainty, in terms of its use of defaults and retractability of propositions, than numeric methods.

### Cohen's Theory of Endorsements

A second approach to handling uncertainty qualitatively is the *Theory of Endorsements*, developed by Cohen (1985), and is based on linguistic uncertainty. The motivation behind Cohen's work is the observation that states of uncertainty are composites of reasons to believe and disbelieve, and that strength of evidence is a summary of these factors. Cohen is more concerned with developing '*a plausible model of human reasoning about uncertain situations*' than developing a method that is efficient and computationally tractable (Cohen, 1985). This method uses *endorsements*, or information for and against a proposition. Cohen (1985) uses a ledger book metaphor to describe the laying-out of endorsements for and against (*pro* and *con*) a proposition. The pros and cons are not treated with equal weighting; the weighting of each piece of evidence is determined by its credibility. Cohen describes the ranking of endorsements as the '*weighing of evidence*'; this is an unfortunate and perhaps remiss choice of words, and the use of terms such as 'weights' of evidence has been present (referring to numeric values) in probability theory for at least 4 decades previously (eg Good, 1950). Cohen's 'weights' are strictly qualitative.

If inconsistencies occur, the total evidence "weight" is assessed to see if there is more evidence for or against a proposition; the certainty of the proposition is typically represented as its strongest endorsement (Cohen, 1985). Cohen and Grinberg (1983) add a further point of interest; certainty must be considered with

respect to the task involved. Cohen *et al.* (1983) assert that '*the idea of complete certainty is an artifact of numerical representations of degree of belief*', and relative certainty in one goal may be uncertainty in another. For example, one may be certain enough of one's income to buy a car, but not certain enough to buy a house (Cohen *et al.*, 1983).

As well as recording the evidence for/against a proposition, endorsements identify the activity required to resolve the uncertainty inherent within that evidence. For example, an endorsement that is an "over-generalisation" may prove to be false in certain cases (Cohen, 1987b). An example of such a rule may be:

*if fish are present  
then newts are absent*

A system that recognises such a rule is an over-generalisation (ie there will be times when the conclusion is false when the premise is true) will not automatically accept newt absence when fish are present. It looks for corroborating evidence (ie a rule with the same conclusion, but a different premise) prior to acceptance.

Bonissone and Tong (1985) note that endorsements provide a good mechanism for providing explanation, as they maintain the entire history of justifications (evidence for and against a proposition), as well as retaining the relevance of any proposition with respect to the system's goal. However, means of evidence combination, propagation, and ranking of endorsements are not universal, and must be explicitly specified for each particular context.

The theory of endorsements has been heavily criticised for its lack of clear methods in addressing propagating, combining and ranking endorsements, and in balancing set of pro and con endorsements of a proposition (Saffioti, 1987). Cohen (1985) anticipated the latter argument (rather weakly) by suggesting that people rarely need to balance one piece of evidence against another, as a coherent world-view will rarely provide such conflicts.

Whilst poorly developed as an uncertainty calculus, Cohen's approach is of some interest. It offers a purely qualitative means to handle uncertainty, is more concerned with modelling human reasoning than computational efficiency, and it can explain its line of reasoning coherently. Saffioti (1987) supports Cohen's approach by noting that the lack of formal methods in the theory of endorsements may be seen as a sign of our lack of understanding of uncertainty as a type of knowledge, rather than a flaw in Cohen's approach. Clark (1990) notes that Cohen (1985) does not preclude the use of numerical measures of belief as endorsements, and this points the way to developing a possible means of combining the various approaches into a single coherent framework.

## Reasoned Assumptions

A third approach to the qualitative handling of uncertainty is the *Reasoned Assumption* approach, proposed by Doyle (1983). Here, the uncertainty inherent within a statement is removed by explicitly listing all the possible exceptions to the statement. Exceptions would be acquired from an expert, who would be encouraged to think about the specific instances where a given rule is invalidated (Doyle, 1983). When gathering a list of exceptions is not feasible, uncertainty is reduced by making **assumptions** about the typicality of a value (ie default value) and the defeasibility (ie liability to defeat) of a statement. When an assumption derived from deductive processes is found to be false, nonmonotonic mechanisms are used to maintain the integrity of the statement list.

Bonissone and Tong (1985) note that such methods are suited to handling incomplete, but not imprecise, information (in much the same way as nonmonotonic logics), and Doyle (1983) acknowledges that assumption-based systems lack means to address a statement's measure of belief. However, Doyle (1983) asserts that the reasoned assumption approach provides a coherent means to explain and justify the reasoning process in a way superior to numeric methods, and that assumption-based systems are more modular than numeric systems. Modularity occurs because statements can be altered without consideration by the system builder of how the action of the working system may be affected, as assumption-based systems perform their own "bookkeeping". Doyle (1983) suggests his approach betters numerical methods by offering '*closer correspondence with the expertise easily obtainable in practice*', and in methodological simplicity.

## Fox's Semantic System

A fourth approach to handling uncertainty qualitatively can be found in the *Semantic System* of Fox (1986a,b), and is concerned with formally identifying and defining states of uncertainty within the semantics of natural language. Fox (1986b) identifies a number of logical types of uncertainty, each of which may be used to express uncertainty in alternative (but not necessarily contrasting) ways. Fox (1986b) gives examples of the *possibility*, *plausibility*, and *probability* of a statement. *P* is possible if no conditions exist that indicate it is impossible, is plausible if the balance of arguments exist that favour *P*, and is probable if one piece of evidence exists for *P* (Fox, 1986b). Fox argues that using such qualitative distinctions, it is possible to reason with uncertain facts and rules in a coherent manner (eg 'if fish are present, then it is *probable* that newts are absent', 'given no information, it is *possible* that

news are present').

This work is supplemented by the development of generic reasoning procedures to address decision making (Fox *et al.*, 1988). This involves 5 distinct stages in the decision process, outlined in Table 6.

<p><b>PROPOSAL:</b> The dynamic proposal and refinement of a set of decision candidates in a decision task.</p> <p><b>ARGUMENTATION:</b> Generation of arguments (or evidence) for and against proposed decision candidates.</p> <p><b>ANNOTATION:</b> Logical evaluation of decision candidates and associated evidence to annotate candidates as possible, eliminated and so on for further use.</p> <p><b>RELATION:</b> Recording significant patterns of combination, such as compatibility of statements, to determine which distinct sets of candidates to evaluate.</p> <p><b>EVALUATION:</b> The use of strong and/or weak quantitative and/or qualitative methods to combine evidence for and against sets of decision candidates to produce absolute or partial rankings of selections or assessments.</p>
--

Table 6: Stages in Decision Making in Fox's Scheme

(Fox *et al.* 1988; after Clark, 1990).

Such procedures operate on statements that consist of application-specific facts (eg positive signs of gastric ulcer include patient is elderly, investigations of acute breathlessness include chest X-ray), including those of inheritance (eg kinds of disease include gastric ulcer), application-specific parameters (eg conditional probability of weight loss given cancer = 0.7), and generic decision making knowledge (eg diagnoses can be derived from a symptom by examining its causes) (Clark, 1990). Fox *et al.* (1988) have used this separation to be able to incorporate and use statistical information into the knowledge base *when it is available*. This may provide a coherent and graceful means of combining qualitative and quantitative uncertainty methods into a single framework.

Whilst having some similarity with both Cohen's (1985) Theory of Endorsements and nonmonotonic/default logics, the efforts of Fox and his co-workers have indirectly addressed the problems inherent within the former approaches. Fox's terms of possibility, plausibility etc may be used to indicate imprecision within a statement, whilst retaining a means to use default information by virtue of the generic decision making knowledge. Fox's semantic system also

provides a coherent framework of evidence combination, aggregation and other reasoning strategies.

Fox's proposals have been developed in a prototypical large knowledge-based system that addresses the knowledge of a medical general practitioner, the Oxford System of Medicine, or OSM (O'Neil *et al.*, 1989). The construction of such a large and ambitious system has problems in constraints on time, effort and data availability. Clark (1990) notes that such constraints require systems to be able to use Fox's 'weaker' qualitative methods of handling uncertainty, but also preferentially address probabilistic uncertainty measures when they are available. Such an approach is highly functional, for several reasons; it makes accessible a range of uncertainty methods that can be suited to the type of uncertain information available within a domain; it is suited to the construction of large-scale applications, due to its modular approach to knowledge representation, and ability to reason in the absence of quantitative information; its modularity and formulation of statements in natural language allows users to browse, assert information in a variety of ways, and be provided with acceptable explanations/justifications of the reasoning process. However, the OSM is still in a relatively early stage of development, and has not been fully formalised, and its computational efficiency and tractability have yet to be properly explored.

### 5.5 Quantitative and Qualitative Approaches to Uncertainty: Discussion

Whilst numerical techniques are currently the most widespread means of addressing uncertainty, there has long been a sense of discomfort throughout AI in handling uncertainty in this way (Saffioti, 1987). Table 7 compares numeric and non-numeric approaches to handling uncertainty. Points 1-5 express common criticisms of numeric methods. Notable criticisms include, that to express, via a single number or set of numbers, the uncertainty inherent within a statement, the arguments for and against that statement, its conditions of applicability, utility, and so on, is often a source of unease to members of the expert system construction team, particularly when such numbers fit ambiguously into any particular scheme (Saffioti, 1987).

Numeric Methods	Symbolic Methods
1. Single number or set of numbers is a 'compilation' of various types of knowledge about a statement, including partial knowledge, arguments for and against, etc.	Allows expression of various types of knowledge.
2. Not easy to use: combination, propagation and inference of uncertainty via numbers is unclear to users, and may not retain the 'meaning' of the original numbers: also leads to unacceptable explanations.	Has criteria of 'ease of use': combination/propagation/inference of uncertainty via logical 'argument' easy to understand.
3. Human do not reason about uncertainty via numbers.	Resembles human reasoning more closely than numeric methods.
4. High precision not justified.	Ability to express uncertainty 'vaguely' is more akin to human reasoning.
5. Unsuitable ranking of hypotheses may occur.	Available to higher-level control therefore available for more "intelligent" ranking of hypotheses.
6. Tools and usage widespread and commonly used, with long history.	Tools and usage rare and isolated with short history.
7. Can handle partial belief.	Cannot handle partial belief (except Fox's work eg Fox 1986b).
8. Computationally efficient.	Less computationally efficient.
9. Carry uncertainty through evidence combination and inferencing.	Uncertainty must be resolved before combination/inferencing performed.

**Table 7: A Comparison of Numeric and Non-Numeric Approaches to Uncertainty**  
(after Cohen, 1987b; Fox, 1986b; Olson, Willers *et al.*, 1990; Saffioti, 1987)

The dissatisfaction with quantitative approaches lies in numeric approaches combining a wide and structured body of knowledge into a flat scale, making such knowledge inaccessible to the system using it (Davis, 1979). Also, the absence of explicit reasons for belief and disbelief in statements makes current numeric methods fail in what Cohen (1987b) refers to as *representational adequacy*. This lack of explicitness is compounded in evidence combination, where it becomes unclear to users how each statement and its attached numeric uncertainty contributes to the conclusions of a system. Additionally, if it is accepted that numbers are meaningful to users, there is no guarantee that the combining functions preserve the meanings of the numbers combined (Cohen, 1987b). Fox, Barber *et al.* (1980) note that the output and justifications of numerically-based inferencing systems specifically in the case of medicine '*are opaque and hard to assimilate to a practical strategy of managing a particular patient*', and it is highly conceivable that such opacity and difficulty in assimilation may occur in a wide range of fields. Fox, Barber *et al.* (1980) add that there is a need for inferencing systems to match a user's own style of reasoning more closely. This general dislike of the process of trying to quantify all aspects of uncertainty may be expressed as a failure to fulfil an *ease of use* criterion (Cohen, 1987b).

A further criticism of numeric methods is the assertion that such approaches to handling uncertainty are not the way that humans think (eg Cohen, 1985), a viewpoint supported by several workers (eg Kahneman *et al.*, 1982). This point relates to the observation that humans have difficulty in interpreting the numerical uncertainty generated in Bayesian systems. Zimmer (1986) notes that:

...processing knowledge about uncertainty categorically, that is, by means of verbal expressions, imposes less mental work load on the decision maker than numerical processing.

Bhatnagar and Kanal (1986) suggest that this is because more clear and explicit interpretations are possible from verbal categories than from numeric values. Such explicitness also improves the human-computer interaction; Pauker (1984) notes, in reference to the domain of medicine, that;

useful explanation for the clinician must involve symbolic reasoning and the provision of a logical argument.

Explicitness not only makes explanation easier, it is possible for knowledge made explicit to be amenable to the implementation of higher-level control (Clark, 1990). Fox (1986b) notes the human use of qualitative terms to express logically distinct states of certainty (eg "plausible", "possible", "probable"), whilst



Saffioti (1987) notes the difficulty in trying to express such states on a flat numerical scale. The work of Tversky and Kahneman (1973) further indicate the non-use of numerical methods in human reasoning by showing that people are often insensitive to statistical criteria (eg sample size, variance etc), but tend to rely on a small set of heuristics in handling uncertainty.

Another major problem with numerical methods is that of unjustified precision; numeric uncertainty values are typically too precise when compared with the precision our knowledge justifies, and it is unclear when a difference in numerical values is too small to be meaningful. Numeric methods have also been criticised in terms of their use in ranking and comparing hypotheses (Saffioti, 1987).

Despite the criticisms of numeric methods, several arguments in their favour exist. Olson, Willers *et al.* (1990) argue that a primary aim of expert systems is to produce results efficiently and accurately, rather than mimic human reasoning at the expense of these criteria, and that this criteria is best satisfied using numeric approaches. Bhatnagar and Kanal (1986) note that numeric methods carry uncertainty through evidence combination and inferencing, whilst systems using non-numeric methods must typically resolve uncertainty by making assumptions before combination or inferencing is performed. This reduces computation efficiency, but allows results to be revised when new information is received.

Numeric methods of the probabilistic school have a longer history and more refined methodologies than non-numeric methods. Whilst the more common numeric (particularly probabilistic) methods are bounded (and possibly limited) by assumptions of independence, when such assumptions are valid, a greater precision in evidence combination is achieved relative to non-numeric methods (Clark, 1990). Numeric methods are widespread and commonly used, which may perhaps be an artifact of the long history of numeric methods of dealing with uncertainty, the associated commonness of tools that allow the handling of numeric uncertainty, and the utility of such approaches.

## 5.6 The Approach to Uncertainty in Paradigm-Based Reasoning

In Chapter 4 of this work, it is argued that the nature of ecological deep reasoning is essentially qualitatively-based. It is also argued that reasoning in such a domain must aim to reproduce the paradigm-based level of reasoning. Given the nature of the domain, along with the limitations of numeric methods, qualitative approaches seem the most appropriate means of addressing uncertainty in this work. However, in considering qualitative methods, a number of problems arise; tools that allow use of these techniques are not available, and literature reports of qualitative

techniques of uncertainty calculi are theoretically rather than practically-based. Thus the nature of this work, focussing on the practical implementation of paradigm-based reasoning with respect to habitat evaluation, is not well suited to any one *existing* technique (as far as can be ascertained) within either set of approaches. Paradigm-based reasoning uses an expression that embodies specific types of relationships (eg causes, indicates, etc), via a type of extended production rule (a 'relationship'). Part of this 'relationship' is discussed in Section 4.5.2, and is of the form:

<Object1> having <Attribute1> that is <Value1> <Relation> <Object2>  
having <Attribute2> that is <Value2> .

eg *A pond having a light intensity that is low causes this pond to have a productivity that is low.*

Using classical logic approaches, a statement such as the one above must be considered as either always true, always false or sometimes true (with an unspecified frequency). However, consideration of the problems of previous approaches, and suggestions for remedying these problems allows the formulation of an approach to handling uncertainty that is fundamentally qualitative. This approach integrates features from several other qualitative approaches (using methods based within non-classical logics (Saffioti, 1987), Cohen's methods of balancing evidence (Cohen, 1985), and Fox's semantic system (eg Fox, Clark *et al.*, 1990)), but also uses some of the considerations and practical methods of numeric approaches, such as the evidence combination 'heuristics' of certainty factors, and the issues that motivate the workers exploring fuzzy logic. This integrated approach may be included as a fundamental part of the above formalism, and is so developed and described in the next section.

### 5.6.1 The Qualitative Conviction Calculus: 'Convictions' as the Units of Uncertainty

Mamdani *et al.* (1985) note that predicate calculus uses two kinds of quantifiers in expressing terms, the *universal* and the *existential*. The universal quantifier can be expressed in the expression 'in all cases', whilst the existential can be expressed in the term 'in some cases'. Examples of these include '*in all cases, newts are amphibians*', and '*in some cases, cars have 4 wheels*'. The former means that the associated expression is always true, whilst the latter means there is at least one case where the associated expression is true (this can go from a single instance, to all instances). Mamdani *et al.* (1985) go on to note that many other quantifiers

exist in language, such as 'many', 'most', 'a few', 'hardly any', and so on. These quantifiers typically allow users and recipients a reasonable (though imprecise) depiction of what is occurring, its frequency, a mental view of the subject, and the level of uncertainty felt by the assertor of the statement. Johnson-Laird (1977) notes that it may be possible to derive axioms that utilise such quantifiers in a logical system, and goes on to note an absence of any psychological work on inferencing with quantifiers, other than the use of the universal, the negated universal ('in no cases'), and the existential within the narrow framework of syllogistic logic. Johnson-Laird (1977) further asserts that there is a need to understand how the larger set of quantifiers is used in human reasoning.

The idea of including quantifiers derived from language, and used in much the same sense as the original linguistic quantifiers, has been carried forward in the work of Zadeh (1986) and Fox (1984). Zadeh (1986) notes the limitation of classical probability theory in its inability to deal with an expanded range of quantifiers, such as 'most', 'many' etc, and adds that this limitation makes a variety of statements that are easily understood by humans, but are inaccessible to probabilistic reasoning mechanisms; for example, 'most small cars are unsafe', 'Brian is much taller than most of his close friends', etc. Zadeh (1986) further asserts that fuzzy logic can successfully deal with such expressions (typically by mapping vague terms into numeric sets). Fox (1984) proposes a similar approach, but suggests using such terms in a qualitative reasoning framework. Fox (1984) suggests that the 50 to 100 English words for describing facts and data (eg "possible", "probable") can be about arranged into a framework of belief terms (Slatter, 1987a). These can then be used to reason '*qualitatively with an explicit semantics, not numerically with an implicit semantics*' (Fox, 1984). Fox (1984) gives an example rule:

If patient *could be* suffering from disease  
and disease is *definitely* fatal  
then patient *may be* in danger.

Such qualitative reasoning by machine, using an explicit set of semantics, is of benefit in many ways. It is equivalent to logical argument and reasoning, a rational approach from the human perspective. This feature may make explanation and justification facilities of a reasoning system more acceptable, intuitive and sensible to users than traces of rules and facts where uncertainties are expressed by numbers. Such an approach also retains a coherent 'imprecision' in its conclusions (ie 'crested newts *are very likely* to be absent' rather than 'crested newts *are* absent'), overcoming a common problem of numeric methods - having an unjustified precision.

A proposal of this research is that it is possible to attach to statements a set of linguistically-derived quantifiers, which function as units of uncertainty. These units are referred to in the rest of this thesis as **convictions**. The framework in which convictions are used is termed the **Qualitative Conviction Calculus (QCC)**.

The following set of convictions are used in the PERSEUS system:

1. "In all cases"
2. "In virtually all cases"
3. "In most cases"
4. "In many cases"
5. "In roughly half of the cases"
6. "In some cases"
7. "In few cases"
8. "In very few cases"

If the first quantifier is accepted as a special case, that of absolute certainty, then the rest of the quantifiers form a seven-point scale of certainty that may be attached to terms. The number of categories (levels of uncertainty) is an important issue, as too many may lead users to become confused about the differences between adjacent categories, whilst too few may lead to categories that are too broad, resulting in impoverished information and poor decision making (Fayers and Jones, 1983). It was found in discussing uncertainty levels in the crested newt case study that expressions such as 'in virtually all cases', 'in all cases, with the odd exception', 'almost always' were commonly used by the expert. The commonness of such expressions suggested a need for the presence of convictions that meant 'in all cases, with very infrequent exceptions' ("*in virtually all cases*"), or 'in no cases, with very infrequent exceptions' ("*in very few cases*") (points 2 and 8 in the above list). The remaining 5 ordered categories (3-7) can be viewed as a set of rankings such that point 5 corresponds to being correct about half of the time. The members of this set are in descending order of the certainty attached to an associated statement (ie statement 'X, in virtually all cases' is believed more strongly than 'X, in some cases'). Note that the 6th conviction, "in some cases" is not equivalent to the existential quantifier of logic. "In some cases" means a proportion of a set, typically less than half, in both everyday speech (Mamdani *et al.*, 1985), and in the qualitative conviction calculus, whilst in syllogistic logic it means any proportion above zero, up to and including **all**. The generic form of the final expression that is produced by joining the extended production (introduced in 4.5.2) and convictions is:

< Object1 > having < Attribute1 > that is < Value1 > < Relation > < Object2 >  
having < Attribute2 > that is < Value2 > < Conviction > .

Such terms are designated **Statements of Conviction**. Examples of such statements of conviction include:

*A pond having a light intensity that is low causes this pond to have a productivity that is low, in virtually all cases.*

*A pond having anglers present indicates this pond has fish present, in most cases.*

It is a goal of this work to determine whether statements of this form are meaningful to both builders (ie knowledge engineers and experts), users and casual observers of the knowledge base. If such terms are intuitively sensible, they may be used for explanation and justification of a reasoning process with little or no further processing.

The QCC approach attempts to combine certain positive attributes of both numeric and qualitative approaches to handling uncertainty. A conviction may be seen as something that is an attachment of a measure of uncertainty to a statement or fact, in much the same way as a certainty factor. However, the qualitative nature of a conviction may be more easily interpreted by users than a numeric value, as the mechanism for handling uncertainty in the QCC is more akin to logical reasoning than numeric combinations used in quantitative methods.

A full description of the qualitative conviction calculus is presented in Chapter 6, as its description involves proper consideration of one of the main issues addressed in Chapter 6: control in intelligent systems. This description will include further discussion of how the QCC, and statements of conviction, compare and contrast with both quantitative and qualitative methods of handling uncertainty.

## CHAPTER SIX

### KNOWLEDGE COMPILATION AND MANIPULATION

#### 6.1 Introduction

A key element of the utility of knowledge based systems is in their ability to use real-world knowledge in a rational and tractable way. In the chapter which follows, several aspects relating to this tractability are examined.

This examination begins with a description of the common methods used for directing the order in which knowledge is utilised (typically called 'control knowledge'), and how this ordering is specified within knowledge based systems, along with associated problems. This leads into a consideration of how 'control' of knowledge can be made more efficient and timely, and how this aspect relates to deep knowledge, as well as its implications on human and machine reasoning. The approach to control and its efficient use within the *qualitative conviction calculus* is reported, and discussed in specific detail. The approach is then compared to other means of handling knowledge, and the potential benefits of this approach are summarised. Other 'second-generation' aspects are also discussed, in relation to the PERSEUS architecture.

#### 6.2 Reasoning with Knowledge

The basic architecture of an expert system has been described in Section 1.1, and pictorially represented in Figure 1. It is traditionally regarded as good practice in knowledge-based system construction to keep separate the knowledge that is used (referred to as the **knowledge base**), and the subsystem that reasons using this knowledge. This subsystem is typically called the **inference engine** (Forsyth, 1989). The inference engine uses some means of searching through, and subsequently reasoning with, the knowledge base. Two very common reasoning strategies are found in knowledge-based systems that use production rules: **forward chaining** and **backward chaining** (Jackson, 1990).

Forward chaining involves the collection of evidence or data, and such data may eventually be used in reaching a goal or conclusion. This approach is easily implemented, and is typically used in situations where all of the data is routinely gathered, such as that already stored or collected in some predetermined format. For this reason, it is sometimes referred to as 'data-driven/directed' reasoning.

Backward chaining is the opposite process: a specific goal is considered, and

suitable evidence is gathered in order to establish a value for the goal. This typically leads to a more natural dialogue with users than a forward chaining approach, since at any stage the system may be able to explain why it is asking a question relative to its goal (Forsyth, 1989). It may also require less data to come to the same conclusions, by avoiding the input of unnecessary data. Backward chaining is sometimes called 'goal-driven' or 'goal-directed' reasoning.

Reasoning founded on backward chaining generates a further problem in search. If the search space is thought of as a tree, with the base as the goal, the leaves as data or evidence, then it is possible that many different paths are viable if the direction of search is from base to leaves. This means that backward search must involve some method of directing that search.

Two particular patterns of search commonly occur in many knowledge-based systems: **depth-first search** and **breadth-first search**. Using the tree analogy, depth-first search involves systematically going to each fork (ie splitting of the tree into two branches) until it reaches a terminal leaf. On failure to satisfy the goal, the search would return to the last fork it encountered, and then go to the next leaf. With breadth-first search, an alternative pattern is used. The nodes (ie points of forking) are first checked at each level of forking (ie level 1 would be those nodes in a direct, unforked line to the base, level 2 nodes would have 1 fork between them and the base etc), before proceeding to the next. It is asserted by some authors that breadth-first search will, on average, find the solution in the quickest time, if one exists, as it typically finds the shortest path between the goal and some evidence that proves the goal (eg Jackson, 1990). Figure 4 illustrates the patterns that these two search strategies tend to take.

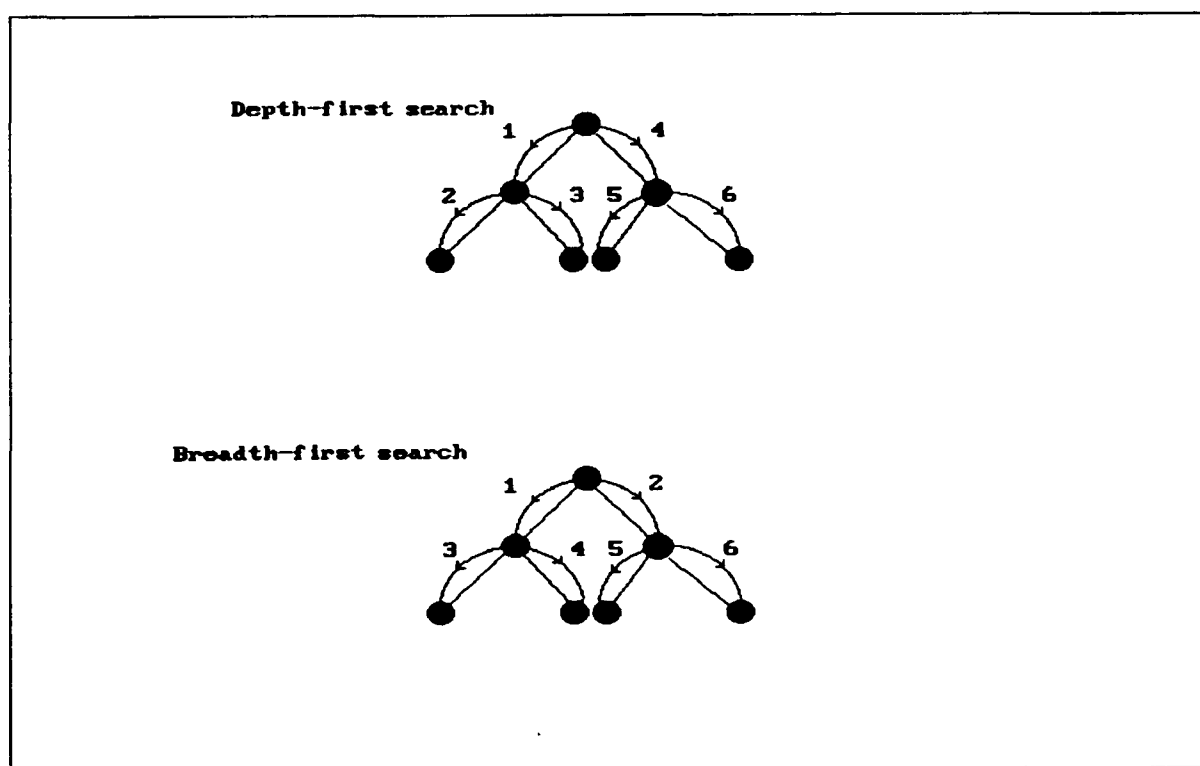


Figure 4: Depth-first and Breadth-first Search (after Jackson, 1990)

Jackson (1990) notes that such search spaces are exhaustive, ie all possible paths must be searched before it is obvious that the goal cannot be instantiated. In large search spaces, exhaustive search quickly becomes time-consuming, and is often intractable. Davis (1980) refers to this problem of unrestricted search being an unrealistic tactic in large spaces as *saturation*.

Various approaches have been devised to restrict the potential search space. For example, the LEONARDO expert system shell employs backward chaining with 'opportunistic' forward chaining as a default search strategy (Creative Logic, 1987). Buchanan and Smith (1989) note that such an opportunistic approach can '*set up expectations that help discriminate a few data elements from an otherwise confusing mess*', ie focus attention on apparently important elements within the system.

Other workers have implemented other types of search strategy, that may be best referred to as **best-first** approaches. Bratko (1990) discusses such an approach, where 'costs' are assigned to the paths between nodes in the search space, and search begins by assessing the 'cheapest' paths first. Naylor (1989) discusses another 'best-first' approach, where rules are used in their order of 'certainty', (ie more certain rules are used in favour of less certain ones).

### 6.3 Control as a Distinct Type of Knowledge

Recently, much criticism has been made of reasoning strategies based on backward and forward chaining. In knowledge-based systems that use these reasoning systems, the flexibility of the approach to problem-solving is poor. One of the main criticisms of such approaches is that the control of the questioning strategy resides in the ordering and structuring of the rules themselves. This higher-level knowledge of the order in which lower-level knowledge should be used is typically called **control knowledge** (Aikins, 1980), or **strategic knowledge** (Clancey, 1983). Clancey (1983) defines control knowledge as that which '*specifies when and how a program is to carry out its operations*'.

The problem of control residing within the lower-level knowledge will now be considered in detail, along with possible methods to address this problem.



### 6.3.1 Implicit Control in Rule-based Systems

Common criticisms concerning control in typical first generation knowledge-based systems include the observation that control of the ordering and use of knowledge is often rigidly applied, and not sensitive to the context of the case. The type of formal representation of knowledge used in first generation systems often makes such knowledge unsuited to sophisticated control.

Aikins (1980) notes that production rules, the representation commonly used in first generation KBS's, are in theory modular pieces of knowledge which each capture some piece of knowledge pertinent to the domain. She goes on to note that:

In practice, however, there are significant interactions among rules. Executing one rule will in turn cause others to be tried when the information needed for the first rule is not already known. Therefore the order of the premise clauses of a rule affects the order in which other rules are executed [and ultimately the ordering of questioning] (Aikins, 1980).

The order in which knowledge is used is therefore dependent upon two aspects in rule-based systems; the ordering of the individual rules within the knowledge base; and the ordering of antecedent parts of the rules (ie the *if...* parts of 'if-then' rules). Familiarity with these aspects allows the control of questioning to be dictated by the knowledge engineer. Whilst this is an apparently useful facility, it means that in such systems the control knowledge is embedded within lower-level knowledge, and is both opaque and implicit. Bainbridge (1988) notes that, in the case of the rule-based system MYCIN, the control is '*wired in*' and therefore not easily available for '*examination, changing or reasoning*'.

The implicitness of control in such systems has led for calls to treat control knowledge as a separate entity in itself (eg Davis, 1980; Clancey, 1983). This is perhaps not surprising, as Saffioti (1987) notes that a characteristic of AI research is its concern with representing and using knowledge in the most explicit form possible. The motivations, and possible methods for treating control knowledge as an explicit entity are discussed in the following section.

### 6.3.2 Explicit Control in Rule-based Systems

The main motivation to make control knowledge an explicit entity has been the recognition of the failings of implicit control. Expert systems, as with any computer systems, often require modification and improvement (Clancey, 1983). When control is implicit, it is sometimes difficult to predict the outcome of changes

in the knowledge base, as rule changes may affect control within the system. Davis (1980) notes that making control knowledge explicit would also make it more easy to prevent bugs in control occurring in the first place.

In abstracting control from the main knowledge base, the remaining knowledge may become less goal-oriented, and it may then be available for a wider variety of tasks, when different control procedures are applied. An example of this may be a goal-independent body of knowledge about ecosystem interactions, where different control strategies may be used for a range of tasks, such as assessment, management, and so on. Aikins (1980) suggests that the formulation of explicit control makes the generation of explanation easier, as it becomes more straightforward for users to understand how the decision process occurs. In many rule-based systems, explanation is provided by showing the rules that are in present usage. Two types of rules are typically used, one to infer one set of information from another (eg *if fish are present, newts are absent*), and the other involves rules or parts of rules which are specifically for the control of a questioning strategy. With the latter type of rule, it is frequently not obvious how some parts of the rule relates to the goal at hand. As Aikins (1980) states:

...the uniform representation of control and inference knowledge in rule-based systems further confuses the user by mixing the two kinds of explanations.

Additionally, explicitness would make control knowledge more easily available for change (Bainbridge, 1988; O'Neil, Glowinski & Fox, 1989).

A further point that arises is that abstract control knowledge may be a closer approximation of how humans reason. Different control strategies may equate to 'plans' in human beings. Examples of systems that attempt to handle control explicitly include CENTAUR (Aikins, 1980), and the Oxford System of Medicine (Fox, Glowinski *et al.*, 1988).

#### 6.4 Knowledge Compilation

Whilst abstracted control knowledge has several benefits, such abstraction may lead to the requirement of a greater amount of knowledge being used at each stage of the reasoning process (ie the inference engine and non-control knowledge being indirectly connected, and linked via the control knowledge), which may reduce efficiency in processing knowledge, leading to slower reasoning. One method of overcoming reduced efficiency whilst still handling control explicitly is via the

automated *reintegration* of control and other knowledge into a single representation. This involves the conversion of existing knowledge into different forms that are more efficient when used in reasoning. Such conversion is a type of **knowledge compilation**. The term "knowledge compilation" was coined by Neves and Anderson (1981) to name a specific cognitive activity identified in humans, relating to skill acquisition. This is further discussed in Section 6.4.2.

Brown (1991) notes that knowledge compilation is a human mental activity which is a type of learning from experience, and defines knowledge compilation as a process where:

existing knowledge is converted to new forms, with an intent to improve problem-solving efficiency.

This definition is equally applicable to compilation in machines. Brown (1991) also notes that there is no single definition of knowledge compilation that is agreed upon by all researchers.

As well as increasing efficiency and a change in the representation level of the knowledge, other features of compilation include a reduced amount of reasoning required in solving a problem, and '*a decrease in explicitness or transparency*' (Keller, 1991). Also, knowledge is typically altered to be more task-oriented, or is transformed from a general to a focussed form, suitable for a very specific set of uses (Sembugamoorthy and Chandrasekaran, 1986).

#### 6.4.1 Knowledge Compilation and Depth

Within AI literature, there is a general recognition that there exist two approaches for the integration of deep and shallow knowledge (eg Punch, 1992). The first approach involves the use of a uniform reasoning process that utilises two or more different knowledge bases, each with successively deeper knowledge. The second method is by the compilation of deep knowledge into "shallower" forms, so that the shallow knowledge is effectively constructed relative to the goal at hand. This latter approach is a more integrative one, and the newly-compiled knowledge can be transposed back into the deeper form, if necessary, which may prove of use in explanation and justification facilities. It is the latter approach that is used in the present research.

It must be noted that there has been an unfortunate trend in AI literature for some workers to use the term 'compiled knowledge' as a synonym for 'shallow knowledge' (eg Bylander *et al.*, 1988). While understandable, as the shallow

knowledge derived from experts must have been compiled by those experts at some stage, this unfortunate referral to "shallow knowledge" based in machines as "compiled knowledge" is confusing. In this work, when discussing reasoning processes in machines, "compiled knowledge" will mean knowledge that has been generated within a machine from other knowledge.

#### 6.4.2 Knowledge Compilation as a Cognitive Process

The term "knowledge compilation" was coined by Neves and Anderson (1981) to describe the process whereby humans go from an '*interpretive application of knowledge to direct application*'. These two workers attribute their choice of wording to the similarity of the process they describe with the compilation of a computer program. In learning a new skill or piece of knowledge, human beings must first be consciously aware of the steps involved. As their experience grows, intermediate steps become unconscious or unnecessary. Learning to multiply numbers, or to drive, are examples of this. The earlier form of the knowledge is typically referred to as **declarative** or **factual** knowledge, whilst the later, more functional knowledge is referred to as **procedural** knowledge. The declarative knowledge is typically independent of any function, whilst the procedural knowledge is heavily task- or goal-oriented (Anderson, 1983). Riese and Zubrick (1985) describe declarative knowledge as '*a list of independent facts without explicit control or execution sequencing information*'. Given this definition, it is apparent that declarative knowledge is control-free, and the act of compilation adds control knowledge to the knowledge base.

Neves and Anderson (1981) assert that knowledge compilation involves two steps; *proceduralisation*, the translation of declarative statements into procedural rules, and; *composition*, the combination of procedural rules into larger rules. An example of proceduralisation is the conversion of the declarations:

*most types of fish will eat all the newt larvae in a pond*  
*the complete predation of a population's young, before they have reproduced, will result in the local population's extinction*  
*newt larvae are the young of the population at a state prior to reproduction*

to a more procedural form:

*if fish are present in a pond, crested newt are likely to become absent in this pond*

Proceduralisation of knowledge has several associated advantages and disadvantages. Characteristics of procedural knowledge in human machine reasoning include; once acquired, procedural knowledge operates in a fast, automatic fashion; in humans, it takes little or no cognitive resources, and occurs with little awareness; it is relatively unavailable for verbalisation; it is generally closed to introspection; and the acquisition/development of procedural knowledge is often unconscious (Gordon, 1989).

#### 6.4.3 Knowledge Compilation in Machines

It has long been recognised by some workers that the "shallow" rules used in first generation knowledge based systems are typically long chains of inference compressed into simple rules, often containing implicit control knowledge (eg Barnett, 1982). A method by which this process could be exploited in intelligent systems has been suggested by Chandrasekaran and Mittal (1983a), who showed how task-specific (ie procedural) knowledge for medical diagnosis can be compiled from deeper (ie declarative) structural-behavioural-functional (SBF) models of systems. A variety of knowledge-based systems have been built to date that compile functional (ie "shallow") knowledge from deep, SBF models. Sembugamoorthy *et al.* (1986) use this approach in generating functional knowledge about a front door buzzer, whilst more recent work has involved several groups compiling functional knowledge from SBF models of the human cardiovascular system (Bratko *et al.*, 1988; Bylander *et al.*, 1988; Mozetic, 1990). The compilation of "deeper", declarative knowledge to task-oriented forms is central to the present research, and is described in Section 6.5. Whilst some workers have suggested the need to utilise "deep" knowledge in the ecological domain (eg Noble, 1987), and others have discussed the need to integrate ecological "concepts" into AI-based ecological models (Davey and Stockwell, 1991), there is currently no forthcoming work that tackles the compilation of deep (ie paradigm-based) knowledge into task-oriented forms within the ecological domain, other than the research described within this thesis.

#### 6.4.4 The Potential Benefits of Knowledge Compilation in Machines

Several benefits accrue from the compilation of declarative knowledge to procedural knowledge in intelligent systems. A knowledge base that is mainly declarative is typically easier to alter and maintain than a more procedural system -

procedural knowledge representations (such as certain production rules) tend to have interactions that are implicit and hard to follow. Knowledge in a declarative form is also easily understood when browsed by users (Araya and Mittal, 1987). Compilation offers a means to integrate distinct sorts of knowledge into a form more efficient for problem-solving (ie more procedural). Brown (1991) notes that the use of declarative knowledge means that the knowledge is typically easier to reuse for different goals and applications than more goal-specific knowledge. Brown (1991) also suggests that compiled systems may be more robust than first generation KBS's, as the knowledge present in compiled systems is not as 'highly tuned' to a specific goal, and no arbitrary boundaries to the knowledge will be present. Lavrac and Mozetic (1989) suggest compilation as one way second generation knowledge based systems may acquire functional/shallow knowledge, overcoming the problem of knowledge acquisition being the "bottleneck" in KBS construction. Attarwala and Basden (1985) suggest that the acquisition of deep knowledge from an expert may be easier than shallow knowledge, and that deep knowledge is typically found in domain textbooks.

Dietterich (1991) notes that new technologies typically have no human experts, only people that are familiar with the deep knowledge. Human expertise typically takes several years to develop, whilst machine compilation is significantly faster. Dietterich argues that compilation may therefore be used to generate useful problem-solving knowledge in the absence of experts.

Additionally, compilation is a process common in humans as expertise develops. In using compilation as a tool to "generate" expertise, there may be a closer match between human reasoning and machine reasoning than was present in first generation systems. This point is discussed further in Section 6.5.

Araya and Mittal (1985) note that explanation generation relates to compilation, typically being the reverse process. The shallow knowledge is used in efficient reasoning, but when a user wants to understand the underlying principles of the shallow knowledge (ie how a question relates to the goal), "decompilation" can be done that shows what pieces of knowledge or chains of inference are involved.

Overall, the benefits of knowledge compilation are the maintenance of efficiency whilst pursuing the benefits of keeping knowledge as explicit as possible. Tong (1991) best describes the motivations behind research into knowledge compilation:

We are adding herein a new twist to Roger Bacon's saying that 'Knowledge is Power', by using knowledge compilation to squeeze more power out of the same knowledge (Tong, 1991).

## 6.5 Compilation within the Qualitative Conviction Calculus

Before discussing how compilation occurs in the Qualitative Conviction Calculus (QCC) proposed and considered within this thesis, it is necessary to review the form in which knowledge is represented within this calculus.

There are three types of representation used in the QCC (see Section 4.5.2 and 5.6.1). These consist of **facts**, **inheritances** and **relationships**.

Facts have the generic structure:

<Object> <Attribute> <Value> <Conviction>.

Examples of this include:

<*Ecosystem*> <*productivity*> is <*high*>, <*in some cases*>.

<*Pond*> <*shading*> is <*low*>, <*in about half of the cases*>.

(NB: Superfluous words are allowed in all three units of representation, to make the terms more natural).

Inheritances have the generic structure:

<Object 1> <Object 2>

where Object 1 is a subset of object 2.

Examples of this include:

<*Ponds*> are kinds of <*ecosystem*>.

<*Duckweed*> is a kind of <*floating vegetation*>.

Relationships have the form:

<Object 1> <Attribute 1> <Value 1> <Relation> <Object 2>

<Attribute 2> <Value 2> <Conviction>

Examples of this include:

<*Pond*> <*species diversity*> being <*high*> <*indicates*> that <*pond*> <*productivity*> is <*low*>, <*in most cases*>.

<*Pond*> <*water clarity*> being <*high*> <*is equivalent to*> <*pond*> <*turbidity*> being <*low*>, <*in all cases*>.

Note that the two "relations" in the above examples, "indicates" and "is equivalent to" are different in kind. The former is directional, so that the first object-attribute-value (O-A-V) tells us something about the latter, but the reverse is not true. This type of relation is termed **asymmetric**. The latter relation, "is equivalent to", is non-directional. Here, the value of either of the O-A-V's may be used to give a value about the other. This type of relation is termed **symmetric**. The actual words used as asymmetric and symmetric relations are chosen by the system builder using

PERSEUS, (see Appendix C), but in the present implementation (TRITON), the asymmetric relations used are *cause(s)* and *indicate(s)*, whilst the symmetric relation is "*is equivalent to*".

Within the present work, the knowledge compilation process focuses mainly on relationships, but prior to this, some compilation occurs involving inheritance terms being combined with other inheritance terms, facts and relationships to form new terms. The method of combination for these three processes, along with examples, are given in Table 8. In typical first generation knowledge-based systems that use inheritance, such combination is done whilst the system is reasoning, but in the present research, the combinations occur prior to running (ie during compilation), making the final reasoning process less time-consuming.

Once the compilation of inheritance terms with other terms is performed, the remaining and more significant part of the compilation process occurs; the compilation of chains of relationships into single terms. This involves two main aspects, the combination of relationships (detailed in Table 9), and the combination of convictions that exist within the relationships.



(A) The Combination of Inheritance Terms

Inheritance: <Object C> <Object B>.

+

Inheritance: <Object B> <Object A>.

---

Inheritance: <Object C> <Object A>.

Example:

Inheritance: <Pond>, are kinds of <aquatic ecosystem>.

Inheritance: <Aquatic ecosystems> are kinds of <ecosystem>.

---

Inheritance: <Ponds> are kinds of <ecosystem>.

(B) The Combination of Facts and Inheritance Terms

Fact: <Obj A> <Attribute A> <Value A>, <Conviction>.

+

Inheritance: <Obj B> <Obj A>.

---

New Fact: <Obj B> <Attr A> <Val A>, <Conv>.

NB: A new fact is only created if there is no previous fact about <Obj B> <Attr A> <Val A>. In this way, the system builder's assertions will not be overridden by default. The same is true for new relationships.

Example:

Fact: <Ecosystem> <productivity> is <high>, <in some cases>.

Inheritance: <Ponds> are kinds of <ecosystem>.

---

New Fact: <Pond> <productivity> is <high>, <in some cases>.

(C) The Combination of Relationships and Inheritance Terms

Relationship: <O1-A1-V1> <Relation> <O1-A2-V2>, <Conviction>.

+

Inheritance: <O2> <O1>.

---

Relationship: <O2-A1-V1> <Relation> <O2-A2-V2>, <Conviction>.

Example:

Relationship: <Ecosystem> <species diversity> being <high> <indicates> <ecosystem> <productivity> is <high>, <in most cases>.

Inheritance: <Ponds> are kinds of <ecosystem>.

---

Relationship: <Pond><species diversity> being <high> <indicates> <pond> <productivity> is <high>, <in most cases>.

Table 8: The Combination of Inheritance Terms with Facts and Relationships

Relationship: <O1-A1-V1> <Relation1> <O2-A2-V2>, <Conviction 1>.  
 +  
 Relationship: <O2-A2-V2> <Relation2> <O3-A3-V3>, <Conviction 2>.  
 -----  
 Relationship: <O2-A1-V1> "implies" <O3-A3-V3>, <Conviction 3>.

If the "convictions" are ignored for this example, an example of this type of combination would be:

<Pond> <species diversity> being <low> indicates that <pond> <productivity> is <low>.

+

<Pond> <productivity> being <low> causes the <pond> to be <unsuitable> for the <crested newt>.

-----  
 <Pond> <species diversity> being <low> implies that the <pond> is <unsuitable> for the <crested newt>.

-----  
 The chain of inference can be any number of steps long, with the generic "implies" relation used to express connection.

Example: A indicates B, B causes C, C is equivalent to D, D causes E, can become: A implies E.

Table 9: The Combination of Relationships within the Qualitative Conviction Calculus

6.5.1 Constructing Chains of Reasoning within the QCC

In combining relationships, chains of inference are created that may contain any number of relationships. Each relationship has an associated conviction. These convictions (see Section 5.6.1) are listed below.

1. "In all cases"
2. "In virtually all cases"
3. "In most cases"
4. "In many cases"
5. "In roughly half of the cases"
6. "In some cases"
7. "In few cases"
8. "In very few cases"

When a set of relationships are combined (as described in Table 9), the associated convictions are also combined to form a single conviction. When no convictions are repeated in this series of relationships, the newly created relationship takes the weakest conviction from that series. A suitable analogy is that of a chain, where the strength of the chain is that of its weakest link.

An example is:

- 1) *A pond that has fish present indicates fish predation of crested newt larvae is present, in virtually all cases.*
- 2) *Fish predation of crested newt larvae being present causes significant loss of crested newt larvae from the pond, in many cases.*
- 3) *Significant loss of crested newt larvae from the pond indicates the pond is unsuitable to support a viable crested newt colony, in virtually all cases.*

which compiles to:

*A pond with fish present implies the pond is unsuitable to support a viable crested newt colony, in many cases.* (Note that this last statement should not suggest that the pond is suitable to support a colony in the remainder of cases; it means that given a pond has fish present, one may be reasonably certain that the pond is unsuitable for crested newts. Note that the conclusions of this calculus err towards conservatism, and the attached conviction may be taken as an expression of the minimal level of affirmation for the latter part of the statement - *the pond is unsuitable for crested newts* - given the former - *fish are present*).

This "weakest link" approach is intuitively sensible, as any chain of reasoning can only be as strong as its weakest element. However, there is a further refinement to this approach. The "weakest link" approach becomes less acceptable when a chain of reasoning contains several equally weak links. In such cases, it may be sensible to consider that the compiled relationship is *weaker* than its weakest conviction. Consider the theoretical chains:

- 1) *A implies B, in all cases;*  
*B implies C, in all cases;*  
*C implies D, in some cases; and*  
*D implies E in all cases.*

This compiles to: *A implies E, in some cases.*

- 2) *A' implies B', in some cases;*  
*B' implies C', in some cases;*  
*C' implies D', in some cases; and*  
*D' implies E' in some cases.*

Using the "weakest link" method, this compiles to: *A' implies E', in some cases.*

In considering these examples, it can be seen that (1) contains only one conviction of "in some cases", whilst (2) contains several. It does not seem reasonable that they should be treated as equally certain. For this reason, a process of knowledge senescence is proposed, whereby when any two links of the same strength occur, they are aggregated to become a single conviction that is one conviction lower on the scale (eg "*A implies B, in some cases*", and "*B implies C, in some cases*" becomes "*A implies C, in few cases*"). This combination occurs throughout a set of relationships until no conviction is repeated in that set. (Example 2 will therefore compile to *A' implies E', in very few cases*). There are two exceptions to the general rule of knowledge senescence. This is when a compiled set of relationships contain two convictions of either "in all cases", or "in very few cases". The latter conviction is the weakest possible (meaning it is possible, but very unlikely), so cannot be further weakened. The former conviction ("in all cases") is a statement of absolute certainty, and so any number of relationships that are absolutely certain will form a compiled relationship that will also be absolutely certain (ie be true "in all cases"). At a technical level, when a pair of either the weakest ("in very few cases") and strongest ("in all cases") convictions occur in a set, the pair are converted to a single conviction of the same level.

The approach to compilation of relationships used in this work is similar to an architecture proposed by Chandrasekaran, Smith *et al.* (1989). Their approach, however, is restrictive in terms of being only causally orientated. Within the present work, it is recognised that important interactions occur that are non-causal (eg indicative relations, associated conditions, etc), and the architecture described by Chandrasekaran, Smith *et al.* (1989) has been broadened by allowing non-causal relations to be expressed (as illustrated in Figure 5, and expressed by the general relation "implies").

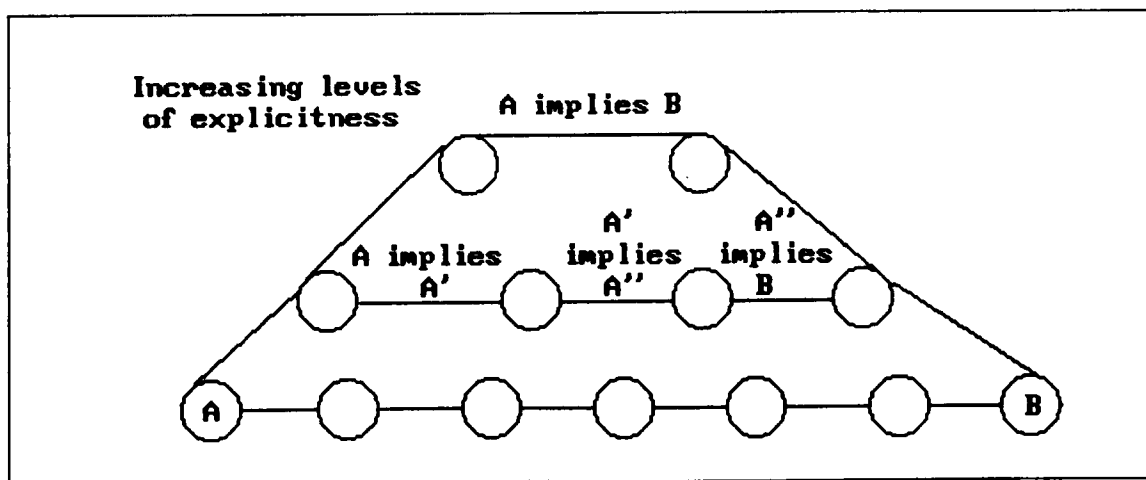


Figure 5: Increasing Levels of Detail Underlying Compilation

## 6.6 Consideration of Evidence within the Qualitative Conviction Calculus

It should be noted that the QCC, and its implementation in PERSEUS, does not utilise an exhaustive search. In fact, the QCC is based upon a *directed* search, so that a rational value is reached within a tractable framework. This is properly discussed in Section 6.6.4.

### 6.6.1 Identifying an Outstanding Value

Once compiled, knowledge is used in the QCC to try to find a value for a pre-specified goal (ie an attempt is made to find the value of a given object-attribute that has been declared as the goal eg *<pond> <suitability for the crested newt>*). The search for a suitable value for the goal proceeds by considering those facts and relationships that are at the strongest level of conviction (ie "in all cases"). If no instantiation can be made at the strongest conviction level, then the knowledge base is scanned at subsequently weaker levels of conviction.

At each level, it is assessed whether a value for the goal is stated as a fact. If this is not so, relationships are then considered to see if any can be used to infer a value for the goal. A value may be identified in three ways; from knowledge already available within the knowledge base, stated as facts; from direct knowledge provided by the user; and via inference from user inputs, using relationships.

### 6.6.2 The Prominence of a Value

Once a potential value has been asserted at a particular conviction, the knowledge base is scanned to see if there exists any terms that may warrant alternative values for the goal. If any such terms are proved, and are at the same level of conviction as evidence supporting the proposed value, then equal evidence for alternative values exists. There is no longer a value for the goal that is **prominent**. The remainder of the knowledge base is further considered. Search continues using this set at the current level of conviction, to see if further evidence exists which supports a single value as being "prominent". This technique is similar to the "ledger book approach" of Cohen (1985), where evidence for different values of a goal are used to "cancel" each other out.

If a "prominent" value does not emerge at a given conviction level, then the search recommences at the next lowest level of conviction, considering all possible values for the goal that occur in the knowledge base. This continues until a value is

found to be prominent, or the knowledge base is exhausted (in which case the search for a value is inconclusive). In the case of a value being prominent and there is not conflicting evidence, a further process occurs, and is discussed in the next section.

### 6.6.3 Collating Supporting Evidence in the QCC: Knowledge Renaissance

When a value becomes prominent in the QCC, a process occurs whereby further evidence is gathered to see if the conviction level attached to the goal value can be strengthened. If another term is found of the same conviction level that supports the prominent goal value, then the overall level of conviction is raised by one level. A generic example is:

*A implies Z, in few cases. + B implies Z, in few cases.*

If A and B are true then:  
*A and B implies Z, in some cases.*

The goal value may become more certain as more evidence is found to support it, rising a level as appropriate values are found. Such combination can be considered analogous to logical reasoning in decision-making. Doctors, for example, will recognise likely hypotheses to test given some symptoms, and will direct further investigations with respect to these hypotheses. A "prominent" value can be thought of as a hypothesis that has been accepted, and the continuation of information-collecting can similarly be considered the gathering of further evidence to support the hypothesis.

In gathering further evidence, checking occurs to make sure that there are no interactions between antecedents **at or above the current level of conviction** (eg *A* and *B* in the above example), so that no dependencies are present. The nature of the compilation process makes such checking necessary, as it may well be that *B* is an intermediate of *A implies Z* (or vice versa). To combine *A* and *B* in such a case would lend more 'conviction' to the goal (*Z*) than it deserves, as the same evidence is being used twice.

The entire process, whereby lower-level terms are searched to see if they can be utilised to strengthen the level of conviction, and then combined to raise the current level of conviction, is termed **knowledge renaissance**. This combination can go on at any level, except for the top two levels: "in all cases" and "in virtually all cases". The former expresses absolute certainty in the goal value, and it is therefore impossible to "strengthen" it. The latter can never be moved to the next stage of certainty, as this is absolute certainty, and by definition a statement with the

conviction "in virtually all cases" has some element of uncertainty. It is therefore illogical to try to strengthen a value that is certain "in virtually all cases".

The above approach to evidence combination is inspired by, and is fundamentally similar to both the pro-con, "ledger book" approach of Cohen (1985) (Section 5.4.2, subsection *Cohen's theory of endorsements*) and the "argumentation" approach of Fox and co-workers (Fox *et al.*, 1988) (Section 5.4.2 subsection *Fox's semantic system*). As such, it shares many of the benefits of these two approaches. It is a good mechanism for providing explanation (maintaining a history of arguments for a specific goal value), is possibly a closer match of human reasoning than numeric methods of handling uncertainty (as it involves logical reasoning about evidence, rather than numeric methods), has a way of handling default information in the absence of more specific information (via asserted facts and inheritances), uses natural expressions as a knowledge representation scheme, offers a rational scheme of evidence combination, and is (relatively) computationally efficient.

Appraisal of this calculus as embodied within the TRITON/PERSEUS implementation, relative to both a first-generation knowledge-based system (HEX), and the conclusions (and associated uncertainties) of a domain expert considering real data, is presented in Chapter 9.

#### 6.6.4 The Qualitative Conviction Calculus as Directed Search

The architecture of the QCC is such that terms of stronger conviction are always considered before terms of weaker conviction. It shall be noted that there may be instances in human decision-making where a single piece of evidence of strong conviction supporting a particular value are outweighed by many pieces of "weaker" evidence. Within the PERSEUS system, however, it is not feasible to examine all the possible combinations of evidence for and against each possible goal-value in all but the smallest of knowledge bases. In particular, an attempt to compile all weaker knowledge terms into stronger terms results in an intractable and unmanageable knowledge base. Equally, attempts to examine such combinations at run-time prove equally intractable in terms of speed of execution, even on personal computers with extremely fast processing capabilities (ie with a 486-Intel chip architecture).

Knowledge engineers utilising the QCC approach must be aware that the directed search detailed within this thesis, and used in PERSEUS, may not be appropriate in knowledge bases containing large numbers of weak terms, particularly where large amounts of knowledge of weak conviction may typically be considered of equal or higher standing than a few pieces of knowledge of strong conviction.

Within the TRITON application (detailed in Chapter 7), no difficulty was experienced with the directed search strategy described within this section.

## 6.7 Control within the Qualitative Conviction Calculus

The interaction of control knowledge and uncertainty has already been discussed in Section 5.2, where the work of some researchers suggests that uncertainty is a type of control problem (Clark, 1990; Cohen 1987a,b). Within the QCC, control and uncertainty are interlinked. PERSEUS, the knowledge-based system shell upon which the QCC operates, has control structures which are geared to examine the knowledge base for more certain information first, focussing upon the convictions attached to facts and relationships. The search proceeds by level of conviction rather than ordering of terms (as with many rule-based systems). In this way, the search is a type of 'best-first' search similar to that developed by Naylor (1989) (see Section 6.2), and the ordering of terms in the knowledge base is not significant to the reasoning strategy (though it may have some effect on the order of questioning). The control knowledge is not a separate entity within the QCC, but is embedded within the QCC as convictions. In traditional rule-based systems, control is established in two ways; the ordering of rules, and the ordering of antecedents within individual rules. The terms within the QCC that are comparable to rules having only one antecedent, so no ordering of antecedents is available. Likewise, the order in which terms are used depends upon their attached convictions, rather than arrangement within the knowledge base.

## 6.8 A Comparison of the Qualitative Conviction Calculus and other Control/Uncertainty Formalisms

The qualitative conviction calculus can be viewed as an approach that attempts to combine elements of existing quantitative and qualitative techniques, seeking to preserve the benefits of each, whilst minimising the faults inherent with each.

Like many of the qualitative approaches, the methods used within the QCC are more akin to logical reasoning than numeric methods. The comparative processes in numeric uncertainty handling are typically unclear to users, and difficult to explain after it has occurred. The implementation of the QCC calculus within PERSEUS is efficient, and operates at a speed comparable to the faster numeric uncertainty calculi (ie probabilistic/certainty factor methods). Explanations generated by the QCC may be clearer than those generated under numeric/rule-based methods, as the QCC may be a closer match to logical (human) reasoning. The QCC can perhaps be seen as a qualitative version of probabilistic/certainty factor approaches to uncertainty, where the numeric measure of uncertainty (the probability/certainty factor) has been replaced by a qualitative measure (the conviction).



## 6.9 Information Assertion and Revocation in PERSEUS

Keravnou and Washbrook (1989) comprehensively review the limitations of first generation knowledge-based systems. Many of these limitations have already been addressed within this thesis (including inclusion of depth, and maintainability of the knowledge base via modularity of knowledge). Two further limitations identified by Keravnou *et al.* (1989) have been addressed in the present research, and these are now discussed. These are; the facility for users to volunteer information (and an associated criticism that first generation systems have too strict a format for the dialogue between human and machine), and; the ability of users to revoke an answer to a question, or to pursue the effects of an alternative answer (to see "what if...").

These limitations are essentially design problems, which are addressed within PERSEUS. The PERSEUS shell has been constructed so that the builder can specify one of three modes of operation for a finished knowledge based system. The finished system may be "user-driven", where the user selects values for object attributes about which the system reasons, it may be "machine-driven", where the system asks questions of the user to come to its conclusions, or it may be "combined mode", where the user may initially assert values to object-attributes, but the system may ask further questions if they are necessary. Additionally, "object adjustment" may be selected as a higher-level control option within PERSEUS. With this enabled, a user may change, or assert any value at any time. This aspect addresses both limitations. This "object adjustment" feature operates via a mechanism based upon the **Truth Maintenance System** of Doyle (1979) (see Section 5.4.2 subsection *nonmonotonic and default logics*). In changing a value, the entire knowledge base is reassessed in terms of the new value(s). This eliminates any possible inconsistencies that may otherwise occur, and increases the functionality of the PERSEUS shell.

## 6.10 Characteristics of the Qualitative Conviction Calculus

It has been suggested within this and previous chapters that there are several benefits to be gained from the use of the qualitative conviction calculus, as opposed to other reasoning/uncertainty calculi. These may be summarised as; the QCC may be a closer match to human reasoning than other machine reasoning methods; it offers a viable method to handle uncertainty in a purely qualitative manner; it has been implemented as a working tool (PERSEUS), and is available for testing/use; and it offers a means to use declaratively-expressed knowledge, and has the associated benefits of declarativeness. The means of testing the QCC (via PERSEUS/TRITON), and the analyses of these tests are covered in Chapters 8 and 9 respectively.

CHAPTER SEVEN  
THE DEVELOPMENT OF FIRST-GENERATION AND SECOND-GENERATION  
KNOWLEDGE-BASED SYSTEMS: A CASE STUDY

### 7.1 Introduction

In this chapter, the practical aspects of the knowledge engineering process addressed within the present research are discussed. This covers both those aspects that are specific to the evaluation of a pond site for suitability to support crested newts, and the more general aspects that relate to the differences (and similarities) in approach to constructing first- and second-generation knowledge-based systems.

The chapter begins with a discussion of the reasons why the case study considered within this research is suitable for a knowledge-based systems approach. This leads into a description of the ecology of the crested newt, and includes a brief discussion of the parameters that have been identified as being central in determining the suitability of a pond to support a viable colony of crested newts. This is followed by a review of the knowledge engineering process within this research, describing the various stages of development, including; the acquisition of knowledge; the requirements of a second-generation 'shell' as specified in this research; design of the systems within this study (HEX, and TRITON); the construction of these systems, together with difficulties encountered during design and construction; and a discussion of the relative costs and benefits in building HEX and TRITON. This chapter then summarises the similarities and differences in building first- and second-generation systems in a specific case involving habitat evaluation, and concludes with a commentary about the overall construction of first- and second-generation knowledge-based systems in general.

### 7.2 Habitat Evaluation as a Suitable Focus for this Study

In contemplating whether expert systems technology is suited to the evaluation of pond sites with respect to a single species, it is necessary to consider those characteristics of a domain/problem which render it suitable for the methodologies of knowledge-based systems. Waterman (1986) suggests there are three issues in considering whether some domain is suited to an expert systems approach; is expert system development possible, justified, and appropriate? Each of these issues will be discussed generally, and then with respect to habitat evaluation.

### 7.2.1 Is the Use of Expert Systems in Pond Evaluation Possible?

To make possible the construction of an expert system, there must usually be one expert available who is able to solve the problem at hand in an efficient manner. In the evaluation of a pond site with respect to crested newts, an expert (Dr. Rob Oldham) has been available for the duration of the research.

### 7.2.2 Is the Use of Expert Systems in Pond Evaluation Justified?

The justification of using expert systems technology in a domain depends upon identifying a requirement for a system. The development of a computer system that can assess a pond for its suitability to support a viable crested newt colony may be considered worthwhile, as this species is currently protected under the Wildlife and Countryside Act (1981). Effective and available identification of pond sites that can or may support this species may aid in the species' protection. However, individuals with sufficient expertise to be able to accurately identify ponds as suitable or unsuitable for the crested newt are uncommon, and the use of a computer system which allows access to such expertise in the absence of human experts is arguably justified in such circumstances. In addition, the translocation of crested newts to newly-developed sites may be considered a viable conservation strategy, and proper assessment of suitability of these sites is key to such ventures.

### 7.2.3 Is the Use of Expert Systems in Pond Evaluation Appropriate?

The appropriateness of a domain to expert systems methods depends in part on the nature and methods of the problem-solving skills engaged within the domain. The skills required should involve some accrual of evidence that, by cognitive processes, leads to some conclusion. This evidence should be identifiable to future users of the system, and the time required by the expert should, *on average*, take more than a few seconds, but less than a few hours. At either extreme, the task is either too trivial (and better suited to traditional programming approaches) or too complex to be suitably addressed by expert systems methods (Waterman, 1986). Other possible criteria that indicate that a task is appropriate for expert systems techniques include; the common use of uncertain or incomplete data; a frequent requirement for detailed justification of questions and proper explanation of judgements; and that the domain at hand is similar to other domains in which expert systems are employed (Beckman, 1991).

In assessing a site, the expert typically takes 15-40 minutes. Occasionally, assessment is longer (in cases that are borderline or unusual), whilst in other cases, assessment is virtually instant, as the pond is obviously unsuitable (eg when it is highly polluted).

The domain of habitat evaluation requires that a number of issues be addressed. It contains uncertainty (as detailed in Chapter 5), requires adequate explanation/justification facilities (to adequately explain this specialised domain), and is similar to other domains within which expert systems have been employed. The evaluation of a habitat is essentially a diagnostic process, and similar domains, such as diagnostic medicine, contain an abundance of expert system developments (eg Buchanan and Shortliffe, 1984).

### 7.3 The Ecology of the Great Crested Newt

The great crested, crested, or warty, newt (*Triturus cristatus*) is one of 6 amphibian species native to Britain, and one of 2 amphibian species (the other being the natterjack toad, *Bufo calamita*) protected under the Wildlife and Countryside Act (1981). Creatures protected under this Act are those, according to Section 22(3), which are '*in danger of extinction in Great Britain or are likely to become so endangered*' (Frazer, 1989). The vulnerable status of this species indicates that, though widespread, it is a species that is diminishing in numbers as time passes.

Newts have an elongated body and tail, with short legs. *T. cristatus* has a warty, rough skin, the upper parts being a darkish brown colour, with rounded black spots. The underside is bright yellow or orange, with black markings. In the breeding season, the male bears a prominent crest along its back, which is used in courtship displays occurring within the water. The length of adults varies between 80 and 142 millimetres, with a mass up to 14 grams (Frazer, 1989).

This species is distributed throughout most of mainland Britain, though it is rare or absent in north and west Scotland, and scarce in Cornwall, Wales and the remainder of Scotland. It is widespread but not common over the remainder of mainland Britain (Hilton-Brown and Oldham, 1991; Swan and Oldham, 1990).

From late summer to spring these newts are found in various refugia, such as under stones or in holes, where they hibernate. Breeding starts with adults returning to water in early Spring, staying in breeding condition for 2-3 months. Sexual maturity occurs between the ages of two and four years, depending upon the amount of food taken, which will itself vary with temperature (Oldham and Nicholson, 1986). Movement towards the pond occurs only at night in the absence of frosts, males apparently gathering in the ponds before females. For breeding the crested

newt favours clear water, particularly where there is little emergent vegetation, though they are able to breed in cloudy water with little vegetation of any kind (Frazer, 1989). The development of eggs and larvae takes a variable amount of time, depending upon temperature, and late developing larvae may overwinter in a larval state, though larvae produced in a given year normally metamorphose within the same year.

Zuiderwijk (1984) regards the adult crested newt as more aquatic than adults of other newt species. Hagstrom (1979) states that the average length of the adult aquatic phase is 13 weeks (at a Swedish site), but established that the adults spend part of this time on land, feeding.

The crested newt feeds on a wide variety of invertebrates, varying according to what types happen to be abundant locally. For example, Creed (1964) notes that, in the New Forest, at one site crested newts existed almost solely upon corixids, and at another site on *Daphnia* species. Crested newt larvae, like the adults, are carnivorous, feeding mainly upon small crustacea. Oldham and Nicholson (1986) gives a summary of the diets of adults and larvae recorded in the literature. They record that adults have been noted to feed upon worms, slugs, caterpillars, slow worms (Steward, 1969), crustaceans, insects, snails (Green, 1984), *Daphnia* species, corixids (Creed, 1964), and the tadpoles of frogs and toads (Cooke, 1974). The juvenile diet is similar, though is based on smaller individuals of similar species or morphology to animals eaten by the adult (Robinson, 1977).

The causes of the diminishing numbers of crested newts are numerous. Frazer (1989) notes several reasons, such as the destruction of ponds, many associated with the provision of piped water and troughs for farm stock, the effects of toxic chemical pollution such as DDT upon tadpoles (Cooke, 1972) and the effect upon adults that take prey that is contaminated with such chemicals, the loss of terrestrial habitat to land development, and hydrosereal succession. Other important excluders also exist, including high concentrations of toxic metallic ions (Cooke and Frazer, 1976), the presence of predators such as fish, grass snakes and water birds (Frazer, 1989), and eutrophic pollution. The most important factor in recent times is likely to be pond senescence (Oldham and Nicholson, 1986).

### 7.3.1 Attributes that influence the success of the Great Crested Newt

The characteristics of a pond identified within the current research as being of some importance to the presence of crested newts are many and varied. Those that have been recognised as being to some degree dependent have been grouped into a single 'attribute'. The 'attributes' identified and validated by the expert are presented in Table 10.

- |   |
|---|
| <ol style="list-style-type: none"> <li>1) The terrestrial plant community surrounding the pond</li> <li>2) The aquatic plant community within the pond</li> <li>3) Pond size/depth</li> <li>4) Drawdown/permanence of the pond</li> <li>5) Shading of the pond</li> <li>6) Pond pollution</li> <li>7) Fish presence</li> <li>8) Duck presence and numbers</li> <li>9) The location in the UK, and altitude.</li> <li>10) The presence of barriers, such as walls, roads, and rivers.</li> </ol> |
|---|

Table 10: Attributes that Influence the Success of *Triturus cristatus* within a pond

Many of these attributes are to some degree interrelated. For example, the attribute of pond shading reduces productivity and therefore the nature of the aquatic plant community, is indicative of a certain type of terrestrial vegetation adjacent to the pond (ie trees), may be an indicator of excessive leaf-fall (and therefore pollution) of the pond, and may be indicative of conditions that prevent water fowl colonising the pond (ie excessive tree overhang).

#### 7.4 The Knowledge Engineering Process within this Study

This section contains details of the various aspects of the knowledge engineering process within the present research. This includes details of general knowledge engineering, and information specifically relevant to the entire development of first- and second-generation knowledge-based systems in habitat evaluation.

##### 7.4.1 Pre-development Requirements

Prior to commencing any project involving expert systems, a number of considerations must be made. The suitability of the domain is one such consideration (see Section 7.2). Others include the choice of tools, an explicit recognition of likely

users, expert involvement, and the minimum specifications of the system. These, and some other minor considerations, are summarised in Table 11. Significant areas will be discussed in the rest of this section.

- |  |
|--|
| <ul style="list-style-type: none"><li>• The goal(s) of the initial system.</li><li>• The target users of the system.</li><li>• Hardware requirements.</li><li>• Software tools, or at least generic types of tool.</li><li>• Other resources available (funding, materials, equipment).</li><li>• The overall user interface, including facilities wanted (help, explanation, etc).</li><li>• Potential long-term aims.</li><li>• Time scales for different parts of the project.</li><li>• Members of the ES development team explicitly identified (knowledge engineers, experts, systems designers, etc).</li></ul> |
|--|

Table 11: Considerations preceding development of an Expert System

### **Target Users**

Stock (1988) notes that serious and formal identification of users and their needs are central to proper implementation and acceptance of a system. In the applications considered here, the primary users were identified as pond recorders, and land/conservation managers and workers. Such users are likely to have some knowledge of the domain, but lack the necessary experience to accurately predict pond suitability. Secondary users that were identified included students of ecology/environmental management, using the systems' explanation and justification facilities as an educational tool.

### **Expert Involvement**

Keyes (1989) suggests that a main reason for the failure of any expert systems project is the lack of an available expert. Other reasons include the participation of too many experts, each with a set of idiosyncratic knowledge and

opinions, resulting in a lack of consensus and little progress.

Once an expert is identified, it must be evaluated whether this person is truly 'expert' in the task at hand (ie have suitable experience). The expert's commitment to a project must be ensured, the expert must be willing and cooperative, and must be able to communicate knowledge effectively (Prerau, 1985). In addition, the expert should be available on a regular basis, there should be a reasonable certainty the expert will be accessible for the duration of the project (eg retirement is not due), and the expert should not feel threatened by the system (Slagle and Wick, 1989).

The expert used in this project is Dr. Rob Oldham, a professional scientist who has for many years been involved in research into amphibian ecology, and is Head of the Ecological Research Laboratory in the Department of Applied Biology, at De Montfort University, Leicester. This laboratory is the coordinating body of the National Amphibian Database, containing the records of pond and habitat surveys from the entire range of mainland Great Britain. Dr Oldham has several years' experience of the task problem, and is both intellectually engaged in the project, and articulate.

### **Hardware Specifications**

The specification of an expert system requires some consideration of the types of machine to which the target user groups are likely to have access. With HEX/TRITON, development has proceeded based upon the following minimum specification: PC-based, with a VGA screen, a minimum 8086 processor, and a hard disk of any size.

### **Selection of Software Tools**

The selection of tools relates to the domain, the nature of the problem being tackled, and the requirements of the user interface.

With the first-generation system, HEX, an expert system "shell" (a high-level tool for constructing expert systems), was employed, called LEONARDO (Creative Logic, 1987). LEONARDO is well suited for tackling problems that require a diagnostic approach, with good facilities for creating an acceptable user interface. It is available on the required platform (IBM-compatible PC's), and has an acceptable speed of execution.



The specialised requirements of the second-generation system TRITON (eg the inclusion of qualitative uncertainty) discounted the use of a shell in construction. PDC Prolog was selected as a suitable tool with which to implement TRITON. This tool has all the advantages of the PROLOG language (detailed in Section 4.5.2), can be compiled into a fast, executable form for the IBM-compatible PC, and has many features that facilitate the construction of knowledge-based systems (eg tools to build acceptable user interfaces; tools that allow fast searches of large knowledge bases). PDC-Prolog was used to construct the higher-level tool PERSEUS, and TRITON was then constructed within PERSEUS. The decision to build an intermediate tool that is based upon the qualitative conviction calculus (QCC) architecture was prompted by a number of potential future requirements. These requirements fall into two main groups. The first is the provision of a readily-available means to evaluate the viability of the QCC approach using either a similar task within the ecological domain, or a task within a different domain that is similar to ecology in terms of being a 'soft', concept-rich field (see Section 1.2.1; Chapter 2). The second requirement is a need to provide second-generation tools/facilities to future developers of knowledge-based systems.

### **The Pre-Development Requirements for PERSEUS**

In developing the qualitative conviction calculus, as described in Chapters 3, 5 and 6, it became apparent that no available higher-level shell or development environment for knowledge-based systems had the appropriate facilities available. Problems encountered in existing shells included the inability to express the knowledge-forms required for the QCC (ie the 'facts', 'inheritances' and 'relationships'), inability to implement both the 'knowledge senescence/renascence' process and the best-first search strategy that underlies the inferencing process required by the QCC, and the absence/inaccessibility of a variety of other aspects (eg changing/revocation of values of objects within the system) relating to more general second-generation system requirements (listed in Table 12). It should be noted that some of these latter facilities may be individually found in certain tools, but no single tool presently available possessed all of them. Requirements of the second-generation system as proposed in the present research are listed in Table 12.

Appendix B gives full details of the complete development of the PERSEUS shell. Its details include the particulars of the initial specification, design, implementation, and evaluation required to ensure that PERSEUS was a tool that could be viably used to construct TRITON.

<p>A) Specific to the Qualitative Conviction Calculus</p> <ol style="list-style-type: none"> <li>1) A means to express knowledge as 'facts', 'inheritances' and 'relationships'.</li> <li>2) An inferencing process that utilises the 'best-first' search, using successively weaker levels of conviction as the focus of this search.</li> <li>3) An inference process that may rationally combine terms using 'knowledge renaissance'.</li> <li>4) A compilation process that creates paths of reasoning from individual terms.</li> <li>5) A compilation process that undertakes 'knowledge senescence' - ie proper 'weakening' of the line of reasoning as terms are 'compiled'.</li> </ol>
<p>B) Relating to Second-Generation Systems</p> <ol style="list-style-type: none"> <li>1) An ability to change/revoke/add values to objects at run-time (both during consultation, and at the point of conclusion).</li> <li>2) An ability to generate rational explanations/justifications of reasoning directly from the knowledge base which are coherent to users.</li> <li>3) Sophisticated parsing methods (ie allowing the recognition that some terms are equivalent to others eg 'ponds' and 'pond') - this allows terms in the knowledge base to be written in a more realistic and coherent way.</li> <li>4) An ability to change the goal of the system without major changes being required by the knowledge base.</li> <li>5) Ability to use a variety of operational modes in the interface between machine and user (eg the PERSEUS system allows 'machine-driven', 'user-driven' and 'mixed-mode' processing - see Section 6.9).</li> <li>6) A method of easily changing the questioning strategy to allow different levels of questioning to be directed to different types of user (eg questions specifically for novices and above, near-experts and above, or full experts only).</li> </ol>

Table 12: Facilities of Second-Generation Knowledge-Based Systems

Required in the Present Research

## 7.4.2 Knowledge Acquisition

The acquisition of knowledge has long been recognised as the major bottleneck in the development of knowledge-based systems (Buchanan *et al.*, 1983). Knowledge may have several sources, including text books, technical documentation, databases, one's own experience, near or partial experts in a domain, and most commonly, full experts. Human experts possess the type of knowledge (*heuristic knowledge*, or "experience") suited to accurately and efficiently solve specific problems, and the majority of successful expert systems rely on knowledge acquired from sessions involving the expert and knowledge engineer. The interaction of knowledge engineer and expert has many well-documented problems introduced by both parties; these are listed in Table 13.

- |  |
|--|
| <ul style="list-style-type: none"><li>• Poor communication skills in either party.</li><li>• Problems of making previously-implicit knowledge explicit.</li><li>• Inaccessibility of the expert.</li><li>• Lack of expert enthusiasm for the duration of the project.</li><li>• The use of the knowledge engineering process as a foil for developing new ideas by the expert.</li><li>• The inexperience of the knowledge engineer.</li></ul> |
|--|

Table 13: Potential Problems in the Interaction between Knowledge Engineer and Expert (after Buchanan *et al.* 1983; Hart, 1986; Welbank, 1983)

Many different techniques exist for the acquisition of knowledge, and these are reviewed in several papers (Cain, 1990; Neale, 1988; Welbank, 1983).

Within the present research, certain techniques have been used extensively, particularly interviews and the recording, transcription, and subsequent analysis of these interviews. Such techniques have often been criticised within the literature (eg Burton, Shadbolt *et al.*, 1987) as being inefficient compared to other approaches. Parsaye and Chignell (1988) note that interviewing is sometimes referred to as 'traditional knowledge engineering'. Despite these criticisms, interviewing proved to be highly successful in generating both heuristic and deep knowledge (in the form of explanations of the heuristic knowledge). The expert was interviewed in familiar surroundings and allowed to express his knowledge on his own terms, and meetings were determined by his availability. The recording and transcribing process allowed the 'capture' of much knowledge that may have been otherwise missed, and proved

effective in this particular research.

Other techniques were tried within this research, but some were felt to be unsuitable for various reasons (ie the expert felt constrained, or did not provide the expected knowledge). An example of a 'failed' method is card-sorting (where the expert was asked to sort cards - with 'concepts'/pond attributes written on - into various piles, using specified criteria eg 'detrimental to newt presence'). Other techniques were tried and found useful, but were used only occasionally, as they were difficult to perform on a regular basis for a variety of reasons (eg time-consuming, generated little knowledge relative to interviews, etc). An example of this is 'think aloud' protocol analysis, where the expert visited a pond site (accompanied by the knowledge engineer), and talked through his decision-making processes (recorded onto audio tape) as they occurred. The exercise was of some utility, especially at the start of the project, but was felt less productive than interviews as the project proceeded (ie as more specific knowledge became the focus of discussion).

As stated, knowledge acquisition meetings were recorded on audio tape. Once made, the audio recordings were transcribed onto paper for easy analysis and reference. The lines of the transcript were numbered, and important factors were highlighted within the text (see example in Table 14). At the end of the text, highlighted factors were listed alphabetically, followed by the line numbers where each occurred. Following the transcript, a short report was written by the knowledge engineer discussing the more important aspects of the meeting, including new knowledge generated, how well the meeting went, and possible targets/questions for future meetings. Additionally, a record of the date of meetings where factors occurred was kept in another file (the 'mentions' file). In this way, if details of any concept/factor needed reexamination, the 'mentions' file could be examined to determine the interviews where a factor was discussed. These interviews could then be reviewed.

The nature of the meetings in this project were determined by the knowledge known at any one time. The majority of meetings were in the form of an interview, which was transcribed and analysed. As the meetings proceeded, the questions became more focussed and detailed, and were often responded to by the expert with appropriate short answers (often 'yes' or 'no').

Table 14 depicts a section from an early interview. Here, the knowledge engineer was recognising and highlighting potential factors important to decision-making. As the project proceeded, a base set of factors equivalent to features that directly affect a pond's suitability for crested newts became apparent, and only these were used. Previous factors were often grouped in this base set eg the concept of 'overgrowth' was considered a facet of the pond's 'plant community'.

1 Mark Cain: Now, what I wanted to talk about today was how you  
2 actually assess a site. In your own words.

3 Rob Oldham: It struck me that the best way to do this may be  
4 to actually go to one but regrettably we can't.

5 MC: Well, we could do that next time... At the moment I can  
6 see myself being available virtually from any day now until  
7 the beginning of September. I haven't actually got anything  
8 on.

9 RO: Good.

10 MC: Apart from meetings with Derek (Teather).

11 RO: I suppose we could actually go to one of our ponds just  
12 over here. I wonder if that would be at all helpful?

13 MC: Maybe if we did that at the next interview. If we  
14 discussed the process now, what would be helpful is if I  
15 could watch you do it, perhaps, and make comments on a hand-  
16 held machine, or you hold the machine and speak into it, as  
17 you're doing the task, and when we come back, discuss those  
18 in more detail. The reason for this is so that you do not  
19 have to keep disturbing your train of thought with  
20 explanations of what you are doing.

21 RO: Yes. There are probably going to be differences in terms  
22 of the approach as I do it in my imagination than if I were  
23 doing it in reality. However, I'm just going to picture a  
24 site, and it doesn't really matter whether we choose the  
25 terrestrial or aquatic aspect first, because clearly they are  
26 both important. The terrestrial habitat..., if we are aware  
27 of where the pond is going to be as we're approaching it,  
28 then the terrestrial habitat is what strikes us first. And,  
29 in that respect, what we are looking for is diversity of  
30 vegetation (**Diversity(Terrestrial, Plant)**), particularly in  
31 respect of the ground cover (**Cover(Terrestrial, Plant)**). It  
32 is possible to have high diversity (**Diversity(Terrestrial,**  
33 **Plant)**) in a forest, but we're interested in a newt's  
34 perspective. The newt is going to require cover  
35 (**Cover(Terrestrial, Plant)**), and food (**Food**  
36 **Source(Terrestrial)**), so we're looking for habitats that are  
37 likely to provide that. And so far we are using mainly  
38 intuition in making that link. This is something we are  
39 investigating in our present project, the extent to which  
40 habitats of different types do provide good food (**Food**  
41 **Source(Terrestrial)**) and cover (**Cover(Terrestrial, Plant)**),  
42 and there certainly does seem to be a correspondence between  
43 habitats that have good diversity (**Diversity**), good ground  
44 cover (**Cover (Terrestrial)**), and good newt populations. We  
45 rarely find good newt populations in intensively managed  
46 areas (**Management (terrestrial)**), whereas we find very good  
47 ones in areas where there are a lot of semi-natural  
48 vegetation.

Table 14: A Section from the Transcript of an Early Interview

As meetings became more structured, there were occasions when transcripts were felt to be unnecessary. Both making the transcript, and subsequent analysis, is time-consuming, so these transcripts are only made when the meetings consisted of discussion. In question-answer situations, only the answers are recorded, written over the original question sheet.

After a few meetings, a set of concepts/factors that clearly affected pond suitability were formally identified, and these were used to generate a set of intermediate scripts, called 'indices'. An example of such an index is given in Table 15, showing the index of 'pond shade'. Within these files, the particulars relating to the factor were kept, including detailed information on how this factor affects the crested newt, how the factor can be reasonably assessed, and how the factor may affect, or indicate the value of, other factors/parameters within the domain. These 'indices' were changed and expanded as meetings generated more information, and proved an very effective way of (i) checking the consistency of the expert's statements, (ii) storing knowledge prior to formal coding, and (iii) integrating knowledge about the domain from sources other than the expert (eg text books).

Such intermediate indices have been referred to as *Mediating Representations* (Johnson, 1985). Such documents are used for a variety of reasons. They allow the knowledge engineer to formulate an understanding of the domain, whilst acting as an indexing and storage device for elicited knowledge. They are a means of integrating knowledge from separate knowledge acquisition sessions and sources, and are a convenient way to maintain and check the coherency of that knowledge. Such representations allow a synthesis of different levels of knowledge, such that the purely experiential knowledge of an expert (eg which aspects are worthy of evaluation) can be put into context of the underlying conceptual, or 'deep', knowledge of the domain (Kuipers *et al.*, 1984; Tunnicliffe *et al.*, 1991).

Whilst this approach to knowledge processing was time-consuming, detailed, and required much attention, it was found to be of considerable benefit to the project as a whole, and has saved much time when consultation with previous sessions has prevented the need for the expert to repeat knowledge that may have been otherwise lost.

There are more structured ways to process knowledge in a similar way to that already discussed, such as using cross-referencing, creation of glossaries, asserting a degree of importance to various factors, annotating transcripts with details from video recording, and so on (Tunnicliffe *et al.*, 1991). However, the degree of structuring is dependent upon the domain, and the time and resources available. Insufficient processing early in a project will result in repeating questions/meetings unnecessarily. Likewise, intense processing in a domain where it is not required

(especially in the later stages) may have the appearance of productivity, but may be as retarding to a project as too little processing. It is up to the knowledge engineering team to decide upon the appropriate balance. In the present research, a high level of processing was initially maintained, with a subsequent levelling to a suitable intensity as meetings became less of a discussion, and more a 'yes/no' answer session. However, recording of each meeting was maintained in case more detailed discussion occurred.

The prototypical HEX system took about 14 months to construct. Table 16 illustrates the pattern of knowledge acquisition within the construction of HEX throughout its life cycle, and shows how the emphasis of knowledge acquisition methods altered at different stages in the project. The overall elapsed time was relatively long, as the construction of HEX was part of a larger body of research, as presented in this thesis. It should also be noted that reasonable gaps between both meetings and iterations were necessary for the knowledge engineering process to facilitate the following; the proper consideration of what areas warranted further questioning; the consolidation of existing knowledge, and; the implementation of well-structured design methodologies. Some delays were also caused by lack of availability of the expert.

### Assessment

This involves an integrated index that involves the (1) proportion of shoreline shaded to 1 metre from edge of shore into the water, (2) the proportion of the shoreline occupied by vegetation of a certain minimal height (about a metre), and (3) the compass direction of the shading. South is the most significant compass direction, so that shading that tends to occur in the Southern end of the pond is the most affecting. It is not just overhanging that is important (though this is the thing that is related to leaf-fall and therefore eutrophication), but shading from objects near the shore line. The further the object is from the shoreline, the less likely it is to have any effect in terms of shading. However, the taller an object is, the more likely it is to cause shading. So a tall object set back from the pond may still have a shading effect. Therefore there must be a geometric relationship between height of objects near the pond and their distance from the pond for them to cause the same amount of shading.

There is a continuum of shading among ponds, rather than discrete classes.

Mary Swan indicates that the median range of shading for ponds containing the Great Crested newt is 1-25%. Mark Cain found, on statistical analysis, that % shading around the pond does not affect newt presence, but south shading does (see 'The Suitability of Parameters').

Rob Oldham considers that trying to state that there is a minimum size, above which shading has little effect, is not quite correct. A large pond that is long and narrow (of a ratio of length:width > 5) can still be strongly affected by shade. Also, if the pond has about 25-75% shading, then >50% shading in the south quarter of the pond is enough to consider the pond unsuitable for the crested newt.

Additionally, heavy shading can be considered as an explicit indicator of low productivity/unsuitable aquatic vegetation complement.

### Effect on the Great Crested Newt, and general information

Shading can reduce water productivity by reducing light, and act as a (possibly beneficial at low levels) eutrophic agent by loading the pond with organic matter. It also reduces warming which will again reduce photosynthesis by generally decreasing rates of chemical reaction. Floating leaves can also act as egg-laying sites. Shading can also increase the relative amount of open water in a body, so being of some benefit to the newt in terms of courtship. It also delays the stage of succession to some degree.

Note that shading also makes the assessment of the newt population directly more difficult by restricting access to or vision of recorders.

### Mentions

09/04/90, 12/07/90, 31/07/90, 09/08/90, 09/08/90, 17/10/90, 03/12/91

Table 15: Shade 'Index' derived from Knowledge Acquisition Sessions



Meeting	Month	Details
1,2	February '90	Preliminary Sessions.
3,4	April '90	
5	June '90	
6,7	July '90	
8	July '90	Consolidation of Elements mentioned in various sources.
9	July '90	More specific knowledge acquisition sessions dealing with subgoals of HEX system.
10	August '90	
11	September '90	
12,13,14	October '90	
15	December '90 ↓	
		Analysis of materials from knowledge acquisition sessions, building of HEX, and other work.
		↓
	March '91	Prototype of HEX completed
		↓
		Meetings 16-21 concerned with considering alternative methods to address "location" as a subgoal of the reasoning strategy.
		↓
22	October '91 ↓	Evaluation of HEX prototype.
-----		
23,24	October '91	Fine-detail knowledge acquisition.
25,26,27	November '91	Analysis of acquisition, refinement of HEX.
28	December '91 ↓	
		↓
29	February '92 ↓	Evaluation of HEX 2.
-----		
30,31	February '92	Refinement of HEX 2, via fine-detail knowledge acquisition.
32	March '92 ↓	
		↓
		Completion of HEX 3.
		↓
33,34	April '92	Refinement of HEX 3, via fine-detail knowledge acquisition.
		↓
35,36	May '92	Evaluation and acceptance of HEX.

Table 16: The Pattern of Knowledge Acquisition pertaining to HEX

Table 16 shows the pattern of development in the overall process of knowledge acquisition and expert system construction for HEX. In all, HEX required 3 refinements, using the project expert for evaluation.

The knowledge acquisition process used in the construction of TRITON relied significantly upon the knowledge gathered whilst building HEX. Table 17 details the progress of knowledge acquisition with respect to TRITON. The time required to gather the necessary further knowledge to construct TRITON was significantly shorter than with HEX. This was due to the nature of the knowledge elicitation methods used in building HEX. Whilst the focus of questioning in building the HEX system concentrated on the heuristic, shallow knowledge, the explanations and justifications of such knowledge given by the expert were typically at a deep, conceptual level (ie the expert explained the paths of interaction between the elements involved). This meant much of the deep knowledge was already gathered in transcripts and summary documents. As a result of this, the knowledge acquisition sessions for TRITON tended to be highly focussed on particular areas of knowledge that were lacking or vague. It would be reasonable to suggest that, if TRITON were to be constructed from scratch, the entire knowledge acquisition process would be slightly, but not significantly, longer than in building the first-generation system, HEX.

Meeting	Month		Details
37-41	June-August '92	↓	Consideration of entities used in TRITON, suitable definitions, questions, values, etc.
41-47	September-October '92	↓	Consideration/description of how entities interact, and the level of uncertainty (conviction) in these interactions.
48-49	October '92	↓	Consideration and refinement of the working TRITON system (involving the fine-tuning of convictions).

Table 17: The Pattern of Knowledge Acquisition pertaining to TRITON

### 7.4.3 The Specification and Design of HEX and TRITON

With HEX, the design process was directed by the goal of the system, the user interface requirements, and the knowledge acquisition process. The specified

goal (ie pond assessment to determine suitability for the crested newt) guided the knowledge acquisition. The knowledge acquisition process involved a *top-down step-wise refinement* methodology (Bratko, 1990), encompassing the breaking down of the larger goal into smaller subgoals. These correspond to the attributes listed in Table 10. These subgoals were decomposed into further subgoals, until specific entities were identified that could be used as the foci of questions given to potential users. It should be noted that within this process, a number of complex relationships were identified, so that some entities were used to instantiate values for several other entities (eg heavy shading not only allows an inference about aquatic vegetation complement - via productivity - but also allows inference about high nutrient levels - via excessive leaf-fall). This approach allowed the construction of flowcharts representing the rules of interaction between goal, subgoals and further subgoals. The approach also aided in directing knowledge acquisition, helping to ensure completeness and coherency within this process.

The design of TRITON was less complex. As detailed in Chapter 6, the control of knowledge in TRITON is a 'best-first' search, and therefore the ordering of knowledge within TRITON has little effect on the ordering of questions generated by TRITON (ie it requests the most 'convincing' evidence first). Within HEX, a backward chaining (with opportunistic forward chaining) approach is used (see Section 6.2), and the ordering of knowledge has a direct effect on the order of questioning. This means that HEX requires detailed planning and design to produce a coherent mode of questioning, and facilitate future changes to the knowledge base. TRITON can be considered to generate its own questioning strategy (ie the ordering of questions) implicitly, and subsequently has a lesser design overhead. This may be seen as advantageous in terms of less development time, but there is a significant disadvantage; the knowledge engineer may have little control over the questioning structure.

#### 7.4.4 The Implementation of HEX and TRITON

The coding (ie transferal of knowledge into a machine-translatable form) of the knowledge bases of both HEX and TRITON was a relatively straightforward process, facilitated by the functionality and ease of use of the two tools upon which HEX and TRITON are constructed, respectively LEONARDO and PERSEUS. HEX was perhaps the more difficult of the two to design, as the structuring of the knowledge base had to closely follow the specifications developed during the design phase (see Section 7.4.3). HEX also required the addition of explanation features, as the rule traces given as default 'explanations' are often considered unsatisfactory for

users (eg Aikins, 1980). The explanations of HEX were constructed using a 'hypertext' framework, with a network of explanations being available to the user. Such explanations, which could be browsed and considered at leisure, and contained both practical information and ecological theory, were felt to be conducive to the educational potential of the HEX system. Whilst specification and design/layout considerations of these 'explanations' was arduous for the knowledge engineering team, the implementation of specified explanations was straightforward.

#### 7.4.5 Comparison of the Overall Construction of HEX and TRITON

The relative costs and benefits of building HEX and TRITON are briefly detailed in Table 18, and more fully discussed in this section.

As noted in Table 18, the knowledge acquisition in HEX was partially directed by the use of top-down refinement (Bratko, 1990). This gave structure and coherency in the knowledge acquisition process, and in implementation. This structure is a main contributing factor in the ease with which HEX could be amended during iterative evaluation. However, the use of this approach in design meant that the design phase was relatively protracted. Conversely, the approach to design in building TRITON was more flexible, and gave little structure to the knowledge acquisition process. The design took relatively little time (given PERSEUS as the tool of implementation).

Based on the experience of the knowledge engineer (M.Cain), the following points have been identified:

(i) The implementation of both systems was relatively straightforward, and comparable in many respects. Difficulties encountered during the implementation of HEX and TRITON frequently applied to both of these systems. A number of problems occurred that are common to expert systems generally, such as; human error (eg mixing upper and lower case when the tool used is case-sensitive), and problems with the tools at hand (eg LEONARDO became more prone to 'crashing' as the program grew). Such problems are commonly discussed in the literature (eg Jackson, 1990), and will not be considered further.

Despite straightforward implementation in both systems, it should be noted that the need to supply proper explanations in HEX, caused by the unsuitability of rule traces as explanation, was itself a major undertaking, and similar requirements for other knowledge-based system projects should be properly considered and costed.

(ii) The rigorous design required in building HEX resulted in a similarly rigorous but effective knowledge acquisition methodology. It may be that care is required to

maintain a rigorous knowledge acquisition methodology in building further systems under PERSEUS (with its less rigorous design constraints). Lack of care is likely to result in less effective knowledge acquisition.

In all, it is a reasonable conclusion that both systems would require similar effort on the part of the knowledge engineering team (with HEX being slightly easier), if both were built from scratch. However, this is only true as long as an appropriate second-generation tool, such as PERSEUS, is available. The construction of PERSEUS, however, was itself a major undertaking, and is detailed earlier in this chapter (Section 7.4.1), and in Appendix B.

Stage	HEX	TRITON
Knowledge Acquisition	Going from general to specific, using top-down refinement.	Continuing from HEX-oriented knowledge acquisition, highly focussed and intense.
Design	Highly specified, integrated with knowledge acquisition, using top-down refinement.	Less emphasis on design, more on ensuring the validity of the knowledge base.
Implementation	Fairly straightforward in both cases, due to efficient, easy to use tools (LEONARDO and PERSEUS)	
Time Scales/ Effort Required	Equivalent to about 1 year's full-time effort by knowledge engineer, seeing the expert about once per week.	Slightly longer than HEX (see 7.4.2), possibly relatively more input by the expert.
Overall Assessment	In development time, the systems would be roughly equivalent. There would need to be slightly greater effort on the part of the expert if building TRITON from scratch. HEX requires more careful design methods, whilst TRITON would require care to ensure rigour of the knowledge acquisition process.	

Table 18: Comparison of the Construction Processes of HEX and TRITON

### 7.5 Difference/Similarities in the Functionality of HEX and TRITON

The user interface of both HEX and TRITON are similar in layout and colour. They have similar functions available (explanation, quit, etc), but not necessarily using the same keys (limitations of the tools used often disallowed the

employment of the same keys for specific purposes). The questions given by both systems are the same, as is the standard pond questionnaire used with both systems (see Appendix F). The systems function at a comparable speed, and may be used on IBM-compatible PC's with the same set of specifications. Many of these similarities were due to design specifications enforced to ensure that evaluators would not be overly influenced by radically different interfaces.

The dissimilarities between HEX and TRITON are many and varied, but are relatively minor in terms of the appearance of the user interface. A major difference between the systems occurs within the explanation facility. The explanation of why a particular question is being asked is available in both systems, but the nature of explanation is different in each. The explanations in HEX have been written and added by the knowledge engineering team, whilst the explanations in TRITON are generated from the knowledge base, via the inference process. The reason for this relates to the nature of the PERSEUS architecture underlying TRITON. As detailed in Chapter 6, the knowledge used by any system built within PERSEUS is effectively 'compiled' from longer chains of reasoning. The explanation process is effectively a 'decompilation' of this knowledge. Therefore, the details given in TRITON's explanation have been actively used in the reasoning process. In HEX, however, the explanations are effectively additions that have no part in the reasoning process.

Other differences occur in terms of minor operational abilities. HEX is able to reason in two ways; it can perform a check that continues until some set of parameters clearly identify the pond as suitable or unsuitable for crested newts. In a second mode of operation, it can be forced to assess all parameters, despite the fact that it may have enough information to come to valid conclusions. HEX then presents a full listing of parameters that may render a pond suitable/unsuitable of crested newts. TRITON performs what is effectively the former reasoning process. It selectively looks for evidence that renders a pond suitable or unsuitable, starting with the most convincing evidence. When it has sufficient evidence to support some value, and no conflicting evidence *at the same level of conviction*, it presents a conclusion based on acquired evidence. The 'forced' reasoning undertaken by HEX relates to the nature of its knowledge base; it is effectively being forced to find values for each of the first-level subgoals (as listed in Table 10). TRITON has no such fixed structures (ie subgoals to be instantiated) underlying the reasoning, and 'forcing' it to ask a full range of questions will change neither its conclusions, nor its explanations.

Other differences include the handling of 'definitions' (NB: this includes definitions, descriptions, and other aids to choosing an answer to a question) within both systems. HEX gives a 'definition' alongside the list of answers to a question, whilst TRITON has a key that a user may select to be presented with a 'definition'.

TRITON has a greater functionality at the point of coming to a conclusion. A user is able, once some conclusion is reached, to change the values of entities within the system, and then direct TRITON to come to some conclusion given these new values. In this way, if a user is unsure about some value, it can be changed, and then a further conclusion is generated, using the corrected information. If this conclusion is the same as the previous one, the user may conclude that this parameter is relatively unimportant.

## 7.6 Commentary on the Development of First- and Second-Generation Knowledge-Based Systems

In this present research, it has been surmised that the overall construction of first- and second-generation systems requires effort of the same magnitude in each case, with perhaps marginally less work required in building the first-generation system. The emphasis of the steps in construction within each type of system varies, with first-generation systems requiring greater rigour in design, and second-generation systems requiring a suitably thorough approach to knowledge acquisition.

However, a number of additional factors came under consideration within this research. HEX required extra explanation facilities, as rule traces have long been identified as not well suited for explanation (Aikins, 1980). Likewise, TRITON required the construction of a specialised second-generation 'shell', PERSEUS. Both of these activities constituted major undertakings within the research, and should not be treated as trivial matters by future knowledge engineers wishing to build such systems.

## CHAPTER EIGHT

### EVALUATION OF FIRST- AND SECOND-GENERATION KNOWLEDGE-BASED SYSTEMS WITHIN THE PRESENT RESEARCH

#### 8.1 Introduction

Evaluation of knowledge-based systems is a poorly-specified, often underrated, but major part of the knowledge engineering process (Wyatt and Spiegelhalter, 1990). This chapter discusses the reasons for undertaking a formal evaluation of any knowledge-based system, and considers the parameters that may be used in an evaluation. This leads into a description of the user groups that participated in the evaluation process within this research, along with a discussion on the methods used in collecting data from these groups.

Where necessary, each of the above steps are related specifically to the first- and second-generation knowledge-based systems that are the focus of this research, HEX and TRITON respectively. It should be noted that the evaluation of HEX and TRITON involves consideration of each system as a functional tool, and the comparison of these systems as first- and second-generation systems.

The main areas addressed in this chapter are then summarised, prior to proper discussion of the analysis of the evaluation data in Chapter 9. Details of the analysis are given in Appendix A.

#### 8.2 The Requirement for Evaluation

The evaluation of a knowledge-based system may have several purposes, including; the proper testing of the system using a variety of parameters; to obtain feedback for system refinement; to establish the system is 'safe'; to determine whether the system has achieved its original specification; and to develop new evaluation methodologies (Miller and Sittig, 1990).

Overall, it is reasonable to suggest that a system that has not undergone proper evaluation may contain significant faults. Proper evaluation allows confidence in the system by the development team, and facilitates improvements to the system's operation, making it more likely to be successful in performing allotted tasks.

In the present research, the motivations for evaluation were numerous; to assess the validity of the Qualitative Conviction Calculus; to assess the acceptability and comparability of the explanations of first- and second-generation knowledge-based systems developed in this research; to properly evaluate the HEX and



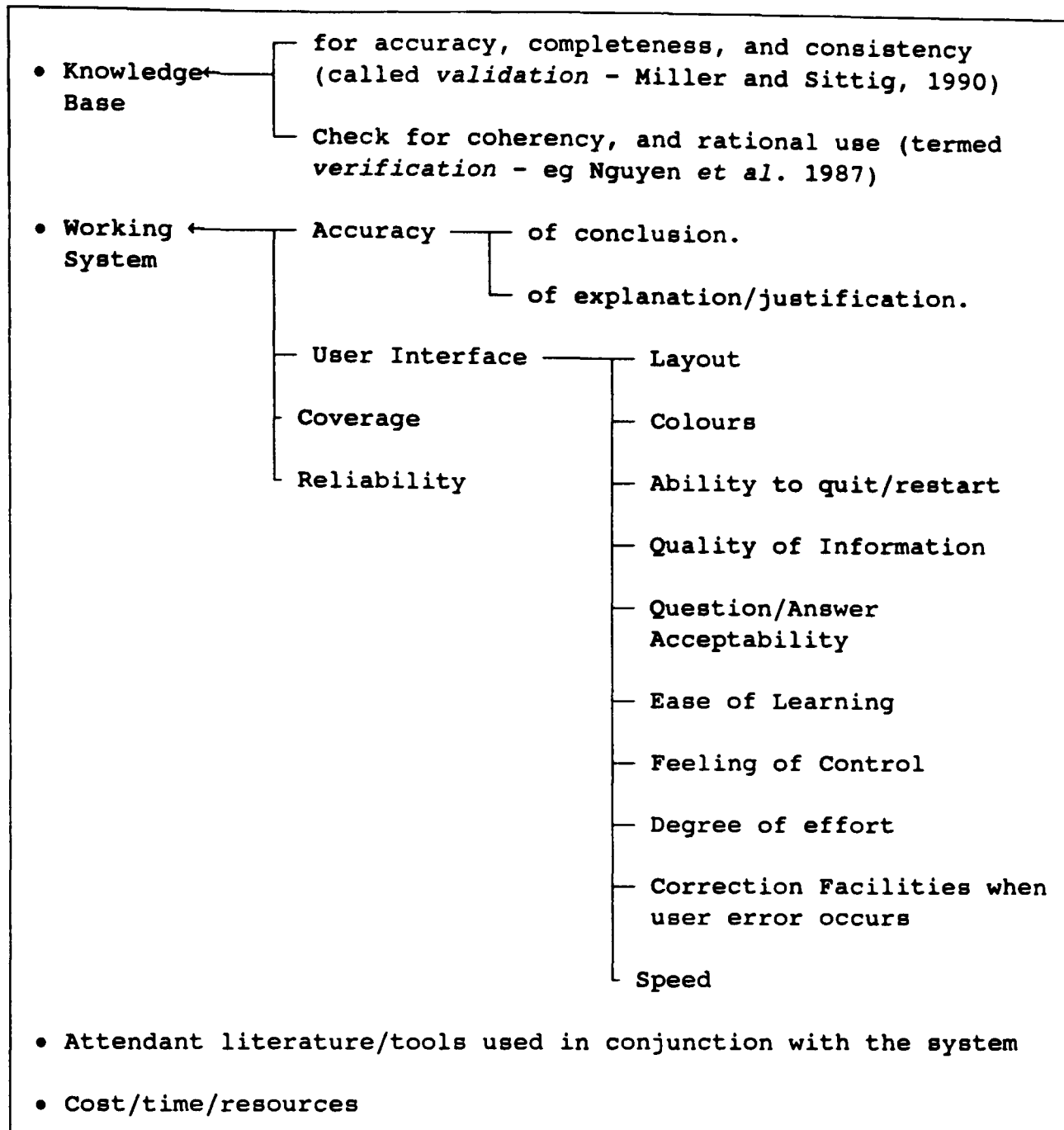
TRITON systems as working knowledge-based systems; and to compare the architectures of first- and second-generation knowledge-based systems (in the domain of habitat evaluation), in order to assess the relative benefits/problems of each architecture in terms of the knowledge engineering process and end-users.

### 8.3 The Parameters of Evaluation

The parameters of evaluation are numerous (eg Sharma and Conrath, 1992), and those commonly used are summarised in Table 19. The evaluation process may be divided into two stages; the evaluation/validation that took place during development of the HEX and TRITON systems, and; the 'final' evaluation, involving suitable user groups. This chapter will be specifically concerned with the 'final' evaluation. Details of developmental validation are readily available within the literature of knowledge-based systems (eg Preece, 1990; Schmoldt and Martin, 1989).

It is worth noting that not all of the parameters in Table 19 are necessary to address HEX and TRITON as (i) working tools, and (ii) first- and second-generation architectures, but there is significant overlap (eg 'accuracy' is of great interest for both points, whilst the 'clarity of questions' may be irrelevant to the comparison of architectures, as both systems use the same questions).

Details of these parameters, and a brief discussion on the necessity for evaluating each, are given in the following subsections.



**Table 19: The Parameters of Evaluation in an Expert System**

(abstracted from: Berry and Hart, 1990; Clegg *et al.*, 1988; Parsaye and Chignell, 1988; Sharma and Conrath, 1992)

### 8.3.1 Avoidance of Bias

It is well-documented that biases occur during evaluation of knowledge-based systems, and may be introduced by the evaluators, the experimental conditions, and the sampling choice (Fieschi, 1990). Avoidance of bias in evaluation involves the use of suitable data sets and uninvolved peers to review the system, and statistical comparisons between actual data, expert performance, and system performance (Wyatt *et al.*, 1990).

### 8.3.2 Accuracy of HEX and TRITON

Evaluation of accuracy of both final conclusions, and of explanations/justifications given during operation, is of prime importance in the evaluation of a knowledge-based system (Berry and Hart, 1990). This importance is reflected in the work carried out by the knowledge engineer and expert in system validation, prior to 'final' evaluation. Accuracy is assessed in the final evaluation work by; (i) asking peer experts to assess the accuracy of the systems, and; (ii) empirically comparing the data from 100 ponds with the conclusions of HEX, TRITON and the expert involved in development.

### 8.3.3 The User Interface of HEX and TRITON

Preece (1990) asserts that a knowledge-based system should have *user acceptance*. Berry and Hart (1990) note that the importance of the user interface has often been underestimated by past knowledge engineers, and suggest that evaluation of the user interface is fundamental to the evaluation process and success of any knowledge-based system.

The user interface encapsulates ergonomic and organisational aspects of the system, including screen layouts, colours, ease of use, amount and grain of information on the screen, layout of possible answers (eg number of choices in a menu), ability to quit and/or restart, and so on (Berry and Hart, 1990). Clegg *et al.* (1988) suggest there are a number of other issues that require deliberation when evaluating the usability of a system including; ease of learning; feeling of control in the user; degree of concentration/effort required whilst using the system; speed of system response; ease of information input/output, and; error correction facilities.

The specific user interface parameters evaluated in this research include screen layouts, ease of use, ease of learning, degree of concentration required, and overall 'usability'.

### 8.3.4 The Coverage of HEX and TRITON

Coverage (ie the ability of a knowledge-based system to handle a broad range of cases) is important in evaluating a knowledge-based system (Parsaye and Chignell, 1988). Indeed, it is often claimed that 'deep' knowledge-based systems may be able to cover a broader range of cases than an equivalent shallow system, as the degradation of ability with more and more unusual cases may be less pronounced

in deeper systems (Keravnou and Washbrook, 1989). Coverage is evaluated in this work by asking peer experts to assess the coverage of the systems involved.

### 8.3.5 The Reliability of the HEX and TRITON Software

The reliability of any software (ie its liability/resistance to 'crashing') is a critical factor in user acceptance (Kreutzer and McKenzie, 1991). Whilst there were some problems in using the LEONARDO shell during the development of HEX, and the development of TRITON under PERSEUS, the running versions of HEX and TRITON have been totally reliable to date, and have been used on a variety of IBM-compatible personal computers for several hundred runs.

### 8.3.6 Attendant Materials used in conjunction with HEX and TRITON

It is appropriate that any literature, artifacts or tools that accompany the HEX and TRITON systems be included in the evaluation process (Sharma and Conrath, 1992). This includes guide/reference texts, pond assessment forms, and other materials used with these systems.

## 8.4 Participating Groups in the Evaluation Process

Four distinct "groups" were recruited for the 'final' evaluation. These are listed in Table 20.

- |   |
|---|
| <ol style="list-style-type: none"><li>(1) The Expert upon whose knowledge HEX and TRITON were constructed. (Rob Oldham)</li><li>(2) The Knowledge Engineer (Mark Cain)</li><li>(3) A Group of individuals that are experts or near-experts in amphibian/pond ecology (termed 'peers'). (15 participants)</li><li>(4) A group of students currently undergoing the final year in an ecological degree (ie with some ecological background, but little knowledge of practical pond/habitat assessment). (20 participants)</li></ol> |
|---|

Table 20: Participating Groups within the Evaluation Process

(Numbers in each group given in brackets)

Each of these groups were used in the evaluation of different, but often overlapping, elements of the knowledge-based systems HEX and TRITON. Specific details are presented later in this chapter (Table 21; page 123), following a proper description of the methods used in evaluation.

### 8.5 Data Collection in the Evaluation Process

There are a variety of techniques available for formal evaluation of a knowledge-based system (see overview by Berry and Hart, 1990). These include; interviews; questionnaires; system walk-through (where either the expert or the knowledge engineer attempt to use the system as a 'user'); formal observation of users by the knowledge engineer; user diaries; system logging (where the system automatically records the input/output of the users); and experiments, where two or more different versions of the same system, or two different systems performing the same task, are compared. The choice of techniques is based upon a number of considerations; the nature of the task handled; the number of people in the development team; the number and type of evaluators available; access to the evaluators; and available means of storing feedback.

Within this research, evaluation involved formal experimentation to compare user responses to HEX and TRITON, using questionnaires as a means to formally obtain the views of evaluators. Questionnaire was selected as the most suitable method for use with the two larger groups of evaluators (experts and students) for the following reasons; the questionnaire format is a quick and easy way for evaluators to provide information, is a convenient means to store gathered information, and facilitates formalisation of user responses (allowing straightforward analysis of these responses).

Two distinct questionnaires were developed in the evaluation of HEX and TRITON (see Appendix G). The first questionnaire was presented to the student and 'peer expert' groups. This questionnaire contained questions requesting information on biographical details (eg address for future contact, if required, computer literacy, ecological background), material associated with the systems (eg pond assessment form, user guide), commentary on various aspects of the user interface, the justification and explanation facilities, an assessment of the educational merits of these systems, and a 'final comments' section, where users could express opinions about the systems not otherwise recorded. The version presented to the peer expert group contained additional questions that could only be realistically answered by experts. These additions included questions on accuracy of system conclusions and explanations, robustness, questioning strategy, and detected errors. The same questionnaire was used for both HEX and TRITON (with minor changes in wording,

to make the questionnaire system-specific).

The second questionnaire was filled in by the peer group alone, following the evaluation of HEX and TRITON. This questionnaire involved a comparison of example 'default' explanations as generated by HEX and TRITON. This was felt necessary, as the explanations of HEX were added by knowledge engineer and expert, whilst the explanations of TRITON were machine-generated from the knowledge base. For proper comparison of explanations generated by first- and second-generation systems, a small sample of 'default' (ie machine-generated) explanations of HEX were compared with the explanations of TRITON.

In addition, data from 100 ponds, collected by an independent source, was used to assess accuracy and comparability of the domain expert's, and systems', conclusions (see Section 8.6.1).

### 8.5.1 Evaluation Experiments

A student group of 55 individuals were initially asked to evaluate the HEX system. A smaller set of this group (20 individuals) were subsequently recruited to evaluate TRITON, after a delay of some months. This delay was imposed, as well-documented evidence exists that indicates that many users, when exposed (over a short period of time) to two alternative computer systems that perform similar tasks, will react adversely to the second system, a phenomenon sometimes referred to as the '*first is best*' effect (eg Gardner and Munroe, 1992). A delay in the order of months was felt sufficient to minimise this '*first is best*'/ordering effect. Additionally, it should be noted that the interfaces of HEX and TRITON were designed to operate in a similar way, to minimise differences in external appearance to users.

With the peer group, however, both time and geographical constraints (these peers were distributed over a wide area of Britain) dictated that such a delay was not easily implemented. Each expert looked at both systems with a short time delay between each. Typically, both systems were evaluated on the same day. To address the possibility of an order effect, some of the peers were asked to evaluate HEX first, whilst the remainder evaluated TRITON first.

In the case of the remaining two evaluators, the expert and knowledge engineer, much of the evaluation performed by these individuals took place during development, using transcribed interviews, and is detailed in Section 7.4.2. However, both the knowledge engineer and expert partook in a number of other 'final' evaluation procedures. Each of these tended to be for specific parameters (accuracy, user interface, coverage, reliability), and are identified in Table 21.

## 8.6 Evaluation Details

### 8.6.1 Assessment of Accuracy of HEX and TRITON

In the 'final' evaluation stage, the 'peer expert' group of evaluators was asked to assess the accuracies of the conclusions and explanations of the HEX and TRITON systems, using provided cases, and cases of their own experience. Such an approach to evaluating accuracy in a knowledge-based system, using domain experts uninvolved with the systems' development, is a common evaluation method, typically called *peer review* (Wyatt and Spiegelhalter, 1990).

These individuals were asked to use data from at least 4 'average' ponds (either provided, or from experience), and from at least 1 pond that had an unusual profile (from their own experience). The individuals were directed to make frequent use of the explanation facilities whilst using this data. In this way, these individuals would have some familiarity with the range of cases that could be properly addressed by these systems (ie the 'coverage' - see Section 8.3.4), in terms of 'usual' and 'unusual' ponds, prior to considering whether the systems' conclusions and explanations were accurate or otherwise.

In addition to the 'peer review', the HEX and TRITON systems and the domain expert were tested against data from 100 ponds where the presence or absence of crested newts was properly recorded. This data was gathered from an single, independent, reliable source. The data was limited to ponds from Leicestershire or surrounding counties, as this was the scope of study of the independent source. The expert was asked to state whether he considered crested newts were likely to be present or absent, give details about what were the most important parameters to influence his decision, and express a level of 'conviction' in his answer. The data was also given to the HEX and TRITON systems. This allowed comparison of the domain expert's conclusions and convictions with both reality, and the assertions of HEX and TRITON (Section 9.4).

### 8.6.2 Evaluation of the User Interfaces of HEX and TRITON

During the 'final' evaluation both student and peer groups were asked to comment (via questionnaire) on various aspects of the user interface for each system, including screen layouts, ease of use, ease of learning, degree of concentration, and overall 'usability'. The other parameters of the user interface detailed in table 19 were not specifically addressed, but these groups were asked to note any problems or observations, however minor, in the 'comments' section of their questionnaires.

### 8.6.3 Evaluation of Explanations within HEX and TRITON

Of particular relevance to the evaluation of the HEX and TRITON systems as first- and second-generation architectures was the quality of explanations. The explanations given by HEX were hand-crafted by the expert (Dr Oldham), whilst the explanations given by TRITON were derived directly from the deep knowledge within TRITON. In addition to a comparison of expert-generated explanations and the default explanations of TRITON, a separate, paper-based test was carried out that allowed proper comparison of the 'default' explanations of HEX (ie the first-generation system) and TRITON (the second-generation system).

### 8.6.4 Evaluation of Attendant Materials used in Conjunction with HEX and TRITON

The expert and knowledge engineer were specifically involved with the development of the texts that accompany HEX and TRITON, and were therefore not suitable candidates for the proper assessment of these texts. The student group, who used the pond assessment form in a field situation, were asked to comment on this form, and the user guides provided to learn HEX and TRITON. The peer (expert) group was not asked to fill in a pond assessment form in the field, only being asked to comment on the utility of the user guides.

## 8.7 Summary

Evaluation is a poorly-specified but extremely important area in developing a knowledge-based system. Presented in this chapter is a description of the evaluation processes that have been used to assess the HEX and TRITON systems, both as working "products", and in terms of first- and second-generation architectures. Analysis of the data gathered within this evaluation is given in Appendix A. Results are discussed in Chapter 9, along with specific details concerning a formal comparison of the use of first- and second-generation architectures in habitat evaluation.



Parameters of Evaluation	Participating Groups	Methods Used
Accuracy - of conclusions - of explanations	Expert (E); Knowledge Engineer (K); Peers (P); Students (S)	Machine-based evaluation; Peer group assess accuracy; Independent data used to compare expert/systems' conclusions against real data
User Interface (all aspects)	E;K;P;S	Machine-based evaluation, with comments.
Coverage	E;K;P;S	Machine-based evaluation; Independent data used to compare expert/systems' conclusions against real data
Reliability	K	Monitoring
Attendant Artifacts	S	Questionnaire and comments

Table 21: Summary of Evaluation Details

## CHAPTER NINE

### ANALYSIS OF EVALUATION DATA

#### 9.1 Introduction

This chapter considers the results of the analysis of the data gathered during the evaluation process. This examination will focus on both the overall trends in the data, and particular findings that are significant. Full details of statistical analyses are given in Appendix A.

The evaluation study can be divided into two main strands; the assessment of the first- and second-generation knowledge-based systems developed within this research by different groups of evaluators, and; the comparison of actual pond data against the conclusions of a domain expert, and the conclusions of the first- and second-generation systems.

#### 9.2 Comparison of User Opinions

One strand of the evaluation involved groups of student and experts as evaluators of the first-generation and second-generation knowledge-based systems developed in this research (HEX and TRITON respectively). 15 experts evaluated both HEX and TRITON, whilst 55 students evaluated HEX, and 20 of these went on to evaluate TRITON. Recording the conclusions of these two groups was via appropriate questionnaires (see Appendix G). The areas under investigation were not suitable for both evaluation groups to address, as some areas required specialist knowledge. Each evaluator filled in a questionnaire for both HEX and TRITON separately. A number of parameters were addressed within these questionnaires, and these can be classed into three main groups; user interface; reasoning/questioning strategy, and; conclusions and explanations. As well as the questionnaire given to both groups, the experts were asked to comment on examples of 'default' explanations of the HEX and TRITON systems.

Since the same student and expert users provided opinions of both HEX and TRITON, it is possible to compare individual user's opinions of the systems for the individual questions that relate to the three aspects of interest; user interface, reasoning/questioning strategy, and conclusions/explanations. That is, for each question it is possible to state whether an individual rates HEX or TRITON more highly, or considers them equal. The resulting within-person comparison can then be analysed using binomial probabilities (eg Agresti, 1991). Full details are given in appendix A.

In addition, a rating score has been generated for each user for the user interface, and the overall system, by summation of numeric values attached to individual answers (the most favourable answer having the highest numeric score, the least favourable having a score of 1). These rating scores generated for both systems can then be associated with individual users, and used in comparing the HEX and TRITON systems in terms of the user interface, and overall system rating. This data has been analysed using nonparametric procedures as detailed in Appendix A, part 2.

### 9.3 Questionnaire-Based Evaluation of User Opinions

In this section, the three main groups of parameters - user interface, reasoning/questioning strategy, and conclusions/explanations - will be discussed in turn. The overall rating of HEX and TRITON will also be discussed. In considering these groups, the general trend underlying each grouping will be summarised, and exceptional findings will be clearly identified. However, it is of note that for many of the individual parameters considered here, there was no significant difference between;

- (i) Individual's rating of HEX compared to TRITON,
- (ii) Opinions of the student and expert groups (in terms of ratio of numbers showing a higher rating for HEX relative to numbers showing a higher rating for TRITON),
- (iii) Ratings of the experts, when considering the order of presentation of the systems.

Full details of statistical analyses are given in Appendix A (page 2). In the following tests and tables, the  $p$ -value generated in testing the null hypothesis ( $H_0$ ) of no difference is presented.

Significant differences were observed for some criteria; for example, there were differences in the user views of the default explanations generated by the first and second generation systems HEX and TRITON respectively.

#### 9.3.1 User Interface

Table 22 summarises the user interface data. In general, both groups of evaluators reacted favourably to the systems, with at least 86% of evaluators finding both systems 'definitely' or 'mostly' acceptable for the first 3 questions. The answers given in Table 22 were combined for each user by attaching numeric values to answers, so that 'Definitely' rated 5, 'Mostly' rated 4, and so on - with the

reverse happening for 22(d), as 'Definitely' here indicated an unfavourable answer - and the values for each user were summed. The user's two sums for each system were then compared, to see if each user rated HEX or TRITON more highly. There was no significant difference between the proportion of both students and experts showing a higher rating score for HEX and TRITON (Table 23). This lack of significant difference between HEX and TRITON for both systems may be partly attributed to the efforts of the knowledge engineer to make the interfaces as uniform as possible for HEX and TRITON, so as not to unduly affect other aspects of the evaluation.

One aspect of the user interface addressed as a second-generation issue within this research is the ability of a system to allow revocation or changing of values during run-time and at conclusion. Whilst systems built under PERSEUS may have both types of revocation/change, the ability to revoke/change values was only permitted at the conclusion of an assessment in TRITON. This was done so that the abilities of both systems were not too dissimilar.

The ability to revoke/change answers at conclusion within TRITON was commented upon by a number of users. Most of these (unprompted) comments were favourable (1 expert and 3 students noted explicitly that they liked it, though a greater number stated verbally that it was a useful facility; 1 expert explicitly stated that he found it confusing). Overall, it was a feature that attracted the considerable commentary from those evaluators looking at TRITON.

(A) Do you feel that the various screen layouts you encountered were acceptable?

Answers	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	37	13	7	6
Mostly	17	7	7	7
For a reasonable part	1	0	1	2
Only some of the time	0	0	0	0
Not at all	0	0	0	0

(B) Do you feel it was easy to examine the information within HEX/TRITON?

Answers	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	38	16	10	7
Mostly	16	3	5	7
For a reasonable part	1	0	0	1
Only some of the time	0	0	0	0
Not at all (Unmarked)	0	0 (1)	0	0

(C) Was the HEX/TRITON system easy to learn?

Answers	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	45	15	13	11
Mostly	8	4	2	3
For a reasonable part	2	0	0	1
Only some of the time	0	0	0	0
Not at all (Unmarked)	0	0 (1)	0	0

(D) Did HEX/TRITON require a high degree of concentration to use?

Answers	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	2	0	1	1
Mostly	12	4	4	6
For a reasonable part	23	8	6	5
Only some of the time	12	6	1	1
Not at all	6	2	0	0

Table 22: Summary of User Interface Data

	<i>p</i> -value: $H_0$ (no difference)
(A) HEX vs TRITON	
Students (N = 20)	0.623
Experts (N = 15)	0.274
Combined (N = 35)	0.332
(B) Comparison of Student/ Expert Ratings	0.283
(C) Ordering of Presentation to Experts (N = 15)	0.255

Table 23: Comparison of Ratings Derived from User Interface 'Scores'

### 9.3.2 Conclusions and Explanations

In the following subsections, a number of aspects relating to the conclusions and explanations of HEX and TRITON are discussed, including accuracy, clarity, utility to users, and educational merit. There was found to be no significant difference between the rating of HEX compared to TRITON for these and other criteria, in all but two cases. Firstly, expert evaluators rated the utility of information (ie 'usefulness') given by HEX higher than that of TRITON. Secondly, there is a significantly higher rating of default explanations of TRITON by expert evaluators, relative to the 'default' explanations (ie rule traces) of HEX.

#### **Accuracy of Conclusions and Convictions**

The accuracies of HEX and TRITON conclusions were assessed by asking the expert evaluators to state whether they considered each system was acceptably accurate in its conclusions. There was no significant difference between the rating of HEX and TRITON (see Table 24). In considering the convictions attached to the conclusions of TRITON, a substantial proportion (12 of 15) of experts felt that the conviction was either acceptably accurate, or acceptably accurate with few exceptions (see Table 25).

(A) Do you think that the assessment of whether the pond was suitable or unsuitable was acceptably accurate?

Answers	Experts (N=15)	
	HEX	TRITON
Yes	5	7
Yes, with notable exceptions	9	7
No	1	1

(B) Comparison of Ratings for Data in (A)

	p-value: $H_0$ (no diff)
(1) HEX vs TRITON (N = 15)	0.500
(2) Ordering of Presentation to Experts (N = 15)	0.429

Table 24: Data concerning the Accuracy of Conclusions

Do you think the strength of conviction that the system expressed in the conclusion (eg 'strong evidence for..', 'some evidence for..' etc) was correct:

	Experts (N=15)
Yes	9
Yes, with few exceptions	3
No, overconfident in conviction	1
No, too weak in conviction	1
No, too strong and too weak at different times	1
Don't know	0

Table 25: Data of Experts' Opinions of Accuracy of TRITON-generated Convictions

### Explanation Facilities

In considering the explanation facilities of HEX and TRITON given during consultation, both user groups found the justifications clear and understandable (Table 26). There was no significant difference between the rating of HEX or TRITON by the student and expert evaluation groups.

A) Did you find the justifications of questions (The 'Explain?' facility), when you used them, were clear and understandable?

Answers	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	35	9	7	5
Mostly	19	9	5	7
For a reasonable part	1	1	3	3
Only some of the time	0	1	0	0
Not at all	0	0	0	0

(B) Comparison of Ratings for Data in (A)

	p-value: $H_0$ (no diff)
(1) HEX vs TRITON	
Students (N = 20)	0.172
Experts (N = 15)	0.500
Combined (N = 35)	0.180
(2) Comparison of Student/ Expert Ratings	0.300
(3) Ordering of Presentation to Experts (N = 15)	0.039

Table 26: Results of Data of Explanations given during Consultation

A significant difference occurred in considering the effect of the order in which systems were presented to expert users (Table 26(B.3)). There was a significant difference in expert evaluators' opinions relative to the order in which they examined the systems. The source data (Appendix A, page A-6, question 4a) indicates that, for the parameter of explanation, the expert users tended to favour the second system that was presented to them. This is difficult to explain. The relationship is not highly significant, and none of the other (many) parameters showed an ordering effect.

Experts were also asked to comment on the validity of the explanations given with the systems' final conclusions (Table 27). Again, no significant difference was observed between HEX and TRITON.



(A) Do you think the explanations of the final conclusions were correct?		
Answers	Experts (N = 15)	
	HEX	TRITON
Yes	9	7
Yes, with notable exceptions	6	8
No	0	0
Didn't understand them	0	0
Don't know	0	0

(B) Comparison of Ratings for Data in (A)		p-value: $H_0$ (no diff)
(1) HEX vs TRITON	(N = 15)	0.623
(2) Ordering of Presentation to Experts	(N = 15)	0.417

Table 27: Results of Data of Explanations given at Conclusion

Both sets of evaluators found the explanations of both HEX and TRITON generally coherent. Analysis indicates that the machine-generated explanations of TRITON are comparable to HEX's hand-built explanations (in terms of end-user acceptability), with a reasonably favourable level of acceptance by evaluators. However, in considering the 'default' explanations of a first-generation system relative to the default explanations of a second-generation system (using 3 examples), the results were often highly significant, with the evaluators very strongly favouring TRITON. This was found to be the case in considering the parameters of comprehensibility, clarity, utility, and overall user 'preference'. Table 28 gives examples of this trend, summarising the data for comprehensibility and preference. For the remaining parameters of accuracy and sufficiency of detail for educational and practical users, most of the results were significant, and all had a small  $p$ -value ( $< 0.09$ ). These again favoured TRITON in all cases. The default explanations data is fully presented in Appendix A (part 3).

These results are a strong indication that the architecture underlying the second-generation system TRITON is capable of better default explanation of reasoning than the architecture of HEX. The implications of this are discussed in Section 9.5.1, and in Chapter 10.

**Comprehensibility:**

(A) From the details given in the explanations, do you understand how the focus of each example relates to the pond's suitability to support a viable crested newt colony are related (N=15)?

Answers	Example 1		Example 2		Example 3	
	HEX	TRITON	HEX	TRITON	HEX	TRITON
Definitely	0	10	0	10	0	9
Mostly	2	4	5	4	4	5
For a reasonable part	4	1	3	0	2	1
Only some of the time	4	0	1	1	5	0
Not at all	5	0	6	0	4	0

(B) Comparison of Ratings for Data in (A) - HEX vs. TRITON

	p-value: $H_0$ (no diff)
Example 1	0.000031
Example 2	0.00092
Example 3	0.000061

**Preference:**

(A) Which explanation do you prefer?

	HEX	Neither	TRITON
Example 1	1	2	12
Example 2	1	0	14
Example 3	1	1	13

(B) Comparison of Ratings for Data in (A) - HEX vs. TRITON

	p-value: $H_0$ (no diff)
Example 1	0.0017
Example 2	0.00049
Example 3	0.00091

Table 28: Results of Data of 'Default' Explanations - Criteria of Comprehensibility, and Preference

## Utility of Information

A summary of the data concerning the utility of the systems' information is presented in Table 29. Note that systems' information was made up mainly of conclusions and explanations (with some minor information available from 'definitions').

A) Do you feel that the overall system gave useful information?				
Answers	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	24	7	6	3
Mostly	23	12	7	8
For a reasonable part	7	1	2	3
Only some of the time	1	0	0	1
Not at all	0	0	0	0

(B) Comparison of Ratings for Data in (A)		p-value: $H_0$ (no diff)
(1) HEX vs TRITON		
Students (N = 20)		0.377
Experts (N = 15)		0.035
Combined (N = 35)		0.048
(2) Comparison of Student/Expert Ratings		0.196
(3) Ordering of Presentation to Experts (N = 15)		0.750

Table 29: Data Summary concerning the Utility of Information given

In terms of the utility of information, the experts showed a significantly higher rating of HEX. This is reasonable, as the HEX statements were hand-built by the knowledge engineer and domain expert prior to evaluation. Conversely, TRITON explanations are generated by machine, and tend to appear as poorly-stated English. For this reason, it would have been reasonable to expect all evaluators to rate HEX explanations more highly than TRITON explanations - the student group alone did not show such a favouring.

## Educational Merit

In requesting a response for 'educational merit' for HEX and TRITON, no significant difference between these systems was observed for both students and experts (see Table 30).

A) Do you feel you have gained any insight into the pond ecosystem whilst using HEX/TRITON?				
Answers	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	16	8	5	3
Mostly	13	6	2	3
For a reasonable part	19	4	2	2
Only some of the time	7	3	3	5
Not at all	0	0	2	2

(B) Comparison of Ratings for Data in (A)		p-value: $H_0$ (no diff)
(1) HEX vs TRITON		
Students (N = 20)		0.632
Experts (N = 15)		0.313
Combined (N = 35)		0.395
(2) Comparison of Student/Expert Ratings		0.336
(3) Ordering of Presentation to Experts (N = 15)		0.250

Table 30: Data concerning the Education Merit of HEX and TRITON

### 9.3.3 Reasoning/Questioning Strategy

The reasoning strategy of each system was commented upon by expert evaluators using two criteria; the 'robustness' of each system, and the questioning strategy of each system. Expert evaluators showed a significantly higher rating of the questioning strategy of HEX, and a noticeably (but marginally non-significant) higher rating for the 'robustness' of TRITON.

## Robustness

The expert evaluators were asked to comment on the robustness of both systems (ie ability to handle a wide variety of cases). A summary of responses is given in Table 31.

(A) Do you feel that the system is able to handle a wide variation of cases?		
Answers	Experts (N = 15)	
	HEX	TRITON
Yes	6	10
Yes, with notable exceptions	6	3
No	2	1
Don't know	1	1

(B) Comparison of Ratings for Data in (A)		p-value: $H_0$ (no diff)
(1) HEX vs TRITON (Experts) (N = 15)		0.063
(2) Ordering of Presentation to Experts (N = 15)		0.143

Table 31: Results of Data concerning Robustness of HEX and TRITON

Investigation of the data by statistical analysis indicated a marginally non-significant difference between the two systems, with a noticeable favouring for TRITON. It is noteworthy that several of the experts stated they could detect an underlying logic to the reasoning strategy used by TRITON (mainly from examination of the highly-structured explanations and justifications), and this gave them the sense that TRITON was a more logical, and therefore more robust system. In actuality, results discussed in Section 9.4 indicate that TRITON is comparable in its range of accuracy to HEX (to properly test this, a set of ponds from many different locations collected in a uniform way would have to be used in assessment - such a data set was not available at the time of evaluation). It may be that the logic underlying TRITON's reasoning is more transparent than that of HEX, and gives users a better sense of how the reasoning is occurring. This may give users greater confidence in TRITON's capabilities, lending TRITON a sense of 'robustness'.

## Questioning Strategy

The expert evaluators were asked to comment on the questioning strategy (ie the acceptability of the order in which questions were asked). The details of responses are given in Table 32.

(A) Do you feel that the overall ordering of questions is acceptable for gathering information for the evaluation of a pond in terms of suitability for the crested newt?

Answers	Experts (N = 15)	
	HEX	TRITON
Yes	13	8
Yes, with notable exceptions	2	5
No	0	2
Don't know	0	0

(B) Comparison of Ratings for Data in (A)

	p-value: $H_0$ (no diff)
(1) HEX vs TRITON (N = 15)	0.0352
(2) Ordering of Presentation to Experts (N = 15)	0.875

**Table 32: Results of Data concerning the Questioning Strategies of HEX and TRITON**

There was found to be a significant difference between answers concerning HEX and TRITON, with HEX being favoured. Many of the experts said they felt more comfortable with the HEX questioning strategy, which was consistent and reasonable. Many were less comfortable with the questioning strategy of TRITON. Most frequently, the experts were unhappy with the fact that TRITON asked very few questions (on average) relative to HEX. In some cases, TRITON came to a decision with a minimum of three questions, and often came to a conclusion given the answers to five or six questions. HEX, on the other hand, tends to ask a substantially larger number of questions (see Table 36). This seemed to give users the impression that HEX was considering more factors. Often, however, HEX's decisions were ultimately based on the same criteria as TRITON's. Additionally, the accuracy of both systems appears to be comparable (see Section 9.4).

#### 9.3.4 Overall Ratings of HEX and TRITON

The lack of significant difference between HEX and TRITON in terms of a number of parameters is confirmed when considering the data as a whole. Figure 6 shows the distribution of rating scores assigned to each system by student and expert evaluators. For students, with a potential range of (12-60), it should be noted that all fall well above the scale midpoint (36) for both HEX and TRITON. A similar pattern occurred in the expert ratings of HEX and TRITON (midpoint of 52). This may be taken as an indication that the overall reaction to both HEX and TRITON of both groups of evaluators is favourable.

Table 33 shows the differences between overall rating scores of students and experts of HEX and TRITON. There is no significant difference between both student ratings for each system, expert ratings for each system, and between student and expert ratings of HEX (when only questions answered by both groups are considered). However, Table 33(d) shows there is a significant difference between expert and student ratings of TRITON.

These findings indicate that the student and expert evaluation groups found little to distinguish the HEX and TRITON systems overall. The lower rating of TRITON by experts relative to student ratings may indicate that the experts found this system slightly less satisfactory than would a novice or non-expert, but it may also be attributable to the inherent conservatism of judgement found in experienced individuals.

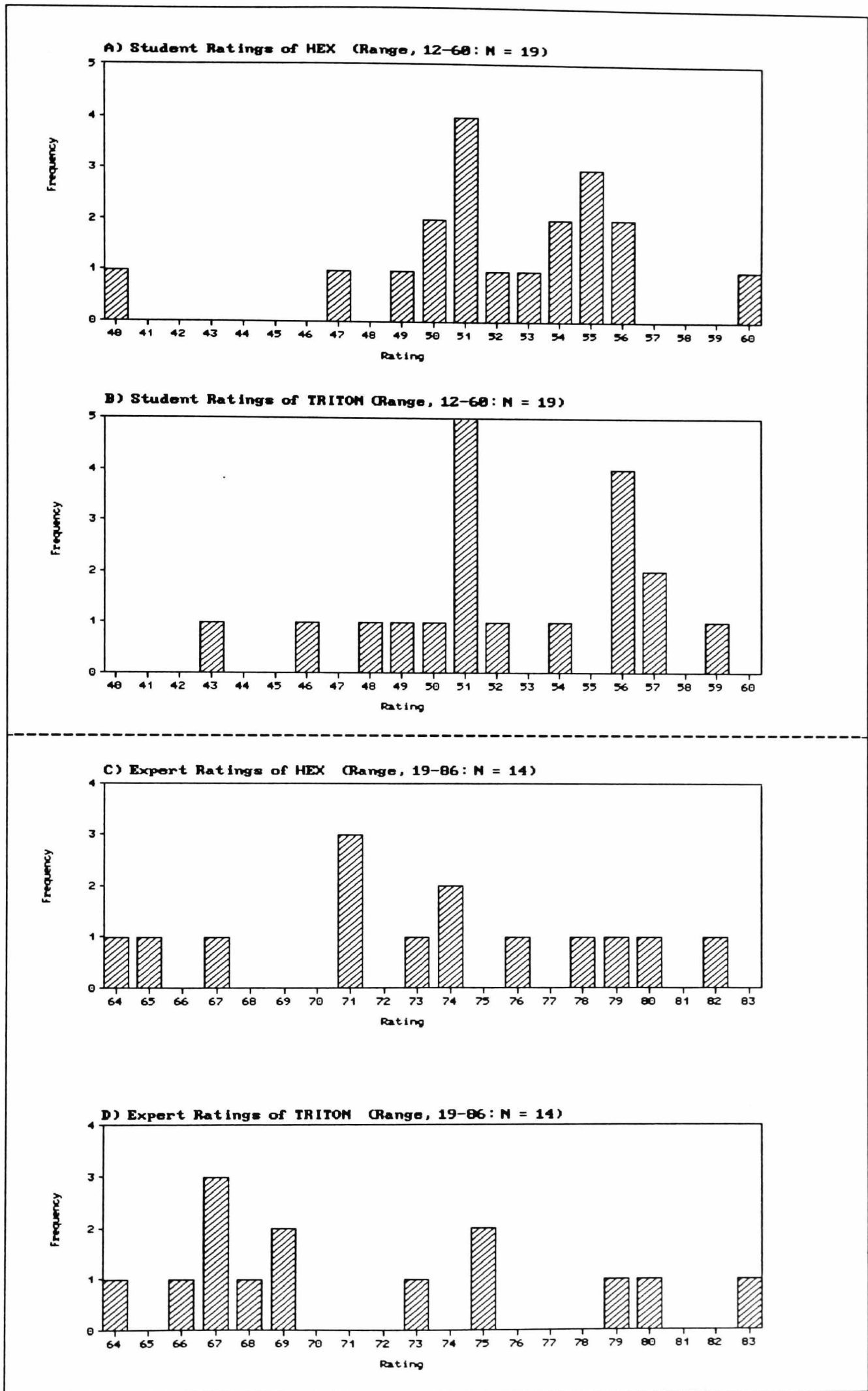


Figure 6: Overall Rating of Student and Expert Evaluators for HEX and TRITON



A) Comparison between HEX and TRITON - Students

	N	MEDIAN	p-value: $H_0$ (no diff)
HEX data	19	52.000	> 0.5
TRITON data	19	51.000	

B) Comparison between HEX and TRITON - Experts

	N	MEDIAN	p-value: $H_0$ (no diff)
HEX data	19	73.500	> 0.25
TRITON data	19	69.000	

C) Comparison of Student and Expert Evaluations of HEX

	N	MEDIAN	p-value: $H_0$ (no diff)
Student data	55	51.000	> 0.4
Expert data	14	50.000	

D) Comparison of Student and Expert Evaluations of TRITON

	N	MEDIAN	p-value: $H_0$ (no diff)
Student data	19	51.000	0.05 > $p$ > 0.01
Expert data	15	47.000	

Table 33: Overall Rating of HEX and TRITON by Student and Expert Evaluators

9.4 Comparison of Real Pond Data and the Conclusions of the Domain Expert, HEX and TRITON

The second strand of the evaluation process involved the comparison of actual data from 100 ponds (with recording of crested newt presence/absence)

against the conclusions of likelihood of pond suitability to support crested newts, as given by the domain expert (Dr Rob Oldham), HEX and TRITON. Additionally, the domain expert was asked to attach a level of conviction to his conclusions, which was later compared to TRITON's convictions in its conclusions. A number of analyses were undertaken on this data:

- (i) The actual presence/absence of newts was compared to the assertions of the domain expert, HEX and TRITON.
- (ii) The conclusions of the expert, HEX and TRITON were compared to each other for each pond.
- (iii) The conclusions/convictions of the domain expert and TRITON were compared.

The overall findings of this analysis indicate that there is reasonable agreement between (i) the presence/absence of crested newts in a pond, and the conclusions of suitability/unsuitability from the domain expert, HEX and TRITON, and (ii) the conclusions of domain expert, HEX and TRITON with each other. Table 34 summarises the findings of these 6 comparisons.

Compared Values	Percentage of sites showing agreement (N = 100)	Kappa-value, indicating 'strength' of relationship
1) Crested newt presence/absence vs domain expert's conclusions about suitability/unsuitability	72%	0.358
2) Crested newt presence/absence vs HEX's conclusions about suitability/unsuitability	71%	0.322
3) Crested newt presence/absence vs TRITON's conclusions about suitability/unsuitability	78.8%	0.526
4) Domain expert's conclusions about suitability/unsuitability vs HEX's conclusions about suitability/unsuitability	87%	0.550
5) Domain expert's conclusions about suitability/unsuitability vs TRITON's conclusions about suitability/unsuitability	81.8%	0.426
6) HEX's conclusions about suitability/unsuitability vs TRITON's conclusions about suitability/unsuitability	88.9%	0.643

Table 34: Summary of Data concerning the Comparison of Real Data with the Conclusions of the Domain Expert, HEX, and TRITON

The data indicates similar levels of agreement between actual sites and the conclusions of domain expert, HEX, and TRITON, with marginally better agreement between actual sites and TRITON. It is apparent, however, that the expert, HEX and TRITON make a significant number of mistakes in their assessment. In a few cases, the expert and both systems conclude a pond is suitable, but no crested newts have been identified at the pond (this occurs in 1-3% of cases). However, this is not a serious mistake, as there are easily explained circumstances where a pond is 'suitable' for newts, but may still not have crested newts present (eg the pond may be remote from "recruitment" sites). A more serious error occurs when the expert or systems conclude that a pond is unsuitable for crested newts, but there is in fact crested newts present at the pond. This occurs in 20-26% of the cases in the sample of ponds used, for the expert and both systems. It should be noted, however, that the judgements made by the expert and HEX and TRITON systems relate to the suitability of a pond to support a *viable* population of crested newts. This has been specified by the domain expert as a reproducing and sustained population of crested newts, containing at least 100 adults. The data used for the real ponds had no information concerning numbers of individuals counted, and it is a common occurrence that small numbers of adults can persist for several years in unsuitable sites, and often eggs and even larvae are found at such sites. For this reason, and in light of the results gathered from the expert evaluators (93.3% considered both HEX and TRITON reasonably accurate in its conclusions), it may be acceptable to consider the number of errors indicated by the comparison with 100 actual sites as being inflated.

Table 34 also shows that the 'strengths of agreement' between the conclusions of the human expert, HEX, and TRITON is relatively high (81.8-88.9%). In addition, the source data showed the expert and both systems come to same conclusions in a large number of cases (in 78% of cases). This suggests two conclusions;

(i) Both systems contain a reasonably accurate embodiment of the domain expert's knowledge.

(ii) The first- and second-generation systems, using different methods of control, different levels of knowledge, and different representation structures, come to the same conclusions, in a large majority of cases.

Of further interest is the comparison of the domain expert's conclusions and statements of certainty with those of TRITON. TRITON can express 9 possible conclusions about whether a pond is suitable for crested newts; 4 levels of certainty that a pond is suitable, 4 levels of certainty in a pond being unsuitable, and a summary that no conclusion may be drawn from given data. The 4 levels of certainty

expressed relate to a final *conviction* in the conclusion in the following way;

- 1) 'extremely convinced' in the conclusion when the conviction is 'in all cases', or 'in virtually all cases',
- 2) 'strongly convinced' in the conclusion when the conviction is 'in most cases' or 'in many cases',
- 3) 'reasonably convinced' in the conclusion when the conviction is 'in about half of the cases' and 'in some cases',
- 4) 'possibly (true)' in the conclusion when the conviction is 'in few cases', or 'in very few cases'.

The expert was asked to conclude from real data of 100 different ponds, using these 4 'certainties', or a 'no conclusions drawn', when coming to conclusions of suitability/unsuitability for each pond. Table 35 displays the relationship between the conclusions and attached certainties of the domain expert and those of TRITON. In most cases, TRITON gives an equal or more conservative conclusion (ie with lower 'certainty') compared to that of the domain expert. This tallies in part with the statements of the expert evaluators; 12 out of 15 felt the levels of certainty expressed in the final conclusions of TRITON were correct, with no or few exceptions. Where the domain expert and TRITON disagree, in their conclusions, it is often the case that either both, or just TRITON, appear unassured in the certainty of its conclusions (ie 'possibly true').

These findings suggest that the statements generated within the Qualitative Conviction Calculus (as implemented within PERSEUS) may be a potentially close (but not exact) match to the way that the domain expert reasons under qualitative uncertainty, and a close match to the way qualitative uncertainty is interpreted by other experts in this domain.

<u>Suitability of Pond</u>	<u>Level of Certainty</u>	<u>Code</u>
Suitable,	extremely convinced	SE
Suitable,	strongly convinced	SS
Suitable,	reasonably convinced	SR
Suitable,	possibly	SP
-	No conclusions drawn	N
Unsuitable,	possibly	UP
Unsuitable,	reasonably convinced	UR
Unsuitable,	strongly convinced	US
Unsuitable,	Extremely convinced	UE

Expert	TRITON Conclusions									Tot.
	UE	US	UR	UP	N	SP	SR	SS	SE	
UE	-	4	9	2	-	-	-	-	-	21
US	-	5	10	6	-	-	-	-	-	21
UR	-	2	14	8	1	2	-	-	-	27
UP	-	1	5	5	-	8	-	-	-	19
N	-	-	-	-	-	2	-	-	-	2
SP	-	-	-	5	-	5	-	-	-	10
SR	-	-	-	3	-	-	-	-	-	3
SS	-	-	-	-	1	2	-	-	-	3
SE	-	-	-	-	-	-	-	-	-	0
Tot.	0	12	38	29	2	19	0	0	0	100

Table 35: A Comparison of Conclusions/Certainties of the Domain Expert and TRITON

### 9.5 Commentary on the Findings of the Evaluation

Within this section, each of the second-generation issues addressed within the research (and introduced in Chapter 3) will be discussed in terms of the findings of the analysis of evaluation data. These are depth (and compilation), uncertainty, control, and user interface issues respectively.

### 9.5.1 Depth

This research has involved the construction of deep knowledge-based systems containing a finer grain of knowledge than equivalent shallow systems. The research has shown that, by using compilation processes, these deep systems can reason at a comparable speed to, and generate results comparable to, an equivalent shallow system. These compilation processes offer a two-fold benefit; they allow the efficient use of deep knowledge, and they allow access to deep knowledge, which can be used in explanation facilities.

A further finding of this research has been that it is possible to use deep knowledge to generate default explanations that are significantly more acceptable to users than the 'default' explanations of shallow systems. In addition, the 'default' explanations of the deep system may be regarded, in terms of the application systems composed in this study, as equivalent to expert-generated (hand-built) explanations. A benefit of having more acceptable default explanations, as in TRITON, is that maintenance of such explanations occurs automatically when the knowledge base is altered.

### 9.5.2 Uncertainty

The level of success achieved by the Qualitative Conviction Calculus is demonstrated in Table 35, and by considering expert evaluators' choices when considering the conclusions of TRITON. Table 35 indicates that, in 29% of cases, TRITON agrees exactly with the expert, and in a further 43% of cases, expressed the same conclusions as the expert, but with a more conservative expression of certainty. From these results, it may be reasonable to suggest that the QCC deserves further study as a potential means to model the processing of qualitative uncertainty in the human mind.

### 9.5.3 Control

The 'best-first' approach to control, with the focus of search falling upon the convictions of compiled terms, proved to be a significantly faster in coming to conclusions than the backward chaining method (with opportunistic forward chaining) embodied within HEX, whilst being of similar accuracy. Table 36 shows the average time and number of questions asked by each system, when used to evaluate 100 actual sites.

N = 100	Mean	Median	Maximum	Minimum
Number of Questions-HEX	20.35	21	33	2
Number of Questions-TRITON	7.6	7	15	3
Time (seconds) - HEX	103.29	101.5	150	66
Time (seconds) - TRITON	58.06	55	125	20

Table 36: Number of Questions Asked, and Time Taken, by HEX and TRITON when used to Evaluate 100 actual pond sites

As discussed in 7.4.3, the PERSEUS approach allows the knowledge engineer to be relatively unconcerned with the order of knowledge, but means the knowledge engineer has little control over the questioning strategy. A result of this is illustrated in the response of expert evaluators when asked about the questioning strategy. These evaluators showed a significant favouring of the questioning strategy of HEX, and many expressed misgivings about the small number of questions asked by TRITON relative to the number asked by HEX.

#### 9.5.4 User Interface

A common criticism of first-generation systems is an inability to revoke values during run-time and at the point of conclusion. Systems constructed within the PERSEUS shell may have both these facilities. In TRITON, the ability to change values at the point of conclusion was allowed. Whilst evaluators were not specifically asked for comments on this ability, a number of them did make verbal comments upon it, most being favourable. Some of the evaluators specifically noted in the 'final comments' section of the evaluation questionnaire that they thought this facility a useful one (1 expert from 15, 3 students from 20 praised this facility without prompting; 1 expert found it confusing).

## 9.6 Summary

The main criticism of the TRITON system is in its questioning strategy; users showed a significantly higher rating of HEX in terms of this parameter.

Users showed a significantly higher rating of the 'default' explanations of TRITON relative to the 'default' explanations of HEX via a number of parameters. Additionally, users showed little difference in the rating of hand-built explanations of HEX and the 'default' explanations of TRITON. This indicates that deep knowledge, when suitably manipulated and represented, addresses and answers a common criticism of shallow systems; that the 'default' explanation facilities (ie rule traces) are often not meaningful to users.

The conclusions of the domain expert and TRITON about pond suitability for crested newts given data from 100 ponds, in terms of conclusions with attached convictions, showed a noticeable similarity. TRITON tended to express the same, or a more conservative, conviction as the domain expert. The comments of expert evaluators about the acceptability of TRITON's convictions supported this findings, with a majority of the experts expressing confidence in TRITON's statements of conviction.

Overall, the analysis of the evaluation indicates that, for many parameters, evaluators show little favouring of either HEX or TRITON, and often find both equally acceptable. This indicates that these two systems, from the perspective of an end-user, operate in a similar manner.



## CHAPTER TEN: DISCUSSION

### 10.1 Introduction

This chapter discusses the overall conclusions of this research. The aims of the present research are first recapped, followed by a brief description of the methods employed to address these aims. The conclusions of this work are then presented, and this leads into a discussion of the implications of these conclusions.

### 10.2 Aims of the Research

The aims of the present research have been numerous, but interrelated. This research has involved the utilisation of the domain of ecological habitat evaluation as a test-bed for a variety of research issues, relating to 'second-generation' knowledge-based systems. These include the exploration of issues relating to deep knowledge, uncertainty, control, and the user interface, all with specific reference to ecology.

The exploration of ecological deep knowledge is a particularly important strand in this research. By considering existing work in deep knowledge, and how human ecologists reason using deep knowledge, it was a goal of this research to examine the problems of knowledge representation and reasoning in deep ecological knowledge-based systems. A further goal was to consider how such systems could run at a speed and efficiency comparable to first-generation systems.

The implementation of a proper means to address uncertainty relative to ecological deep knowledge is another important issue. The aim was to consider existing methods of implementing uncertainty in knowledge-based systems, and use or adapt any existing methods that suitably represent uncertainty in ecological deep knowledge. If these existing methods of reasoning under uncertainty could not be used or modified appropriately, novel methods were to be developed.

Control is a further issue that required examination. Given developments in the representation of ecological depth and uncertainty, it was necessary to consider existing methods of controlling (ie ordering the use of) knowledge, and to see if any were acceptable or appropriate to use with the deep knowledge representations developed. If existing methods were not suitable, then novel methods would be required.

In building the second-generation knowledge-based system, inclusion of certain user interface/system abilities often associated with second-generation systems was required. Examples of these include revocation of values for objects

within the system.

A further aim of the present research was to construct, use, and compare first- and second-generation knowledge-based systems via a number of criteria, including issues of construction/development, and impact on potential end users. In doing so, this research aimed to explore the pros and cons of constructing and using first- and second-generation knowledge-based systems in habitat evaluation.

### 10.3 Methods Employed

As noted in the previous section, the aims of this research were numerous. The methods employed to address these aims were similarly numerous.

To properly address the issues of ecological deep knowledge, it was necessary to properly review the literature of ecological knowledge-based systems (Chapter 2), consider the nature of ecological deep knowledge used by human ecologists in reasoning (Chapter 2), and the literature of, and existing methods used to represent, deep knowledge (Chapter 3). This review led to a suggestion of how best to represent ecological deep knowledge (Chapter 4).

Desirable criteria of an ideal uncertainty calculus, and existing methods of handling uncertainty in knowledge-based systems were then considered (Chapter 5), along with the proposed representation scheme of Chapter 4. The means that human ecologists, and ecological textbooks, employ to handle uncertainty (ie using qualitative statements) were also examined. From these considerations, a method of handling uncertainty was proposed (Section 5.6).

Having addressed both depth and uncertainty, it was necessary to consider two further aims of this research that are related; to make the deep reasoning process efficient and effective, and to appropriately control the reasoning process. To make the deep reasoning process more efficient, a method often employed to this end, 'knowledge compilation', was examined, and subsequently employed (Chapter 6). Control methods were then considered relative to the compiled knowledge. A novel method of 'best-first' search was developed, which relies on the gathering of most 'certain' information first, and appropriate balancing of contradicting/supporting information (Sections 6.5-6.9). The representation and reasoning scheme used has been termed the Qualitative Conviction Calculus (QCC).

In tandem with the theoretical considerations discussed above, a substantial body of practical work was also undertaken, concerning the proper comparison of first- and second-generation knowledge-based systems. This required three main aspects - the construction of these systems; comparison of these systems by suitable users, and; comparison of the performance of these two systems relative to the

domain expert (Dr. Rob Oldham), given a set of actual pond data.

Two systems were constructed in this research, both of which could evaluate a pond site to see if it is suitable to support a viable crested newt colony. The first- and second-generation systems were called HEX and TRITON respectively. The overall construction of both systems addressed a number of aspects relative to the knowledge engineering process (discussed more fully in Chapter 7). A further major undertaking within this research was the design, construction and use of a second generation knowledge-based system shell (called PERSEUS), within which TRITON was implemented. Details of the construction of PERSEUS are given in Appendix B.

Once HEX and TRITON were constructed, the systems were evaluated by two user groups, made up of student users and peer experts respectively (Chapter 8). These two groups used both systems, and were asked to comment (via questionnaire) on various aspects of each, including user interface, educational merit, the usefulness of the associated handbook, and so on. The peer experts were asked to comment on additional aspects which could not be suitably addressed by inexperienced individuals, including accuracy, robustness, questioning strategy, and the comparability of the 'default' explanations of HEX and TRITON.

In addition, the domain expert, HEX, and TRITON, were each presented with data from 100 pond sites. The domain expert was requested to state whether he considered each pond suitable or unsuitable for crested newts, and to state the degree of certainty in his answer. The data was input to HEX and TRITON, and the systems used to generate conclusions as to whether a pond was suitable or unsuitable for crested newts. It should be noted that TRITON also produces a degree of certainty in its answers.

## 10.4 Conclusions

The findings/conclusions of the present research are summarised in Table 37.

1. It is possible to build second-generation/deep knowledge-based systems within the domain of pond evaluation for a protected species that are as equally acceptable to end-users as first-generation/shallow systems.
2. It is possible to construct first- and second-generation knowledge-based systems that come to similar conclusions, even when different methods of representation, reasoning, and different grains and volumes of knowledge are used in each system.
3. In this work, the knowledge acquisition required in constructing the second-generation system was not significantly more 'expensive' or difficult than that required in building the first-generation system. Further, given adequate development tools, the overall knowledge engineering process is likewise not significantly more expensive or difficult in the second-generation system.
4. The control method used in the second-generation system (TRITON) gave rise to two significant findings; the users felt the second-generation system was likely to be more robust, but felt the questioning strategy of the first-generation system was more acceptable.
5. The default explanations of a second-generation knowledge-based system are significantly more acceptable to end-users than the default explanations of a first-generation system, in the present research. In addition, the default explanations of the second-generation knowledge-based system were found to be equally acceptable to end-users as explanations hand-crafted by the domain expert and used within the first-generation system.
6. In the application studied, the Qualitative Conviction Calculus (as embodied within the PERSEUS shell) generated conclusions and associated expressions of certainty similar to a human expert, based on the same data.

Table 37: The Findings of this Research

Within the present research, it has been established that it is possible, within the domain of habitat evaluation, to build second-generation knowledge-based systems that are as equally acceptable as first-generation systems to a range of users (in this case, relative novices, and domain experts). Some individual parameters were rated differently in first- and second-generation systems (eg questioning

strategy), but overall, both user groups within this study showed little difference in their rating of both a number of specific parameters, and in the overall rating of both systems (see Figure 6, page 138).

As well as each system being equally acceptable to end users, HEX and TRITON came to the same conclusions in 88.9% of cases. This demonstrates that, despite having different architectures, different control/reasoning methods, and different amounts and grains of knowledge, these two systems generate similar conclusions given the same data. However, the second-generation system is likely to have the benefits associated with deep knowledge. For example, the declarative (ie non-context-dependent) nature of deep knowledge may mean the knowledge base is more robust (a point supported by expert evaluators' opinions), easier to maintain (ie the knowledge engineer can be concerned solely with content, rather than content and ordering), and more easily reusable for other problems or contexts within this domain.

In the present research domain, the entire knowledge engineering process for both first- and second-generation knowledge-based systems required similar amounts of time and effort from parties involved (see Chapter 7). This should be contrasted with the speculations of some authors, who suggest the acquisition of deep knowledge is likely to be significantly more difficult than the acquisition of shallow knowledge (eg van Someren *et al.*, 1990). The work required within each of the knowledge engineering activities proved to be slightly different for each system, however (see Table 18). In many existing knowledge-based systems, the order of the knowledge base is critical to the way in which the knowledge is utilised/controlled by the system. In the architecture proposed within this research, the order of knowledge is of less importance; it is the content (specifically, the level of conviction) of each of the terms in the knowledge base that determines the sequence in which knowledge is utilised by the system. Systems constructed under the QCC architecture, such as TRITON, can therefore be referred to as 'content-critical', rather than 'order-critical'. However, the knowledge base design and knowledge acquisition phases of knowledge engineering are closely associated - a step-wise approach to constructing/ordering a knowledge base may direct the progress of knowledge acquisition, and ensure acquisition is exhaustive. For this reason, loss of a requirement for ordering would result in an unstructured approach to knowledge acquisition. Extra care and effort will therefore be needed to maintain the rigour of approach to knowledge acquisition.

Knowledge acquisition methods that have proved effective in the present research for the acquisition of deep knowledge are mainly those that fully record the explanations of the domain expert in justifying his reasoning processes (ie recording and transcribing meetings), and properly rechecking these justifications in later

sessions, and from different sources (eg other experts, texts). Chapter 7 contains fuller details of knowledge acquisition methods that have proved effective in the acquisition of deep knowledge within this research.

In assessing the default explanations of first- and second-generation knowledge-based systems in this research, the users showed a significantly (and in many cases a very highly significantly) greater rating of the second-generation explanations. There may be a number of contributing reasons for this, including the greater detail and declarative nature of the second-generation statements (ie such statements are relatively free of a specific, and possibly idiosyncratic, context). The improvement in acceptability of the second-generation statements of explanation may prove of benefit to knowledge engineers, as existing systems are likely to require 'added' explanation facilities, so as to provide acceptable explanations. In this research, the addition of such explanations to HEX was found to be a non-trivial task that required extensive expert input. By providing an automatically-generated set of explanations that are more acceptable to users, it may be possible to reduce the workload of both knowledge engineer and expert.

Another finding of the present research was that the qualitative conviction calculus tended to generate similar conclusions/convictions as a human expert, given the same data. When the expert and TRITON disagreed in conclusions (eg the expert felt the pond suitable, TRITON concluded it was unsuitable), both tended to express a low confidence in their conclusions. This indicates that, in this application domain, it is possible that the QCC (and more generally the PERSEUS shell) is modelling the human expert's decision-making processes under qualitative uncertainty. This aspect is commented upon further in the next section.

## 10.5 Implications

This research indicates that it is possible, given appropriate tools, to build second-generation knowledge-based systems that work realistically, in the domain of habitat evaluation. In this particular domain, it was found the first- and second-generation systems were on the whole equally acceptable to users, required roughly equal amounts of time and resources to construct (but required different emphases within the specific steps of the knowledge engineering process), and generated conclusions with an equivalent degree of accuracy.

A number of considerations must be taken into account by a knowledge engineer in deciding whether to build a first- or second-generation system. These include the availability of suitable tools, the effort required to build different types of system, and the size or likely uses of the final system. Most of the tools available at

present to build knowledge-based systems can only be used to construct first-generation systems; the construction of a second-generation system from scratch (eg from a programming language) is a non-trivial task. However, there may be benefits associated with the building of second-generation systems (acceptable explanations are generated within the machine, rather than needing to be hand-built) that may make such an undertaking worthwhile. For example, if the knowledge engineer anticipates a knowledge base will be changing regularly, then the initial effort to build the second-generation system may be worthwhile. The nature of the domain and the problem to which a knowledge-based system is targeted may dictate whether a first- or second-generation system should be employed. For example, a system that is very small and requiring little knowledge may be better constructed as a first-generation system. On the other hand, a larger system, or a system where knowledge is reused, may be best developed and maintained in a declarative form commonly associated with second-generation systems.

At a theoretical level, the results in this research indicate that the QCC may be a reasonably acceptable model of human reasoning under uncertainty, in the domain of pond evaluation. This is indicated by two bodies of evidence; the correspondence of the conclusions and associated statements of certainty of the domain expert with those of TRITON, and the acceptance of TRITON's conclusions and statements of certainty by the majority of the expert evaluators.

Several authors have suggested that AI/KBS methods may be a means to rationally embody, and facilitate the development of, ecological science (Rykiel, 1989; Noble, 1987). Such methods may facilitate organisation and dissemination of material, allow the rapid assessment of assumptions, hypotheses and so on, and be used to determine the likely consequences and logical consistency of long and complicated reasoning paths (Rykiel, 1989). The development of deep systems may facilitate the implementation of AI tools that aid in development of the science of ecology. The work discussed here may be a means to implement these suggestions. As Noble (1987) notes:

If expert systems are used only to bring together a number of ecological rules-of-thumb and to package them in a way more readily acceptable to a user then ecological understanding will advance very little. If, however, in our attempt to formulate the knowledge bases, we are forced to rethink the nature of ecological relationships then expert systems may have some impact.

## 10.6 Future Work

In this research, there has been specific identification of the various levels of efforts required in building a first and second-generation knowledge based system, along with some identification of the relative benefits/limitations associated with each. One strand of further work that could complement this information would be the formal identification of situations where each type of system may be most appropriately used.

Future work that builds upon the present research includes the further validation/refinement of the PERSEUS shell (within which the QCC methodology is embodied). Whilst the results in this research suggest that the second-generation architecture developed has proved relatively successful, it must be said that the success is at present confined to the domain of habitat evaluation. To properly validate the methodology developed in this research, the PERSEUS shell must be successfully used in a domain other than habitat evaluation for crested newts. A suitable progression for further testing of the PERSEUS shell would be to select a set of domains for testing, ranging from a domain close to the original domain (eg a system for evaluating ponds with respect to suitability for common frogs), through to domains within ecology, but assessing different habitats. Such assessment would also facilitate the consideration of whether the knowledge base of the original system was reusable, and perhaps quantify this aspect. If this stage is reached, then it would be necessary to test the PERSEUS shell in a domain other than ecology, but with a similarly 'soft' conceptual basis (eg economics, psychology).

It has been noted in Section 10.5 that the present research indicates the QCC may be a possible model of human reasoning under uncertainty. To evaluate and/or refine the QCC as a model of human reasoning under uncertainty, further testing is required, and would be a significant part of the potential future progress outlined in the previous paragraph. However, to properly test and develop the QCC as a model of generic human uncertainty processing (particularly the processes of compilation and knowledge senescence/renascence) would require tests occurring in a generic (ie domain-independent) setting (eg empirical testing using disinterested subjects, likely to be considering 'common sense' phrases).



## 10.7 Summary

To conclude, in the domain of habitat evaluation, first- and second-generation knowledge-based systems were found to be comparable to both novice and expert users, via a number of parameters. Equally, the systems were comparable in accuracy, and the conclusions and associated statements of certainty of the second-generation system were similar to those of the domain expert in a substantial number of cases. This research has established the feasibility of building usable second-generation knowledge-based systems for the purposes of habitat evaluation.

In total, these findings represent novel information that substantially adds to the total body of knowledge concerning ecological knowledge-based systems.

## REFERENCES

- Abu-Hanna, A. & Gold, Y.I. (1990) Adaptive, Multilevel diagnosis and Modeling of Dynamic Systems, *International Journal of Expert Systems*, 3(1): 1-30
- Agresti, A. (1990) *Categorical Data Analysis*, John Wiley and Sons: New York.
- Aikins, J. (1980) Representation of Control Knowledge in Expert Systems, *Proceedings of the National Conference on Artificial Intelligence: AAAI-80*, Stanford University, pp.121-123
- Aikins, J. (1983) Prototypical Knowledge for Expert Systems, *Artificial Intelligence*, 20: 163-210
- Allen, J.F. (1981) An Interval-based Representation of Temporal Knowledge, in Drinan, A. (Ed.) *Proc. 7th International Joint Conference of Artificial Intelligence: IJCAI-81*, William Kaufmann: Los Altos, Calif. pp.221-226
- Allen, J.F. (1984) Towards a General Theory of Action and Time, *Artificial Intelligence*, 23: 123-154
- Anderson, J.R. (1983) *The Architecture of Cognition*, Harvard University Press: Cambridge, Massachusetts.
- Araya, A.A. & Mittal, S. (1987) Compiling Design Plans from Descriptions of Artifacts and Problem Solving Heuristics, in: *Proceedings of the 10th International Joint Conference on Artificial Intelligence: IJCAI-87, Volume 1*, Edited by J.McDermott, Morgan Kaufmann: Los Altos, Calif. pp.552-558
- Attarwala, F.T. & Basden, A. (1985) A methodology for constructing expert systems, *R & D Management* 15(2): 141-149
- Bainbridge, B. (1988) The explicit representation of control knowledge, in: *Approaches to Knowledge Representation: An Introduction*, edited by G.A. Ringland & D.A. Duce, John Wiley, New York. pp.175-187
- Barnett, J.A. (1982) Some Issues of Control in Expert Systems, *Proceedings of the International Conference on Cybernetics and Society*, IEEE Computer Society, pp.1-5
- Baur, G. & Pigford, D.V. (1990) *Expert Systems for Business: Concepts and Applications*, Boyd & Fraser: Boston.
- Becker, L.A., Bartlett, R. & Soroushian, F. (1989) Using Simulation to Compile Diagnostic Rules from a manufacturing Process Representation, *SIGART Newsletter*, 108, April, 172-173
- Beckman, T.J. (1991) Selecting Expert System Applications, *AI Expert*, 6(2): 42-48
- Begon, M., Harper, J.L., & Townsend, C.R. (1986) *Ecology: Individuals, Populations and Communities*, Blackwell Scientific Publications, Oxford.
- Bench-Capon, T.J.M. (1991) *Knowledge Representation: An Approach to Artificial Intelligence*, Academic Press: London.
- Berry, D.C. & Hart, A.E. (1990) Evaluating expert systems, *Expert Systems*, 7(4): 199-208
- Bhatnagar, R.K. & Kanal, L.N. (1986) Handling Uncertain Information: A Review of numeric and non-numeric methods, *Uncertainty in Artificial Intelligence*, Kanal, L.N. & Lemmer, J.F. (Eds), Elsevier Science Publ., North-Holland. pp.3-26

- Bobrow, D.G. (1985) *Qualitative Reasoning about Physical Systems*, MIT Press, Cambridge, Massachusetts.
- Bonissone, P. (1987) Reasoning, Plausible, in: *Encyclopedia of Artificial Intelligence, volume 2*, edited by S.C.Shapiro, John Wiley & Sons: New York, pp.854-863
- Bonissone, P. (1990) Now that I have a good theory of uncertainty, what else do I need? in: *Uncertainty in Artificial Intelligence 5*, edited by M. Henrion, R.D. Shachter, L.N.Kanal & J.F.Lemmer. Elsevier Science: North-Holland. pp.237-253
- Bonissone, P.P. & Tong, R.M. (1985) Editorial: Reasoning with uncertainty in expert systems, *International Journal of Man-Machine Studies*, 22: 241-250
- Bratko, I. (1990) *PROLOG: Programming for Artificial Intelligence (2nd Ed.)*, Addison-Wesley: Wokingham, England.
- Bratko, I., Mozetic, I. & Lavrac, N. (1988) Automatic Synthesis and Compression of Cardiological Knowledge, *Machine Intelligence 11*, Hayes, J.E., Michie, D., & Richards, J. (eds), Clarendon Press, Oxford, pp.435-454.
- Bredeweg, B. & Wielinga, B.J. (1988) Integrating Qualitative Reasoning Approaches, in: *Proceedings of the 8th European Conference on Artificial Intelligence*, pp.195-201
- Brown, D.C. (1991) Compilation: The Hidden Dimension of Design Systems, in: *Intelligent CAD III*, edited by H. Yoshikawa, F. Arbab, and T. Tomiyama, Elsevier Science: North Holland. pp.99-108
- Buchanan, B., Barstow, D., Bechtel, R., Bennett, J., Clancey, W., Kulikowski, C., Mitchell, T. & Waterman, D. (1983) Constructing an Expert System, in: *Building Expert Systems*, edited by F. Hayes-Roth, D.A. Waterman, & D.B. Lenat, Addison-Wesley: Reading, Massachusetts. pp.127-167
- Buchanan, B.G. & Shortliffe, E.H. (Eds.) (1984) *Rule-Based Expert Systems: The MYCIN Experiments*, Addison-Wesley: Reading, Massachusetts.
- Buchanan, B.G. & Smith, R.G. (1989) Fundamentals of expert systems, *The Handbook of Artificial Intelligence IV*, Edited by Barr, A., Cohen, P.R. & Feigenbaum, E.A.: Addison-Wesley, Reading, Massachusetts. pp.149-192
- Buech, R.R., Martin, C.J. & Rauscher, H.M. (1990) An Application of expert systems for wildlife: Identifying forest stand prescriptions for deer, *Artificial Intelligence Applications in Natural Resource Management*, 4(1): 3-10
- Bundy, A. (1984) *Incidence Calculus: A Mechanism for Probabilistic Reasoning*, DAI Research Paper No.216, Dept of Artificial Intelligence, University of Edinburgh.
- Burton, A.M., Shadbolt, N.R., Hedgecock, A.P. & Rugg, A. (1987) A formal evaluation of knowledge elicitation techniques for expert systems: Domain 1, in: *Knowledge Acquisition for Engineering Applications*, by Pavelin, C.J. & Wilson, M.D. (Eds), Rutherford Appleton Laboratories Report RAL-87-055: Didcot, UK. pp.20-28
- Bylander, T. (1990) Some causal models are deeper than others, *Artificial Intelligence in Medicine*, 2(3): 23-128
- Bylander, T., Smith, J.W., & Svirbely, J.R. (1988) Qualitative Representation of Behaviour in the Medical Domain, *Computers and Biomedical Research*, 21: 367-380

Cain, M.F. (1990) The Techniques of Knowledge Elicitation and their use in the building of deep and shallow knowledge based systems, *Computing and Mathematical Sciences Research: Working Paper 35*, Dept of Computing and Mathematical Sciences, Leicester Polytechnic.

Cain, M.F. (1993) The Application of Artificial Intelligence Techniques in Ecological Domains: A Suggested Classification and Bibliography, *Computing and Mathematical Sciences Research: Working Paper 43*, Dept of Computing and Mathematical Sciences, De Montfort University.

Camara, A.S., Antunes, P.C., Pinheiro, M.D., & de Seixas, M.J.F. (1987) Linguistic Dynamic Simulation - A new approach, *Simulation*, **49**(5): 208-212

Chandrasekaran, B. (1991) Models versus rules, deep versus compiled, content versus form: Some distinctions in knowledge systems research, *IEEE Expert* **6**(2): 75-79

Chandrasekaran, B, Bylander, T. & Sembugamoorthy, V. (1985) Functional Representations and Behaviour Composition by Consolidation: Two Aspects of Reasoning about devices, in: Chandrasekaran & Milne (Eds) Special Section on Reasoning about Structure, Behaviour, and Function. *SIGART Newsletter* **93**: 21-24

Chandrasekaran, B. & Milne, R. (Eds) (1985) Special Section on Reasoning about Structure, Behaviour, and Function. *SIGART Newsletter* **93**: 4-55

Chandrasekaran, B. & Mittal, S. (1983a) Conceptual Representation of Medical Knowledge for Diagnosis by Computer: MDX and Related Systems, *Advances In Computers*, **22**: 217-293

Chandrasekaran, B. & Mittal, S. (1983b) Deep versus compiled knowledge approaches to diagnostic problem-solving, *International Journal of Man-Machine Studies*, **19**: 425-436

Chandrasekaran, B., Smith, J.W., & Sticklen, J. (1989) 'Deep' Models and their relation to diagnosis, *Artificial Intelligence in Medicine*, **1**: 29-40

Cheeseman, P. (1985) In defense of Probability, *Proceedings of the 9th International Joint Conference of Artificial Intelligence: IJCAI-85 (2nd volume)*, Los Altos, California, pp. 1002-1009

Clancey, W.J. (1979) Transfer of Rule-based expertise through a tutoring dialogue, *Ph.D. Dissertation*, Stanford University.

Clancey, W.J. (1983) The Advantages of Abstract Control Knowledge in Expert System Design, *Proceedings of the National Conference on Artificial Intelligence: AAAI-83*, William Kaufmann: Los Altos, Calif. pp.74-78

Clancey, W.J. & Letsinger, R. (1981) NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching, in: Drinan, A. (Ed.) *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, William Kaufmann: Los Altos, Calif. Vol.2, pp.829-836

Clark, D. (1988) AI and the Management of Uncertainty, *Knowledge Engineering Review*, **3**(1): 59-63

Clark, D. (1990) Numerical and Symbolic approaches to uncertainty management in AI, *Artificial Intelligence Review*, **4**: 109-146

- Clegg, C., Warr, P., Green, T., Monk, A., Kemp, N., Allison, G. & Lansdale, M. (1988) *People and Technology: How to evaluate your company's new technology*, Ellis-Horwood: Chichester.
- Cohen, P.R. (1985) *Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach*, Pitman: Boston.
- Cohen, P.R. (1987a) The Control of Reasoning under Uncertainty: A discussion of some programs, *Knowledge Engineering Review*, 2(1): 5-25
- Cohen, P.R. (1987b) Numeric and Symbolic Reasoning in Expert Systems, in: *Advances in Artificial Intelligence II*, edited by B. du Boulay, D. Hogg & L. Steels, Elsevier Science: North-Holland. pp.527-541
- Cohen, P.R. & Grinberg, M.R. (1983) A Framework for Heuristic Reasoning about Uncertainty, in: Bundy, A. (Ed.) *Proceedings of the 8th International Joint Conference on Artificial Intelligence: IJCAI-83, Volume 1*, William Kaufmann: Los Altos, California, pp.355-357
- Cohn, A.G. (1985) Deep Knowledge Representation Techniques, in: Merry, M. (Ed.) *Expert Systems '85: Proceedings of the 5th Technical Conference of the BCS Specialist Group on Expert Systems*, Cambridge University Press. pp.299-306
- Cohn, A.G. (1989) Approaches to qualitative reasoning, *Artificial Intelligence Review*, 3(2/3): 177-232
- Coiera, E. (1992) The Qualitative Representation of Physical Systems, *The Knowledge Engineering Review*, 7(1): 55-77
- Colfer, C.J.P., Yost, R., Agus, F. & Evensen, J. (1989) Expert Systems: A possible link from field work to policy via farming systems, *AI Applications in Natural Resource Management* 3(2): 31-40.
- Colinvaux, P. (1986) *Ecology*, John Wiley: New York.
- Conley, W. & Sengupta, U. (1989) A demographic simulator with deeply coupled semantic and numeric data structures, *Ecological Modelling* 46: 35-56
- Console, L., Molino, G., Pavia, R., Signorelli, M & Torasso, P. (1989) Simulating Generic Situations on Causal Models, in: *AIME-89: Proceedings of the European Conference on Artificial Intelligence in Medicine*, Springer-Verlag. pp.177-186
- Cooke, A.S. (1972) The effects of DDT, dieldrin and 2,4D on amphibian spawn and tadpoles, *Environmental Pollution* 3: 51-68
- Cooke, A.S. (1974) Differential predation by newts on anuran tadpoles. *Brit. J. Herpetol.* 5: 386-390
- Cooke, A.S. & Frazer, J.F.D (1976) Characteristics of newt breeding sites, *Journal of Zool. Lond.* 178: 223-236
- Coulson, R.N., Folse, L.J. & Loh, D.K. (1987) Artificial Intelligence and Natural Resource Management *Science* 237: 262-267 (17 July 1987).
- Cox, E. (1992) The Great Myths of Fuzzy Logic, *AI Expert*, 7(11):40-45
- Craik, K. (1943) *The Nature of Explanation*, Cambridge University Press.

- Creative Logic (1987) *Leonardo: The Manual*, Creative Logic Ltd, Middlesex.
- Creed, K. (1964) A study of newts in the New Forest, *Brit. J. Herpet.* **3**: 170-181
- Dan, Q. and Dudeck, J. (1992) Certainty Factor Theory: Its probabilistic interpretations and problems, *Artificial Intelligence in Medicine*, **4(1)**: 21-34
- Davey, S.M. & Stockwell, D.R.B. (1991) Incorporating Wildlife Habitat into an AI Environment: Concepts, Theory, and Practicalities, *AI Applications*, **5(2)**: 59-104
- Davis, J.R. & Nanninga, P.M. (1985) GEOMYCIN: Towards a geographic expert system for resource management, *Journal of Environmental Management* **21**: 377-390
- Davis, R. (1979) Panel on Dealing with Uncertainty, in: *Proceedings of the 6th International Conference on Artificial Intelligence: IJCAI-79 (Volume 2)*, Information Processing Society of Japan. pp.1101-1102
- Davis, R. (1980) Meta-Rules: Reasoning about Control, *Artificial Intelligence*, **15**: 179-222
- Davis, R. (1984) Diagnostic Reasoning Based on Structure and Behaviour, *Artificial Intelligence*, **24**: 347-410
- De Kleer, J. (1993) A View on Qualitative Physics, *Artificial Intelligence*, **59(1,2)**: 105-114
- De Kleer, J. & Brown, J.S. (1981) Mental Models of Physical Mechanisms and their acquisition, in: Anderson, J.R. (Ed.) *Cognitive Skills and their Acquisition*, Erlbaum Associates, Hillsdale, New Jersey. pp.285-309
- De Kleer, J. & Brown, J.S. (1983) The origin, form, and logic of qualitative physical laws, in: Bundy, A. (Ed.) *Proceedings of the 8th International Joint Conference on Artificial Intelligence: IJCAI-83, Volume 2*, William Kaufmann: Los Altos, California, pp.1158-1169
- De Kleer, J. & Brown, J.S. (1984) A Qualitative Physics based on Confluences, *Artificial Intelligence* **24**: 7-83
- Dempster, A.P. (1967) Upper and lower probabilities induced by a multivalued mapping, *Ann. Math. Stat.*, **38**: 325-399
- Dietterich, T.G. (1991) Bridging the Gap between Specification and Implementation, *IEEE Expert*, **6(2)**: 80-82
- Doyle, J.A. (1979) A Truth Maintenance System, *Artificial Intelligence*, **12**: 231-272
- Doyle, J.A. (1983) Methodological simplicity in expert system construction: The case of judgements and reasoned assumptions, *AI Magazine*, **4(2)**: 39-43
- Dugerdil, P. & Guillod, J. (1990) Deep model reasoning in a medical expert system, in *10th International Workshop: Expert Systems and Their Applications - Second Generation Expert Systems*, EC2, pp.85-96
- Egler, F.E. (1986) Physics Envy in Ecology, *Bulletin of the Ecological Society of America*, **67**: 233-235
- Ekblad, S.L., Hamilton, W.T., Stuth, J.W., & Conner, J.R. (1991) RangeCon: A management evaluation system for assessing success of selected range improvement practices, *AI Applications* **5(3)**:9-20

- Eldredge, N. (1987) *Life Pulse*, Penguin: London.
- Faltings, B. (1992) A Symbolic Approach to Qualitative Kinematics, *Artificial Intelligence*, **56(2,3)**: 139-170
- Fayers, P.M. & Jones, D.R. (1983) Measuring and analysing quality of life in cancer clinical trials: a review, *Statistics in Medicine*, **2**: 429-446
- Fieschi, M. (1990) *Artificial Intelligence in Medicine: Expert Systems*, (Translated by D. Cramp), Chapman and Hall: London.
- Folse, L.J., Packard, J.M. & Grant, W.E. (1989) AI Modelling of Animal Movements in a heterogenous habitat *Ecological Modelling* **46**: 57-72
- Forbus, K.D. (1984) Qualitative Process Theory, *Artificial Intelligence*, **24**: 85-168
- Forbus, K. (1985) Qualitative Physics, in: Chandrasekaran & Milne (Eds) Special Section on Reasoning about Structure, Behaviour, and Function. *SIGART Newsletter* **93**: 7-9
- Forbus, K.D., Nielsen, P. & Faltings, B. (1987) Qualitative Kinematics: A Framework, *Proceedings of the 10th International Joint Conference on Artificial Intelligence: IJCAI-87, Volume 1*, Edited by J.McDermott, Morgan Kaufmann: Los Altos, Calif. pp.430-435
- Forsyth, R. (Ed.) (1989) *Expert Systems: Principles and Case Studies (2nd Edit.)*, Chapman & Hall: London.
- Fox, J. (1984) Inferences about beliefs and values, *ALVEY IKBS Inference Research Workshop*, Imperial College, London.
- Fox, J. (1986a) Knowledge, Decision Making, and Uncertainty, in: *Artificial Intelligence and Statistics*, edited by W.A.Gale, Addison-Wesley, Reading, Massachusetts. pp.57-76
- Fox, J. (1986b) Three arguments for extending the framework of probability, in: *Uncertainty in Artificial Intelligence*, Kanal, L.N. & Lemmer, J.F. (Eds), Elsevier Science Publ., North-Holland. pp.447-458
- Fox, J., Barber, D. & Bardhan, K.D. (1980) Alternatives to Bayes? A Quantitative Comparison with rule-based diagnostic inference, *Methods of Informatics in Medicine*, **19(4)**: 210-215
- Fox, J., Clark, D.A., Glowinski, A.J. & O'Neil, M.J. (1990) Using Predicate Logic to Integrate Qualitative Reasoning and Classical Decision Theory, *IEEE Transactions on Systems, Man, and Cybernetics* **20(2)**: 347-357
- Fox, J., Glowinski, A., O'Neil, M.J., & Clark, D.A. (1988) Decision Making as a logical process, in: Kelly, B. & Rector, A. (Eds) *Research and Development in Expert Systems V*, Cambridge University Press, pp.160-175
- Frazer, D. (1989) *Reptiles and Amphibians in Britain*, Bloomsbury:London.
- Frost, R.A. (1986) *Introduction to Knowledge Based Systems*, Collins: London.
- Fu, L-M. (1991) CAUSIM: A rule-based causal simulation system, *Simulation*, **56(4)**: 251-257
- Gardner, J. and Munroe, G. (1992) Authoring and Authoring Systems, *The CTISS File*, **13** (April): 45-46.



- Garvey, T.D., Lowrance, J.D. & Fischler, M.A. (1981) An inference technique for integrating knowledge from disparate sources, *Proceedings of the 7th International Joint Conference on Artificial Intelligence: IJCAI-81*, edited by A. Drinan, William Kaufmann: Los Altos, Calif. pp.319-325
- Gashnig, J. (1982) Prospector: An expert system for mineral exploration, *Infotech State of the Art Report*, Series 9, No. 3.
- Genesereth, M.R. (1984) The use of design descriptions in automated diagnosis, *Artificial Intelligence*, 24: 411-436
- Gertner, G. & Guan, B.T. (1990) Conceptual Classification: An AI alternative to statistical procedures for classification problems, *AI Applications in Natural Resource Management*, 4(4): 25-32
- Gleick, J. (1987) *Chaos*, Cardinal: London.
- Good, J.I. (1950) *Probability and the Weighting of Evidence*, New York: Hafner.
- Gordon, J. & Shortliffe, E.H. (1984) The Dempster-Shafer Theory of Evidence, in: Buchanan, B.G. & Shortliffe, E.H. (Eds.) *Rule-Based Expert Systems: The MYCIN Experiments*, Addison-Wesley: Reading, Massachusetts. pp.272-292
- Gordon, S.E. (1989) Theory and Methods for Knowledge Acquisition, *AI Applications in Natural Resource Management* 3(3): 19-30
- Gould, S.J. (1989) *Wonderful Life: The Burgess Shale and the Nature of History*, Penguin: London.
- Graham, I. (1989) Inside the Inference Engine, in: Forsyth, R. (Ed.) *Expert Systems: Principles and Case Studies (2nd Edit.)*, Chapman & Hall: London. pp.57-83.
- Green, D. (1984) A study of the great crested newt (*Triturus cristatus*) in Durham and Wear, South. *Durham County Conservation Trust Ltd.*
- Guariso, G. & Werthner, H. (1989) *Environmental Decision Support Systems*, Ellis Horwood: Chichester.
- Guha, R.V. & Lenat, D.B. (1990) CYC: A midterm report, *AI Magazine*, 11(3): 32-59
- Guida, G. (1986) Reasoning about Physical Systems: Shallow versus deep models, in: Mamdani, A. & Efstathiou, J. (Eds) *Expert systems and Optimisation in Process Control*, Technical Press: Aldershot, pp.135-159
- Haeckel, E. (1866) *Generelle Morphologie Der Organismen*, Berlin, Reimer (2 volumes).
- Hagstrom, T. (1979) Population ecology of *Triturus cristatus* and *T. vulgaris* (Urodela) in S.W. Sweden, *Holarctic Ecolog.* 2: 108-114
- Hart, A. (1986) *Knowledge Acquisition for Expert Systems*, Kogan Page Ltd.
- Hart, P.E (1982) Direction for AI in the eighties, *SIGART Newsletter*, 79: 11-16
- Hayes, P.J. (1979) The Naive Physics Manifesto, in: D. Michie (ed.), *Expert Systems in the microelectronic age*, Edinburgh University Press, pp. 242-270.

- Heckerman, D.E. & Shortliffe, E.H. (1992) From Certainty Factors to Belief Networks, *Artificial Intelligence in Medicine*, 4(1): 35-52
- Henrion, M. (1987) Uncertainty in Artificial Intelligence: Is probability epistemologically and heuristically adequate?, in: *Expert Judgement and Expert Systems*, edited by J. Mumpower, L. Philips, O. Renn, & V. Uppuluri, Springer-Verlag: Berlin. pp.105-129
- Henwood, K.L. & Pidgeon, N.F. (1992) Qualitative Reasoning and Psychological Theorizing, *British Journal of Psychology*, 83: 97-111
- Herbert, M.R. & Williams, G.H. (1987) An initial evaluation of the detection and diagnosis of power plant faults using a deep knowledge representation of physical behaviour, *Expert Systems* 4(2): 90-99
- Hilton-Brown, D. & Oldham, R.S. (1991) The Status of the widespread amphibians and reptiles in Britain, 1990, and changes during the 1980's, *Nature Conservancy Council Report No.131*, ISSN 0952-4355.
- Hokans, R.H. (1984) An artificial intelligence application to timber harvest schedule implementation, *INTERFACES* 14(5): 77-84
- Holt, J., Cheng, J.A. & Norton, G.A. (1990) A Systems Analysis approach to brown planthopper control on rice in Zhejiang Province, China. III. An Expert System for making recommendations, *Journal of Applied Ecology* 27: 113-122
- Horn, W. (1991) Utilizing detailed anatomical knowledge for hypothesis formation and hypothesis testing in rheumatological decision support, *Artificial Intelligence in Medicine*, 3(1): 21-39
- Horvitz, E. & Heckerman, D.E. (1986) The inconsistent use of measures in certainty in Artificial Intelligence Research, in: *Uncertainty in Artificial Intelligence*, edited by L.N. Kanal & J.F. Lemmer, Elsevier Science: North Holland. pp.137-151
- Howard, R.J. & Kantrud, H.A. (1986) Habitat Suitability Index Models: Northern Pintail (Gulf Coast Wintering), *U.S. Fisheries and Wildlife Services Biological Report* 82 (10.121).
- Hunter, J., Kirby, I. & Gotts, N. (1991) Using quantitative constraints in models of cardiac physiology, *Artificial Intelligence in Medicine*, 3(1): 41-61
- Jackson, P. (1990) *Introduction to Expert Systems (2nd Edit.)*, Addison-Wesley: Wokingham, England.
- Jeffcote, M. (1991) *The Role of Expert Systems in Conservation Management*, M.Phil. Thesis, Leicester Polytechnic.
- Johnson, N.E. (1985) Varieties of Representation in Eliciting and Representing Knowledge in IKBS, *International Journal Syst. Res. Information Sci.*, 1(2): 69-90
- Johnson-Laird, P.N. (1977) Reasoning with Quantifiers, in: *Thinking: Readings in Cognitive Science*, Cambridge University Press, pp.129-142
- Johnson-Laird, P.N. (1983) *Mental Models*, Cambridge University Press.

- Jones, M.J. (1988) Counsellor - An expert system application in the agrochemical industry, in: Zazueta, F.S. & Bottcher, A.B. (Eds), *Proceedings of the 2nd Int. Conf. on Computers in Agricultural Extension Programs Vol. 1*, The Florida Cooperative Extension Service, Institute of Food & Agricultural Sciences. pp.209-214
- Jones, P. (1989) Uncertainty Management in Expert Systems, in: Forsyth, R. (Ed.) *Expert Systems: Principles and Case Studies (2nd Edit.)*, Chapman & Hall: London. pp.106-121
- Kahneman, D, Slovic, P. & Tversky, A. (1982) *Judgement under Uncertainty: Heuristics and Biases*, Cambridge University Press.
- Keller, R.M. (1991) Applying Knowledge Compilation Techniques to Model-Based Reasoning, *IEEE Expert*, 6(2): 82-87
- Kendall, K.E. & Kendall, J.E. (1992) *Systems Analysis and Design (2nd Edit.)*, Prentice-Hall, Englewood Cliffs, NJ.
- Keravnou, E.T. & Washbrook, J. (1989) What is a deep expert system? An analysis of the architectural requirements of second generation expert systems, *Knowledge Engineering Review*, 4(3): 205-233.
- Keuneke, A.M. (1991) Device Representation: The Significance of Functional Knowledge, *IEEE Expert*, 6(2): 22-25
- Keyes, J. (1989) Why expert systems fail, *AI Expert*, 4(1): 50-53
- Klein, D. & Finin, T. (1987) What's in a deep model? A characterisation of knowledge depth in intelligent safety systems, in; McDermott, J. (ed.), *IJCAI '87; Proceedings of the 10th international joint conference of AI Vol.1*, Los Altos: Morgan Kaufmann 559-562
- Krebs, C.J. (1985) *Ecology: The Experimental Analysis of Distribution and Abundance (3rd Edit.)*, Harper & Row: New York.
- Kreutzer, W. & McKenzie, B. (1991) *Programming for Artificial Intelligence: Methods, Tools and Applications*, Addison-Wesley: Sydney.
- Kuhn, T.S. (1970) *The Structure of Scientific Revolutions (2nd edit.)*, Chicago University Press.
- Kuipers, B. (1979) On Representing Commonsense Knowledge, in: Findler, N.V. (Ed.) *Associative Networks: Representation and Use of Knowledge by Computers*, Academic Press, New York. pp.393-408
- Kuipers, B. (1984) Commonsense Reasoning about Causality: Deriving Behaviour from Structure, *Artificial Intelligence*, 24: 169-203
- Kuipers, B. (1986) Qualitative Simulation, *Artificial Intelligence*, 29: 289-338
- Kuipers, B. (1987) Reasoning, Causal, *Encyclopedia of Artificial Intelligence, volume 2*, edited by S.C.Shapiro, John Wiley & Sons: New York, pp.827-832
- Kuipers, B. & Kassirer, J. (1984) Causal Reasoning in Medicine: Analysis of a Protocol, *Cognitive Science*, 8: 363-385
- Kunstaetter, R. (1987) Intelligent physiologic modeling: an application of knowledge based systems technology to medical education, *Computer Methods and Programs in Biomedicine*, 24: 213-225

- Kunz, J.C. (1983) Analysis of Physiological Behaviour using a causal model based on first principles, *Proceedings of the National Conference on Artificial Intelligence: AAAI-83*, William Kaufmann: Los Altos, Calif. pp.225-228
- Lauritzen, S.L. & Spiegelhalter, D.J. (1988) Local computations with Probabilities on Graphical Structures and their Application to Expert Systems, *Journal of the Royal Statistical Society (B)* 50(2): 157-224
- Lavrac, N. & Mozetic, I. (1989) Methods for knowledge acquisition and refinement in second generation expert systems, *SIGART Newsletter*, 108, April, 63-69
- Lenat, D.B., & Feigenbaum, E.A. (1991) On the thresholds of knowledge, *Artificial Intelligence*, 47: 185-250
- Lindley, D.V. (1987) The probability approach to the treatment of uncertainty in artificial intelligence and expert systems, *Statistical Science*, 2(1): 17-24
- Locke, J. (1961) *An Essay concerning Human Understanding*, Reprint of 5th edition, Dent: London (originally published in 1706).
- Loehle, C. (1987) Applying Artificial Intelligence Techniques to Ecological Modelling, *Ecological Modelling* 38: 191-212
- Loehle, C. & Osteen, R. (1990) IMPACT: An expert system for environmental impact assessment, *Artificial Intelligence Applications in Natural Resource Management*, 4(1): 35-43
- Long, W.J. (1991) Flexible Reasoning about patient management using multiple models, *Artificial Intelligence in Medicine*, 3(1): 3-20
- Luger, G.F. & Stubblefield, W.A. (1989) *Artificial Intelligence and the Design of Expert Systems*, Benjamin/Cummings, Redwood City, California.
- McDermott, D. (1982) A temporal logic for reasoning about processes and plans, *Cognitive Science*, 6(2): 101-155
- McNay, R.S., Page, R.E. & Campbell, A. (1987) Application of expert-based decision models to promote integrated management of forests and deer, *Trans N.A. Wildl. and Natur.Res. Conf.* 52: 82-91
- Mamdani, E.H. & Efstathiou, H.J. (1985) Higher-order logics for handling uncertainty in expert systems, *International Journal of Man-Machine Studies*, 22: 283-293
- Maynard Smith, J. (1974) *Models in Ecology*, Cambridge University Press.
- Michie, D. (1982) High-road and low-road programs, *AI Magazine* 3(1): 21-22.
- Miller, P.L. & Sittig, D.F. (1990) The evaluation of clinical decision support systems: what is necessary versus what is interesting, *Medical Informatics*, 15(3): 185-190
- Minsky, M. (1991) Logical versus analogical or symbolic versus connectionist or neat versus scruffy, *AI Magazine*, 12(2): 34-51
- Mozetic, I. (1990) Diagnostic efficiency of deep and surface knowledge in KARDIO, *Artificial Intelligence in Medicine*, 2: 67-83

- Muetzelfeldt, R., Robertson, D., Bundy, A. & Uschold, M. (1989) The use of PROLOG for improving the rigour and accessibility of ecological modelling, *Ecological Modelling*, **46**: 9-34
- Naylor, C. (1989) How to build an inference engine, in: Forsyth, R. (Ed.) *Expert Systems: Principles and Case Studies (2nd Edit.)*, Chapman & Hall: London. pp.84-105
- Neale, I.M. (1988) First generation expert systems: A review of knowledge acquisition methodologies, *Knowledge Engineering Review*, pp.105-145
- Neves, D.M. & Anderson, J.R. (1981) Knowledge Compilation: Mechanisms for the Automatization of Cognitive Skills, in: *Cognitive Skills and their Acquisition*, edited by J.R.Anderson, Lawrence Erlbaum, Hillsdale, NJ. pp.57-84
- Newell, A. & Simon, H.A. (1976) Computer Science as Empirical Enquiry: Symbols and Search, *Communications of the ACM*, **19(3)**: 113-126
- Ng, H.T. (1991) Model-based, multiple-fault diagnosis of dynamic, continuous devices, *IEEE Expert*, **6(6)**: 38-43
- Nguyen, T.A., Perkins, W.A., Laffey, T.J. & Pecora, D. (1987) Knowledge Base Verification, *AI Magazine*, Summer, 69-75.
- Nii, P. & Aiello, N. (1979) AGE: A knowledge-based program for building knowledge-based programs, *Proceedings of the 6th International Conference on Artificial Intelligence: IJCAI-79*, Information Processing Society of Japan. pp. 645-655.
- Noble, I.R. (1987) The role of expert systems in vegetation science, *Vegetatio* **69**: 115-121
- Norman, D.A. (1983) Some Observations on Mental Models, in: *Mental Models*, edited by D. Gentner & A.L. Stevens, Lawrence Erlbaum: Hillsdale, New Jersey. pp.7-14
- Norton, G.A. (1987) Developments in expert systems for pest management at Imperial college, UK, *Review of Marketing and Agricultural Economics* **55**: 167-170
- Oldham, R. and Nicholson, M. (1986) Status and Ecology of the Warty Newt (*Triturus cristatus*), *Final Report, Nature Conservancy Council, Contract No. HF 3/05/123, year 3*
- Olson, R.L., Wagner, T.L., & Willers, J.L. (1990) A Framework for Modelling Uncertain Reasoning in Ecosystem Management I. Background and Theoretical Considerations, *AI Applications in Natural Resource Management*, **4(4)**:1-10
- Olson, R.L., Willers, J.L., & Wagner, T.L. (1990) A Framework for modelling Uncertain Reasoning in Ecosystem Management II. Bayesian Belief Networks, *AI Applications in Natural Resource Management*, **4(4)**:1-10
- O'Neil, M., Glowinski, A. & Fox, J. (1989) A symbolic theory of decision-making applied to several medical tasks, *AIME '89: Proceedings of the European Conference on Artificial Intelligence in Medicine*, Springer-Verlag. pp.62-71
- Parsaye, K. & Chignell, M. (1988) *Expert Systems for Experts*, John Wiley & Sons: New York.
- Patil, R.S., Szolovits, P. & Schwartz, W.B. (1981) Causal Understanding of Patient Illness in Medical Diagnosis, *Proceedings of the 7th International Joint Conference on Artificial Intelligence (Volume 2): IJCAI-81*, William Kaufmann: Los Altos, Calif. pp.893-899

- Patten, B.C. (Ed.) (1971) *Systems Analysis and Simulation in Ecology: Volume 1*, Academic Press: London.
- Pauker, S.G. (1984) Discussion within "Statistical and Knowledge-based approaches to clinical decision-support systems, with an application in gatroenterology" by D.J. Spiegelhalter & R.P. Knill-Jones, *Journal of the Royal Statistical Society*, **B(147)**: 35-77
- Pearl, J. (1985) *How to do with probabilities what people say you can't*, Technical Report CSD-850031, Cognitive Systems Laboratory, Computer Science Dept, University of California.
- Pearl, J. (1986) On evidential reasoning in a hierarchy of hypotheses, *Artificial Intelligence*, **28**: 9-15
- Pianka, E.R. (1988) *Evolutionary Ecology (4th Edit.)*, Harper & Row, New York.
- Pople, H.E. (1982) Heuristic methods for imposing structure on ill-structured problems: The structuring of medical diagnosis, in P. Szolovits (Ed.) *Artificial Intelligence in Medicine*, AAAS Selected Symposium Series, West View: Boulder, Colorado. pp.159-176
- Popper, K.R. (1972) *Objective Knowledge: An Evolutionary Approach*, Clarendon: Oxford.
- Preece, A.D. (1990) Towards a methodology for evaluating expert systems, *Expert Systems*, **7(4)**: 215-223
- Prerau, D.S. (1985) selection of an Appropriate Domain for an Expert System, *Expert Systems*, **7(4)**: 215-233
- Price, C. & Lee, M. (1988) Applications in Deep Knowledge, *Artificial Intelligence in Engineering*, **3(1)**: 12-17
- Punch, W.F. (1992) Large interactions of compiled and causal reasoning in diagnosis, *IEEE Expert* **7(1)**: 28-35
- Rauscher, H.M., & Cooney, T.M. (1986) Using expert system technology in a forestry application: The CHAMPS experience, *Journal of Forestry* **84(3)**: 14-17
- Rice, D.A. & Nix, S.J. (1989) A simple rule-based system for species selection in artificial regeneration, *AI Applications in Natural Resource Management* **3(1)**: 38-43
- Rich, E. (1983) Default reasoning as likelihood reasoning, *Proceedings of the National Conference on Artificial Intelligence: AAAI-83*, William Kaufmann: Los Altos, Calif. pp.348-351
- Richards, J.N.J., Bez, H.E., Gittins, D.T., & Cooke, D.J. (1986) On Methods for Interface Specification and Design, *International Journal of Man-Machine Studies*, **24**: 545-568
- Rieger, C. & Grinberg, M. (1977) The Declarative Representation and Procedural Simulation of Causality in Physical Systems, in: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-77)*, pp.250-255
- Riese, C.E. & Zubrick, S.M. (1985) Using rule induction to combine declarative and procedural knowledge representations, *2nd Conference on Artificial Intelligence Applications*, IEEE Computer Society, pp.603-606

- Roach, J.W., Virkar, R.S., Weaver, M.J., & Drake, C.R. (1985) POMME: A computer-based consultation system for apple orchard management using PROLOG, *Expert Systems* 2(2): 56-69
- Robinson, J. (1977) A study of the newt colony on Castor Hanglands National Nature Reserve, Unpublished.
- Rollinger, C.R. (1983) How to represent evidence - aspects of uncertain reasoning, in: Bundy, A. (Ed.) *Proceedings of the 8th International Joint Conference on Artificial Intelligence: IJCAI-83, Volume 1*, William Kaufmann: Los Altos, California, pp.358-361
- Rosch, E. (1978) Principles of Categorization, in Rosch, E. & Lloyd, B.B. (eds) (1978) *Cognition and Categorisation*, Lawrence Erlbaum: Hillsdale, New Jersey. pp.27-48
- Rykiel, E.J. Jr (1989) Artificial Intelligence and Expert Systems in Ecology and Natural Resource Management, *Ecological Modelling*, 46: 3-8
- Saffioti, A. (1987) An AI view of the treatment of uncertainty, *Knowledge Engineering Review*, 2(2): 75-97
- Schmoldt, D.L. & Martin, G.L. (1989) Development and Evaluation of an expert system for diagnosing pest damage of red pine in Wisconsin, *Forest Science*, 35(2): 364-387
- Sembugamoorthy, V. & Chandrasekaran, B. (1986) Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems, in: *Experience, Memory and Reasoning*, Kolodner, J.L. & Riesbeck, C.K. (Eds) Lawrence Erlbaum, Hillsdale, NJ. pp.47-73
- Shafer, G. (1976) *A Mathematical Theory of Evidence*, Princetown University Press.
- Shapiro, R. (1986) *Origins*, Heinemann: London.
- Sharma, R.S. & Conrath, D.W. (1992) Evaluating expert systems: the socio-economic dimensions of quality, *Expert Systems* 9(3): 125-137
- Sheridan, F.K.J. (1991) A Survey of Techniques for Inference under Uncertainty, *Artificial Intelligence Review*, 5(1,2): 89-119
- Shibahara, T., Tsotsos, J.K., Mylopoulos, J., & Covvey, H.D. (1983) CAA: A knowledge-based system using causal knowledge to diagnose cardiac rhythm disorders, in: Bundy, A. (Ed.) *Proceedings of the 8th International Joint Conference on Artificial Intelligence, Volume 1: IJCAI-83*, William Kaufmann: Los Altos California, pp.242-245
- Shoham, Y. (1985) Reasoning about causation in knowledge-based systems, *2nd Conference on Artificial Intelligence Applications*, IEEE Computer Society, pp.629-634
- Shortliffe, E.H. (1976) *Computer-Based Medical Consultation: MYCIN*, New York: Elsevier.
- Shortliffe, E.H. & Buchanan, B.G. (1975) A Model of Inexact Reasoning in Medicine, *Mathematical Biosciences*, 23: 351-379
- Shugart, H.H. (1984) *A theory of forest dynamics: the ecological implications of forest succession models*, Springer-Verlag: New York.

- Simon, H.A. (1979) Information-Processing Theory of Human Problem Solving, in: W. Estes (ed.), *Handbook of Learning and Cognitive Processes (vol.5)*, Lawrence Erlbaum: Hillsdale, New Jersey.
- Skellam, J.G. (1972) Some philosophical aspects of mathematical modelling in empirical science with special reference to ecology, in: *Mathematical Models in Ecology*, by Jeffers, J.N.R. (Ed.), Blackwell: Oxford, pp. 13-28
- Slagle, J.R. & Wick, M.R. (1989) A Method for Evaluating Candidate Expert System Applications, *AI Applications in Natural Resource Management*, 3(2): 21-30
- Slatter, P.E. (1987a) *Building Expert Systems: Cognitive Emulation*, Ellis Horwood: Chichester.
- Slatter, P.E. (1987b) Cognitive Emulation in Expert System Design, *Knowledge Engineering Review*, 2(1): 27-41
- Smith, E.E. (1989) Concepts and Induction, in: M.I. Posner (ed.) *Foundations of Cognitive Science*, MIT Press: Cambridge, Massachusetts. pp.501-526
- Smith, E.E. & Median, D.L. (1981) *Categories and Concepts*, Harvard University Press: Cambridge, Massachusetts.
- Sousa, P.J. (1985) Habitat Suitability Index Models: Red Spotted Newt, in: *USDI Fisheries and Wildlife Services Biological Report 82 - 10.111*, Fort Collins, Colorado.
- Sowa, J.F. (1984) *Conceptual Structures*, Addison-Wesley: Reading, Massachusetts.
- Spiegelhalter, D.J. (1986) A Statistical View of Uncertainty in Expert Systems, in: *Artificial Intelligence and Statistics*, edited by W.A.Gale, Addison-Wesley, Reading, Massachusetts. pp.17-55
- Spiegelhalter, D.J. (1987) Probabilistic Expert Systems in Medicine: Practical Issues in Handling Uncertainty, *Statistical Science*, 2(1): 25-30.
- Starfield, A.M. & Bleloch, A.L. (1983) Expert Systems: An approach to problems in management that are difficult to quantify? *Journal of Environmental Management*, 16: 261-268
- Starfield, A.M. & Bleloch, A.L. (1986) *Building Models for Conservation and Wildlife Management*, Collier MacMillan: London.
- Starfield, A.M., Farm, B.P. & Taylor, R.H. (1989) A rule-based ecological model for the management of an estuarine system, *Ecological Modelling* 46: 107-119
- Steels, L. (1987a) The Deepening of Expert Systems, *AI Communications*, 1: 9-16 (August).
- Steels, L. (1987b) Second Generation Expert Systems, in; Bramer, M.A. (ed.) *Research & Development in Expert Systems III: Proceedings of the 6th Annual Tech. Conf. of the British Computer Society specialist group in Expert Systems*, Cambridge University Press, pp.175-183.
- Steels, L. (1990) Components of Expertise, *AI Magazine* 11(2): 28-49
- Steward, J.W. (1969) *The Tailed Amphibians of Europe*, David & Charles, Newton Abbott.



- Stock, M. (1988) Planning expert systems projects, *AI Applications in Natural Resource Management*, 2(4): 9-16.
- Stone, N.D., Coulson, R.N., Frisbie, R.E., & Loh, D.K. (1986) Expert Systems in Ecology: Three approaches to problem solving, *Bull. Entomol. Soc. Am.* Fall: 161-166
- Stone, N.D. & Schaub, L.P. (1990) A hybrid system/simulation model for the analysis of pest management strategies, *AI Applications in Natural Resource Management*, 4(2): 17-26
- Swan, M.J.S. & Oldham, R.S. (1990) Herptile Sites: Interim Report No.1, *Nature Conservancy Council Contract No. F72-15-04, Year 1.*
- Swartout, W.R. (1981) Explaining and Justifying Expert Consulting Programs, in: Drinan, A. (Ed.) *Proceedings of the 7th International Joint Conference on Artificial Intelligence: IJCAI-81*, William Kaufmann: Los Altos, Calif. pp.815-823.
- Szolovits, P. (1985) Types of knowledge as bases for reasoning in medical AI programs, in: DeLotto, I. & Stefanelli, M. (Eds) *Artificial Intelligence in Medicine*, Elsevier Science Publ., North-Holland. pp.209-226
- Thatch, D.W. & Schneider, P.J. (1988) Key ingredients in developing an on-line Knowledge-Base System for IPM, in: Zazueta, F.S. & Bottcher, A.B. (Eds), *Proceedings of the 2nd Int. Conf. on Computers in Agricultural Extension Programs Vol.1*, The Florida Cooperative Extension Service, Institute of Food & Agricultural Sciences. pp.202-208
- Thompson, R. (1959) *The Psychology of Thinking*, Penguin: Middlesex.
- Tong, C. (1991) The Nature and Significance of Knowledge Compilation, *IEEE Expert*, 6(2): 88-91
- Tonn, B.E., Goeltz, R. & Travis, C. (1992) Eliciting reliable uncertainty estimates, *Expert Systems*, 9(1): 25-33
- Travé-Massuyès, L. (1992) Qualitative Reasoning over time: history and current prospects, *The Knowledge Engineering Review*, 7(1): 1-18
- Tunncliffe, A.J. & Scrivener, S.A.R. (1991) Knowledge Elicitation in Design, *Design Studies*, 12(2): 73-80
- Tversky, A. & Kahneman, D. (1973) Judgements under uncertainty: heuristics and biases, *Science*, 185:1124-1131
- U.S. Fisheries & Wildlife Service (1981) Standards for the Development of habitat suitability index models, *USDI Fisheries & Wildlife Services 103-ESM*, Ecological Services, Washington D.C.
- Uschold, M., Robertson, D., Bundy, A. & Muetzelfeldt, R. (1989) Helping Inexperienced Users to Construct Simulation Programs: An Overview of the ECO project, in Vadera, S. (Ed.) *Expert Systems Applications*, Chichester: Sigma Press. pp.117-132
- van Someren, M.W., Zheng, L.L. & Post, W. (1990) Cases, models or compiled knowledge? A comparative analysis and proposed integration, in: *Current Trends in Knowledge Acquisition*, edited by B. Wielinga, J. Boose, B. Gaines, G. Schreiber, & M. van Someren, IOS Press: Amsterdam. pp.339-355
- Washbrook, J. & Keravnou, E. (1990) Making deepness explicit, *Artificial Intelligence in Medicine* 2(3): 129-134

- Waterman, D. (1986) *A Guide to Expert Systems*, Addison-Wesley, Reading: Massachusetts.
- Weigert, R.G. (1975) Simulation models of Ecosystems, *Annual Review of Ecology & Systematics*, 6: 311-338
- Weiss, S.M., Kulikowski, C.A., Amarel, S., & Safir, A. (1978) A model-based method for computer-aided medical decision-making, *Artificial Intelligence*, 11: 145-172
- Welbank, M. (1983) A review of knowledge acquisition techniques for expert systems, *Martlesham Consultancy Services, British Telecommunications, UK*.
- Whitehead, J.D. & Roach, J.W. (1990) Hoist: A second-generation expert system based on qualitative physics, *AI Magazine* 11(3): 108-119
- Wildlife and Countryside Act (1981), Her Majesty's Stationery Office
- Wildman, L.E. (1992) A model-based approach to the diagnosis of the cardiac arrhythmias, *Artificial Intelligence in Medicine*, 4(1): 1-19
- Wittgenstein, L. (1953) *Philosophical Investigations*, Translated by G. Anscombe, Macmillan: New York.
- Wyatt, J. & Spiegelhalter, D. (1990) Evaluating medical expert systems: what to test and how? *Medical Informatics*, 15(3): 205-217
- Zadeh, L.A. (1978) Fuzzy Sets as a Basis for a theory of Possibility, *Fuzzy Sets and Systems*, 1: 3-28
- Zadeh, L.A. (1983a) A Computational Approach to Fuzzy Quantifiers in Natural Languages, *Computer Mathematics Applications*, 9(1): 149-184
- Zadeh, L.A. (1983b) The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems, *Fuzzy Sets and Systems*, 11:199-228
- Zadeh, L.A. (1984) Review of Shafer's 'A Mathematical Theory of Evidence', *AI Magazine*, 5: 81-83
- Zadeh, L.A. (1986) Is Probability Theory sufficient for dealing with Uncertainty in AI: A Negative View, *Uncertainty in Artificial Intelligence*, Kanal, L.N. & Lemmer, J.F. (Eds), Elsevier Science Publ., North-Holland. pp.103-116
- Zimmer, A.C. (1986) What uncertainty judgements can tell about the underlying subjective probabilities, in: *Uncertainty in Artificial Intelligence*, Kanal, L.N. & Lemmer, J.F. (Eds), Elsevier Science Publ., North-Holland. pp.249-258
- Zuiderwijk, A. (1984) Hoe redden we de Kamsalamander, *De Levende Natuur* 85: 67-71

**APPENDIX A**  
**SUMMARY OF EVALUATION DATA**

This is a log of the data recorded and analysed in the assessment of HEX and TRITON by two user groups; students (55 individuals examined HEX, 20 of the original 55 examined TRITON) and experts/near experts (15 individuals).

Not all questions were addressed at both groups. Specifically, question (1) relating to the pond assessment form was only given to the students, as only this group could be impelled to go out and perform an evaluation at a site. Conversely, the students were not asked to assess the accuracy of the systems, the systems' questioning strategy, and the robustness of these systems, as it was felt they would not have a suitable background for these tasks.

This appendix falls into 6 main parts;

- 1) Data gathered from student and peer expert evaluations, with appropriate statistical analysis.
- 2) Comparison of HEX and TRITON overall performance, as expressed by the student and expert groups.
- 3) Comparison of examples of 'default' explanations of HEX and TRITON
- 4) A comparison of actual data from 100 ponds with the assessment of expert, HEX and TRITON
- 5) Correctness of assessment of a specific pond by the student group
- 6) Details of noted errors, criticisms, and suggestions.

Part 1: Data Gathered from Student and Peer Expert Evaluations, with Appropriate Statistical Analysis

This part contains details of the gathered data from the standard questionnaires (see Appendix G). Each question is presented followed by data giving details of answers. The results of analysis are then presented, with significant results (ie  $p\text{-value} \leq 0.05$ ) are emboldened.

Three particular statistical tests were frequently used in analysing this data (Agresti, 1990). These were;

a) A test (looking at student and expert groups separately, and then combined) to assess the null hypothesis that there is no significant difference in users' rating of HEX or TRITON (NB: Only individuals that expressed some preference were considered). This within-person comparison comprised of a test based on a binomial distribution that assumed the relative proportion of each set of individuals would be 0.5 (ie the null hypothesis was that half of the individuals expressing a higher rating for one system would prefer HEX, whilst the remaining half would prefer TRITON).

b) A test to assess whether there was a significant difference in the proportions of those individuals showing a higher rating for HEX or TRITON between the student and expert groups. This only applied to parameters which were examined by both groups. The tests used were based around a contingency table, which was drawn up using those individuals that expressed a higher rating;

Rated more highly	HEX	TRITON
Students	[ A ]	[ B ]
Experts	[ B ]	[ C ]

This table was analysed using appropriate tests. Chi-squared test was used where data was suitable, and Fisher's exact test where data was not suitable for chi-squared tests.

c) A test to assess whether the order in which the experts were asked to examine the systems (HEX followed by TRITON, or TRITON followed by HEX) affected their choices. The test used in this case was Gart's test, based around Fisher's exact test.

Significant values occurred in questions 4a (clarity of explanations), 4d (the utility of information given by the system) and 8a (overall questioning strategy). The data of question 7 (robustness) is almost significant at the 5% level.

**(1) Pond Assessment Form**

Was it easy to fill in the questionnaire provided to store information about the pond?

	Students
Definitely	[17]
Mostly	[25]
For a reasonable part	[11]
Only some of the time	[ 2]
Not at all	[ 0]

**(2) Handbook**

Did the handbook used in conjunction with the HEX/TRITON system give a good overview and introduction to HEX/TRITON?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[31]	[16]	[ 8]	[ 9]
Mostly	[21]	[ 4]	[ 6]	[ 6]
For a reasonable part	[ 2]	[ 0]	[ 1]	[ 0]
Only some of the time	[ 0]	[ 0]	[ 0]	[ 0]
Not at all	[ 0]	[ 0]	[ 0]	[ 0]

Rated more highly:	HEX	TRITON	Neither
Students	[ 2]	[ 6]	[ 12]
Experts (HEX 1st)	[ 1]	[ 2]	[ 8]
Experts (TRITON 1st)	[ 1]	[ 0]	[ 3]
Experts (Total)	[ 2]	[ 2]	[ 11]
Combined	[ 4]	[ 8]	[ 23]

**Analysis:**

(a) Rating ( $H_0$ ; no significant difference between HEX and TRITON)

	Students	Experts	Combined
p-value ( $H_0$ )	0.145	0.6875	0.194

(b) Student/Expert Ratings: p-value ( $H_0$ ; no difference between student/expert ratios) = 0.34

(c) Ordering Effect: p-value( $H_0$ ; no difference in rating caused by ordering) = 0.5

### (3) User Interface

(3a) Do you feel that the various screen layouts you encountered were acceptable?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[37]	[13]	[ 7]	[ 6]
Mostly	[17]	[ 7]	[ 7]	[ 7]
For a reasonable part	[ 1]	[ 0]	[ 1]	[ 2]
Only some of the time	[ 0]	[ 0]	[ 0]	[ 0]
Not at all	[ 0]	[ 0]	[ 0]	[ 0]

(3b) Do you feel it was easy to examine the information within HEX/TRITON?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[38]	[16]	[10]	[ 7]
Mostly	[16]	[ 3]	[ 5]	[ 7]
For a reasonable part	[ 1]	[ 0]	[ 0]	[ 1]
Only some of the time	[ 0]	[ 0]	[ 0]	[ 0]
Not at all	[ 0]	[ 0]	[ 0]	[ 0]
(Unmarked:	[ ]	[ 1]	[ ]	[ ])

(3c) Was the HEX/TRITON system easy to learn?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[45]	[15]	[13]	[11]
Mostly	[ 8]	[ 4]	[ 2]	[ 3]
For a reasonable part	[ 2]	[ 0]	[ 0]	[ 1]
Only some of the time	[ 0]	[ 0]	[ 0]	[ 0]
Not at all	[ 0]	[ 0]	[ 0]	[ 0]
(Unmarked:	[ ]	[ 1]	[ ]	[ ])

(3d) Did HEX/TRITON require a high degree of concentration to use?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[ 2]	[ 0]	[ 1]	[ 1]
Mostly	[12]	[ 4]	[ 4]	[ 6]
For a reasonable part	[23]	[ 8]	[ 6]	[ 5]
Only some of the time	[12]	[ 6]	[ 1]	[ 1]
Not at all	[ 6]	[ 2]	[ 0]	[ 0]

Note that (3a-3d) covers different aspects of the user interface. The answers were combined by attaching numeric values to answers, so that 'Definitely' is equivalent to 5, 'mostly' is 4, etc (the reverse occurred for 3d, as the answer of 'definitely' indicated a unfavourable answer about the user interface). These numeric values were then summed, and the number was taken as a rating score.

The summary of questions (3a) to (3d) is given below:

Rated more highly:	HEX	TRITON	Neither
Students	[ 5 ]	[ 5 ]	[ 10 ]
Experts (HEX 1st)	[ 6 ]	[ 2 ]	[ 3 ]
Experts (TRITON 1st)	[ 1 ]	[ 2 ]	[ 1 ]
Experts (Total)	[ 7 ]	[ 4 ]	[ 4 ]
Combined	[ 12 ]	[ 9 ]	[ 14 ]

**Analysis:**

(a) Rating ( $H_0$ ; no difference between HEX and TRITON)

	Students	Experts	Combined
p-value ( $H_0$ )	0.623	0.274	0.332

(b) Student/Expert Ratings: p-value ( $H_0$ ; no difference between student/expert ratios) = 0.283

(c) Ordering Effect: p-value ( $H_0$ ; no difference in rating caused by ordering) = 0.255

(3e) Would you rank the overall 'usability' of HEX/TRITON as high?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[26]	[11]	[ 7 ]	[ 6 ]
Mostly	[25]	[ 8 ]	[ 5 ]	[ 8 ]
For a reasonable part	[ 3 ]	[ 1 ]	[ 2 ]	[ 0 ]
Only some of the time	[ 1 ]	[ 0 ]	[ 1 ]	[ 1 ]
Not at all	[ 0 ]	[ 0 ]	[ 0 ]	[ 0 ]

Rated more highly:	HEX	TRITON	Neither
Students	[ 3 ]	[ 5 ]	[ 12 ]
Experts (HEX 1st)	[ 3 ]	[ 2 ]	[ 6 ]
Experts (TRITON 1st)	[ 1 ]	[ 0 ]	[ 3 ]
Experts (Total)	[ 4 ]	[ 2 ]	[ 9 ]
Combined	[ 7 ]	[ 7 ]	[ 21 ]

**Analysis:**

(a) Rating ( $H_0$ ; no difference between HEX and TRITON)

	Students	Experts	Combined
p-value ( $H_0$ )	0.363	0.344	0.605

(b) Student/Expert Ratings: p-value ( $H_0$ ; no difference between student/expert ratios) = 0.245

(c) Ordering Effect: p( $H_0$ ; no difference in rating caused by ordering) = 0.667



**(4) Justification/Explanation Facilities**

(4a) Did you find the justifications of questions (The 'Explain?' facility), when you used them, were clear and understandable?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[ 35 ]	[ 9 ]	[ 7 ]	[ 5 ]
Mostly	[ 19 ]	[ 9 ]	[ 5 ]	[ 7 ]
For a reasonable part	[ 1 ]	[ 1 ]	[ 3 ]	[ 3 ]
Only some of the time	[ 0 ]	[ 1 ]	[ 0 ]	[ 0 ]
Not at all	[ 0 ]	[ 0 ]	[ 0 ]	[ 0 ]

Rated more highly:	HEX	TRITON	Neither
Students	[ 7 ]	[ 3 ]	[ 10 ]
Experts (HEX 1st)	[ 1 ]	[ 4 ]	[ 6 ]
Experts (TRITON 1st)	[ 4 ]	[ 0 ]	[ 0 ]
Experts (Total)	[ 5 ]	[ 4 ]	[ 6 ]
Combined	[ 12 ]	[ 7 ]	[ 16 ]

**Analysis:**

(a) Rating ( $H_0$ ; no difference between HEX and TRITON)

	Students	Experts	Combined
p-value ( $H_0$ )	0.172	0.5	0.180

(b) Student/Expert Ratings: p-value ( $H_0$ ; no difference between student/expert ratios) = 0.300

(c) Ordering Effect: p( $H_0$ ; no difference in rating caused by ordering) = 0.039

(4b) Were the details sometimes given to the right of your list of choices/in '<F2> Define' selection for each question useful in making your choice?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[ 21 ]	[ 7 ]	[ 4 ]	[ 5 ]
Mostly	[ 21 ]	[ 11 ]	[ 10 ]	[ 8 ]
For a reasonable part	[ 9 ]	[ 1 ]	[ 0 ]	[ 0 ]
Only some of the time	[ 4 ]	[ 1 ]	[ 0 ]	[ 1 ]
Not at all	[ 0 ]	[ 0 ]	[ 0 ]	[ 0 ]

Rated more highly:	HEX	TRITON	Neither
Students	[ 8 ]	[ 4 ]	[ 8 ]
Experts (HEX 1st)	[ 3 ]	[ 3 ]	[ 5 ]
Experts (TRITON 1st)	[ 1 ]	[ 2 ]	[ 1 ]
Experts (Total)	[ 4 ]	[ 5 ]	[ 6 ]
Combined	[ 12 ]	[ 9 ]	[ 14 ]

**Analysis:**

(a) Rating ( $H_0$ ; no difference between HEX and TRITON)

	Students	Experts	Combined
p-value ( $H_0$ )	0.194	0.5	0.332

(b) Student/Expert Ratings: p-value ( $H_0$ ; no difference between student/expert ratios) = 0.212

(c) Ordering Effect: p( $H_0$ ; no difference in rating caused by ordering) = 0.476

(4c) Did you find the final explanations of pond suitability or non-suitability for the crested newt clear and understandable?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[25]	[11]	[ 8]	[ 8]
Mostly	[18]	[ 6]	[ 6]	[ 6]
For a reasonable part	[ 8]	[ 3]	[ 0]	[ 1]
Only some of the time	[ 2]	[ 0]	[ 1]	[ 0]
Not at all	[ 2]	[ 0]	[ 0]	[ 0]

Rated more highly:	HEX	TRITON	Neither
Students	[ 6]	[ 4]	[ 10]
Experts (HEX 1st)	[ 2]	[ 4]	[ 5]
Experts (TRITON 1st)	[ 2]	[ 1]	[ 1]
Experts (Total)	[ 4]	[ 5]	[ 6]
Combined	[ 10]	[ 9]	[ 16]

**Analysis:**

(a) Rating ( $H_0$ ; no difference between HEX and TRITON)

	Students	Experts	Combined
p-value ( $H_0$ )	0.377	0.5	0.5

(b) Student/Expert Ratings: p-value ( $H_0$ ; no difference between student/expert ratios) = 0.286

(c) Ordering Effect: p( $H_0$ ; no difference in rating caused by ordering) = 0.357

(4d) Do you feel that the overall system gave useful information?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[24]	[ 7]	[ 6]	[ 3]
Mostly	[23]	[12]	[ 7]	[ 8]
For a reasonable part	[ 7]	[ 1]	[ 2]	[ 3]
Only some of the time	[ 1]	[ 0]	[ 0]	[ 1]
Not at all	[ 0]	[ 0]	[ 0]	[ 0]

Rated more highly:	HEX	TRITON	Neither
Students	[ 6]	[ 4]	[ 10]
Experts (HEX 1st)	[ 5]	[ 1]	[ 5]
Experts (TRITON 1st)	[ 2]	[ 0]	[ 2]
Experts (Total)	[ 7]	[ 1]	[ 7]
Combined	[ 13]	[ 5]	[ 17]

**Analysis:**

(a) Rating ( $H_0$ ; no difference between HEX and TRITON)

	Students	Experts	Combined
p-value ( $H_0$ )	0.377	0.035	0.048

(b) Student/Expert Ratings: p-value ( $H_0$ ; no difference between student/expert ratios) = 0.196

(c) Ordering Effect: p( $H_0$ ; no difference in rating caused by ordering) = 0.75

(4e) Do you feel the level of detail was suitable?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[ 11 ]	[ 8 ]	[ 5 ]	[ 4 ]
Mostly	[ 34 ]	[ 7 ]	[ 8 ]	[ 9 ]
For a reasonable part	[ 9 ]	[ 4 ]	[ 1 ]	[ 2 ]
Only some of the time	[ 1 ]	[ 1 ]	[ 1 ]	[ 0 ]
Not at all	[ 0 ]	[ 0 ]	[ 0 ]	[ 0 ]

Rated more highly:	HEX	TRITON	Neither
Students	[ 4 ]	[ 7 ]	[ 9 ]
Experts (HEX 1st)	[ 5 ]	[ 3 ]	[ 3 ]
Experts (TRITON 1st)	[ 1 ]	[ 1 ]	[ 2 ]
Experts (Total)	[ 6 ]	[ 4 ]	[ 5 ]
Combined	[ 10 ]	[ 11 ]	[ 14 ]

**Analysis:**

(a) Rating ( $H_0$ ; no difference between HEX and TRITON)

	Students	Experts	Combined
p-value ( $H_0$ )	0.274	0.377	0.5

(b) Student/Expert Ratings: p-value ( $H_0$ ; no difference between student/expert ratios) = 0.196

(c) Ordering Effect: p( $H_0$ ; no difference in rating caused by ordering) = 0.533

**(5) Education**

Do you feel you have gained any insight into the pond ecosystem whilst using HEX/TRITON?

	Students		Experts	
	HEX	TRITON	HEX	TRITON
Definitely	[ 16 ]	[ 8 ]	[ 5 ]	[ 3 ]
Mostly	[ 13 ]	[ 6 ]	[ 2 ]	[ 3 ]
For a reasonable part	[ 19 ]	[ 4 ]	[ 2 ]	[ 2 ]
Only some of the time	[ 7 ]	[ 3 ]	[ 3 ]	[ 5 ]
Not at all	[ 0 ]	[ 0 ]	[ 2 ]	[ 2 ]

Rated more highly:	HEX	TRITON	Neither
Students	[ 5 ]	[ 5 ]	[ 10 ]
Experts (HEX 1st)	[ 0 ]	[ 1 ]	[ 10 ]
Experts (TRITON 1st)	[ 3 ]	[ 0 ]	[ 1 ]
Experts (Total)	[ 3 ]	[ 1 ]	[ 11 ]
Combined	[ 8 ]	[ 6 ]	[ 21 ]

**Analysis:**

(a) Rating ( $H_0$ ; no difference between HEX and TRITON)

	Students	Experts	Combined
p-value ( $H_0$ )	0.623	0.313	0.395

(b) Student/Expert Ratings: p-value ( $H_0$ ; no difference between student/expert ratios) = 0.336

(c) Ordering Effect: p( $H_0$ ; no difference in rating caused by ordering) = 0.25

**(6) Accuracy (Experts only)**

(6a) Do you think that the assessment of whether the pond was suitable or unsuitable was acceptably accurate?

	HEX	TRITON
Yes	[ 5]	[ 7]
Yes, with notable exceptions	[ 9]	[ 7]
No	[ 1]	[ 1]

Rating	HEX	TRITON	Neither
Experts (HEX 1st)	[ 2]	[ 4]	[ 6]
Experts (TRITON 1st)	[ 1]	[ 0]	[ 3]
Experts (Total)	[ 3]	[ 4]	[ 9]

**Analysis:**

(a) Rating of experts:  $p\text{-value}(H_0; \text{no difference between HEX and TRITON}) = 0.5.$

(b) Ordering Effect:  $p(H_0; \text{no difference in rating caused by ordering}) = 0.429$

(6a.ii) (TRITON ONLY) Do you think the strength of conviction that the system expressed in the conclusion (eg 'strong evidence for...', 'some evidence for...', etc) was correct:

	TRITON
Yes	[ 9]
Yes, with few exceptions	[ 3]
No, overconfident in conviction	[ 1]
No, too weak in conviction	[ 1]
No, too strong and too weak at different times	[ 1]
Don't know	[ 0]

(6b) Do you think the explanations of the final conclusions were correct?

	HEX	TRITON
Yes	[ 9]	[ 7]
Yes, with notable exceptions	[ 6]	[ 8]
No	[ 0]	[ 0]
Didn't understand them	[ 0]	[ 0]
Don't know	[ 0]	[ 0]

Rated more highly:	HEX	TRITON	Neither
Experts (HEX 1st)	[ 3]	[ 4]	[ 4]
Experts (TRITON 1st)	[ 2]	[ 1]	[ 1]
Experts (Total)	[ 5]	[ 5]	[ 5]

**Analysis:**

(a) Rating of experts:  $p\text{-value}(H_0; \text{no difference between HEX and TRITON}) = 0.623$

(b) Ordering Effect:  $p(H_0; \text{no difference in rating caused by ordering}) = 0.417$

(6c) Do you think the explanations of the final conclusions were clear?

	HEX	TRITON
Yes	[12]	[12]
Yes, with notable exceptions	[ 3]	[ 3]
No	[ 0]	[ 0]
Didn't understand them	[ 0]	[ 0]
Don't know	[ 0]	[ 0]

Rating	HEX	TRITON	Neither
Experts (HEX 1st)	[ 0]	[ 0]	[ 11]
Experts (TRITON 1st)	[ 1]	[ 0]	[ 3]
Experts (Total)	[ 1]	[ 0]	[ 14]

This data is not in sufficient quantity to analyse. It is worth noting, however, the large number of individuals who do not show a higher rating of HEX or TRITON.

Part 3 in this appendix has further details of data/analysis of aspects concerning explanation.

**(7) Robustness (Experts only)**

Do you feel that the system is able to handle a wide variation of cases?

	HEX	TRITON
Yes	[ 6]	[10]
Yes, with notable exceptions	[ 6]	[ 3]
No	[ 2]	[ 1]
Don't know	[ 1]	[ 1]

Rated more highly:	HEX	TRITON	Neither
Experts (HEX 1st)	[ 0]	[ 6]	[ 5]
Experts (TRITON 1st)	[ 1]	[ 0]	[ 3]
Experts (Total)	[ 1]	[ 6]	[ 8]

**Analysis:**

(a) Rating of experts:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.063 (Though not significant, the data is noteworthy, indicating a definite bias towards TRITON).

(b) Ordering Effect:  $p(H_0$ ; no difference in rating caused by ordering) = 0.143

**(8) The Suitability of the Questioning Strategy (Experts only)**

(8a) Do you feel that the overall ordering of questions is acceptable for gathering information for the evaluation of a pond in terms of suitability for the crested newt?

	HEX	TRITON
Yes	[13]	[ 8]
Yes, with notable exceptions	[ 2]	[ 5]
No	[ 0]	[ 2]
Don't know	[ 0]	[ 0]

Rated more highly:	HEX	TRITON	Neither
Experts (HEX 1st)	[ 6]	[ 1]	[ 4]
Experts (TRITON 1st)	[ 1]	[ 0]	[ 3]
Experts (Total)	[ 7]	[ 1]	[ 7]

**Analysis:**

(a) Rating of experts:  $p\text{-value}(H_0; \text{no difference between HEX and TRITON}) = 0.0352$

(b) Ordering Effect:  $p(H_0; \text{no difference in rating caused by ordering}) = 0.875$

(8b) Do you feel all of the questions are worded in an understandable way?

	HEX	TRITON
Yes	[ 8]	[11]
Yes, with notable exceptions	[ 7]	[ 4]
No	[ 0]	[ 0]

Rated more highly:	HEX	TRITON	Neither
Experts (HEX 1st)	[ 0]	[ 4]	[ 7]
Experts (TRITON 1st)	[ 2]	[ 0]	[ 2]
Experts (Total)	[ 2]	[ 4]	[ 9]

**Analysis:**

(a) Rating of experts:  $p\text{-value}(H_0; \text{no difference between HEX and TRITON}) = 0.348$

(b) Ordering Effect:  $p(H_0; \text{no difference in rating caused by ordering}) = 0.0667$

(8c) Do you think that all of the questions are relevant (to pond evaluation with respect to the crested newt)?

	HEX	TRITON
Yes	[11]	[12]
Yes, with notable exceptions	[ 4]	[ 3]
No	[ 0]	[ 0]

Rating:	HEX	TRITON	Neither
Experts (HEX 1st)	[ 0]	[ 1]	[ 10]
Experts (TRITON 1st)	[ 1]	[ 1]	[ 2]
Experts (Total)	[ 1]	[ 2]	[ 12]

**Analysis:**

(a) Rating of experts:  $p\text{-value}(H_0; \text{no difference between HEX and TRITON}) = 0.5$

(b) Ordering Effect:  $p(H_0; \text{no difference in rating caused by ordering}) = 0.667.$

Part 2: A Comparison of HEX and TRITON Overall Performance, as expressed by Student and Expert Groups

In order to generate and 'overall' evaluation of these systems by users, answers to evaluation questions were allocated numeric values. The answers suggesting the greatest user satisfaction were designated the highest value (eg if a question has 5 possible answers, the answer denoting the highest level of user satisfaction was allocated '5'; the lowest allocated '1').

Both student and expert groups were presented with 12 questions (Sections 2-5 of part 1 of this appendix), with each question having 5 possible answers. For each student user, The numeric values of answers were added, giving a range of possible accumulated values between 12 (indicating minimum user satisfaction) to 60 (indicating maximum user satisfaction). This was done for responses to both HEX and TRITON. Using these scores, it was possible to statistically compare the responses of individual students to both HEX and TRITON. It was also possible to generate and compare overall 'user satisfaction' distributions of student and expert evaluators.

In addition to the 12 questions already mentioned, the experts were presented with a further 7 questions (about accuracy, robustness, and questioning strategy). The answers to these questions were allocated numbers in the same way as previously, but in this case the number of possible answers to each question was variable (between three and five). This second measure had a range of 7 to 26. This value was added to the measures of the previous 12 questions (giving a range of 19 to 86), and this score was used in comparing individual expert's assessments of HEX and TRITON.

1) Comparison between HEX and TRITON as paired data - Students

20 Students in all looked at both systems. One student did not fill in the questionnaire fully, leaving the 19 data pairs. On analysis of the data, the following statistics were generated;

	N	MEAN	MEDIAN	MODE
HEX data	19	52.105	52.000	51.00
TRITON data	19	52.316	51.000	51.00

Using Wilcoxon matched pairs signed rank sum test (a non-parametric test for comparison of paired data), it was found that there was no significant difference between the data sets ( $p$ -value  $> 0.5$ ).

2) Comparison between HEX and TRITON as paired data - Experts

15 experts looked at both systems, for 19 questions in all (the range of possible values went from 19 to 86). One expert did not fill in the questionnaire fully, leaving 14 data-pairs. The following statistics were generated;

	N	MEAN	MEDIAN	MODE
HEX data	14	73.21	73.50	71
TRITON data	14	71.57	69.00	67

Using the Wilcoxon matched pairs signed rank sum test, it was found that there was no significant difference between these data sets ( $p$ -value  $> 0.25$ ).

3) Comparison of Student and Expert Evaluations of HEX

In this case, 55 students and 14 experts gave analysable results. On analysis of the data, the following statistics were generated;

	N	MEAN	MEDIAN	MODE
Student data	55	51.073	51.000	51.00
Expert data	14	50.29	50.00	48.00

The Mann-Whitney test (assuming non-parametric data) was used to test the null hypothesis that there was no difference between these samples. There was found to be no significant difference between these sets of data ( $p$ -value  $> 0.4$ ).



#### 4) Comparison of Student and Expert Evaluations of TRITON

In this case, 19 students and 15 experts gave analysable results. On analysis of this data, the following statistics were generated;

	N	MEAN	MEDIAN	MODE
Student Data	19	52.316	51.000	51.00
Expert Data	15	48.60	47.00	45.00

Using the Mann-Whitney test, there was found to be a significant (but not highly significant) difference between these data sets ( $0.05 > p\text{-value} > 0.01$ ).

This indicates that the student group submits a higher 'rating' of user satisfaction to TRITON than does the expert group. However, the difference is not highly significant.

Part 3: Experts' comparison of examples of 'default' explanations of HEX and TRITON

The group of 15 experts were asked to compare different explanations along various criteria (comprehensibility, clarity, etc). These explanations were equivalent to the 'default' explanation of HEX and TRITON.

The experts were given three sets of questions, concerned with the relationship of the crested newt and (i) location, (ii) pond size/area, and (iii) percentage of pond surface covered by emergent vegetation. The explanations are stated fully in Appendix G. Statistical tests on data used a binomial distribution examining rating, with the null hypothesis being that HEX and TRITON explanations would each be rated more highly by half the population.

Note that;

H1: response to default HEX explanation concerning location.

H2: response to default HEX explanation concerning area.

H3: response to default HEX explanation concerning surface area covered.

T1: response to default TRITON explanation concerning location.

T2: response to default TRITON explanation concerning area.

T3: response to default TRITON explanation concerning surface area covered.

i) **Comprehensibility:** From the details given in the explanations, do you understand how the pond's location/area/% of emergent vegetation and the pond's suitability to support a viable crested newt colony are related?

	H1/T1	H2/T2	H3/T3
Definitely	[ 0/10]	[ 0/10]	[ 0/ 9]
Mostly	[ 2/ 4]	[ 5/ 4]	[ 4/ 5]
For a reasonable part	[ 4/ 1]	[ 3/ 0]	[ 2/ 1]
Partially	[ 4/ 0]	[ 1/ 1]	[ 5/ 0]
Not at all	[ 5/ 0]	[ 6/ 0]	[ 4/ 0]

Rating:	HEX	TRITON	Neither
H1/T1	[ 0]	[ 15]	[ 0]
H2/T2	[ 1]	[ 13]	[ 1]
H3/T3	[ 0]	[ 14]	[ 1]

Example 1:  $p\text{-value}(H_0; \text{no difference between HEX and TRITON}) = 0.000031$

Example 2:  $p\text{-value}(H_0; \text{no difference between HEX and TRITON}) = 0.00092$

Example 3:  $p\text{-value}(H_0; \text{no difference between HEX and TRITON}) = 0.000061$

ii) **Clarity:** How clearly do you think each of these explanations expresses the relationship between the pond's location/area/% of emergent vegetation and the pond's suitability to support a viable crested newt colony?

	H1/T1	H2/T2	H3/T3
Extremely clear	[ 1/ 7]	[ 0/ 4]	[ 0/ 4]
Very clear	[ 0/ 7]	[ 0/ 6]	[ 2/ 6]
Reasonably clear	[ 4/ 0]	[ 5/ 5]	[ 5/ 4]
Not very clear	[ 7/ 1]	[ 5/ 0]	[ 3/ 1]
Not clear at all	[ 3/ 0]	[ 5/ 0]	[ 5/ 0]

Rating:	HEX	TRITON	Neither
H1/T1	[ 0]	[ 13]	[ 2]
H2/T2	[ 0]	[ 14]	[ 1]
H3/T3	[ 2]	[ 11]	[ 1]

Example 1:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.00012

Example 2:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.000061

Example 3:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.0112

iii) **Accuracy:** Do you consider that these explanations are accurate (in considering the pond's location/area/% of emergent vegetation and pond suitability to support crested newts)?

	H1/T1	H2/T2	H3/T3
Definitely	[ 0/ 2]	[ 1/ 2]	[ 0/ 2]
Mostly	[ 8/ 9]	[ 5/ 7]	[ 5/ 5]
For a reasonable part	[ 2/ 1]	[ 2/ 3]	[ 4/ 6]
Partially	[ 4/ 2]	[ 3/ 2]	[ 4/ 2]
Not at all	[ 0/ 0]	[ 2/ 0]	[ 2/ 0]
Don't know	[ 1/ 1]	[ 2/ 1]	[ 0/ 0]

Rating:	HEX	TRITON	Neither
H1/T1	[ 2]	[ 7]	[ 6]
H2/T2	[ 1]	[ 8]	[ 6]
H3/T3	[ 1]	[ 7]	[ 7]

Example 1:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.0898

Example 2:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.0195

Example 3:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.0352

iv) **Utility:** How would you rank the utility of the information given by each of the explanations for educational and practical users?

#### Educational Users

	H1/T1	H2/T2	H3/T3
Definitely useful	[ 0/ 8]	[ 0/ 7]	[ 0/ 3]
Mainly useful	[ 0/ 6]	[ 1/ 6]	[ 2/ 7]
Reasonably useful	[ 5/ 0]	[ 5/ 1]	[ 4/ 3]
Only partly useful	[ 5/ 1]	[ 4/ 0]	[ 5/ 2]
Not at all useful	[ 5/ 0]	[ 4/ 0]	[ 4/ 0]
Don't know	[ 0/ 0]	[ 1/ 0]	[ 1/ 1]

Rating:	HEX	TRITON	Neither
H1/T1	[ 0 ]	[ 13 ]	[ 2 ]
H2/T2	[ 0 ]	[ 12 ]	[ 3 ]
H3/T3	[ 1 ]	[ 10 ]	[ 4 ]

Example 1:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.00012

Example 2:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.00024

Example 3:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.0059

### Practical Users

	H1/T1	H2/T2	H3/T3
Definitely useful	[ 3/ 7 ]	[ 1/ 5 ]	[ 0/ 5 ]
Mainly useful	[ 3/ 6 ]	[ 4/10 ]	[ 5/ 6 ]
Reasonably useful	[ 5/ 2 ]	[ 6/ 0 ]	[ 3/ 2 ]
Only partly useful	[ 4/ 0 ]	[ 2/ 0 ]	[ 6/ 2 ]
Not at all useful	[ 0/ 0 ]	[ 2/ 0 ]	[ 1/ 0 ]

Rating:	HEX	TRITON	Neither
H1/T1	[ 1 ]	[ 12 ]	[ 2 ]
H2/T2	[ 0 ]	[ 11 ]	[ 4 ]
H3/T3	[ 0 ]	[ 10 ]	[ 5 ]

Example 1:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.0017

Example 2:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.00049

Example 3:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.00098

v) **Detail:** Do you think the explanations are of sufficient detail to enable *educational* and *practical* users to understand the complexities of the relationship between the pond's location/area/% of emergent vegetation and pond suitability to support crested newts?

### Educational Users

	H1/T1	H2/T2	H3/T3
Definitely	[ 0/ 3 ]	[ 0/ 3 ]	[ 0/ 2 ]
Mostly	[ 1/ 8 ]	[ 2/ 8 ]	[ 2/ 6 ]
For a reasonable part	[ 3/ 3 ]	[ 2/ 3 ]	[ 4/ 4 ]
Partially	[ 5/ 1 ]	[ 5/ 1 ]	[ 4/ 2 ]
Not at all	[ 6/ 0 ]	[ 6/ 0 ]	[ 5/ 1 ]

Rating:	HEX	TRITON	Neither
H1/T1	[ 0 ]	[ 13 ]	[ 2 ]
H2/T2	[ 1 ]	[ 13 ]	[ 1 ]
H3/T3	[ 3 ]	[ 9 ]	[ 3 ]

Example 1:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.00012

Example 2:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.00092

Example 3:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.073

**Practical Users**

	H1/T1	H2/T2	H3/T3
Definitely	[ 0/ 5]	[ 0/ 5]	[ 0/ 3]
Mostly	[ 4/ 7]	[ 4/ 7]	[ 3/ 7]
For a reasonable part	[ 3/ 2]	[ 3/ 1]	[ 4/ 3]
Partially	[ 4/ 1]	[ 5/ 2]	[ 5/ 2]
Not at all	[ 4/ 0]	[ 3/ 0]	[ 3/ 0]

Rating:	HEX	TRITON	Neither
H1/T1	[ 0]	[ 13]	[ 2]
H2/T2	[ 1]	[ 12]	[ 2]
H3/T3	[ 1]	[ 10]	[ 4]

Example 1:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.00012

Example 2:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.0019

Example 3:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.0059

vi) **Preference:** Which explanation do you prefer?

	HEX	TRITON	Neither
Example 1	[ 1]	[12]	[ 2]
Example 2	[ 1]	[14]	[ 0]
Example 3	[ 1]	[13]	[ 1]

Example 1:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.0017

Example 2:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.00049

Example 3:  $p$ -value( $H_0$ ; no difference between HEX and TRITON) = 0.00091

Part 4: A Comparison of Actual Data from 100 ponds with  
the Assessment of the Domain Expert, HEX and TRITON

Data was gathered from a reliable, independent source for several hundred ponds. Within this data, a set of ponds was selected to be presented to; the expert; HEX, and; TRITON. The criteria of selection was that the data from each pond should;

(i) Contain a definite record of whether crested newts were present or absent.

(ii) Contain a reasonable indication of whether fish were present or absent (This was included as a defining criterion, as many of the ponds had no information about fish presence. To use such data would have meant many of the conclusions made by the expert would very frequently have a low 'conviction' attached, caused by this single, common, and important parameter).

This left 140 ponds. From these, 100 were selected at random, and the details of each were;

(a) given to Dr Rob Oldham (the domain expert), for commentary about suitability for the crested newt, a statement of certainty in his conclusion, and a statement about main parameters used in making a decision. Whilst not prompted to do so, he explicitly stated that he tended to answer conservatively (ie tended to use restraint in stating a certainty in his conclusions).

(b) processed through HEX to give conclusions about suitability for the crested newt, and give main parameters of decision.

(c) processed through TRITON, to give the same details as gathered from HEX - in addition, some measure of certainty in its conclusions was given by TRITON.

a) Comparison of Actual data and Domain Expert's Conclusions

Using the data gathered, the following table was constructed:

		Conclusions of Domain Expert	
		Suitable	Unsuitable
Presence of crested newts in pond	Present	15	25
	Absent	3	57

Using appropriate analysis (Kappa values) there was found to be a 'fair' strength of agreement ( $k = 0.358$ ), with 72% of the sites showing agreement between the domain expert's conclusions and the actual data.

b) Comparison of Actual data and the Conclusions of HEX

		Conclusions of HEX	
		Suitable	Unsuitable
Presence of crested newts in pond	Present	14	26
	Absent	3	57

Using appropriate analysis, there was found to be a 'fair' strength of agreement ( $k = 0.332$ ), with 71% of the sites showing agreement between the conclusions of HEX and the actual data.

c) Comparison of Actual data and the Conclusions of TRITON

		Conclusions of TRITON	
		Suitable	Unsuitable
Presence of crested newts in pond	Present	19	20
	Absent	1	59

(NB: For one of the ponds, TRITON drew no conclusions - that piece of data has been excluded from this test)

Using appropriate analysis, there was found to be a 'moderate' strength of agreement ( $k = 0.526$ ), with 78.8% of the sites showing agreement between the conclusions of TRITON and the actual data.

d) Comparison of the Conclusions of the Domain Expert and HEX

		Conclusions of HEX	
		Suitable	Unsuitable
Conclusions of Domain Expert	Suitable	11	7
	Unsuitable	6	76

Using appropriate analysis, there was found to be a 'moderate' strength of agreement(  $k = 0.55$ ), with 87% of the sites showing agreement between the conclusions of the domain expert and HEX.

e) Comparison of the Conclusions of the Domain Expert and TRITON

		Conclusions of TRITON	
		Suitable	Unsuitable
Conclusions of Domain Expert	Suitable	10	8
	Unsuitable	10	71

Using appropriate analysis, there was found to be a 'fair' strength of agreement(  $k = 0.426$ ), with 81.8% of the sites showing agreement between the conclusions of the domain expert and TRITON.

f) Comparison of the Conclusions of the HEX and TRITON

		Conclusions of TRITON	
		Suitable	Unsuitable
Conclusions of HEX	Suitable	13	4
	Unsuitable	7	75

Using appropriate analysis, there was found to be a 'good' strength of agreement(  $k = 0.643$ ), with 88.9% of the sites showing agreement between the conclusions of TRITON and the actual data.



g) Comparison of the Statements of Certainty attached to the Conclusions of the Domain Expert and TRITON

Comparison of conclusions/certainties of the domain expert and TRITON can be seen in the table presented below.

<u>Suitability of Pond</u>	<u>Level of Certainty</u>	<u>Code</u>
Suitable,	extremely convinced	SE
Suitable,	strongly convinced	SS
Suitable,	reasonably convinced	SR
Suitable,	possibly	SP
-	No conclusions drawn	N
Unsuitable,	possibly	UP
Unsuitable,	reasonably convinced	UR
Unsuitable,	strongly convinced	US
Unsuitable,	extremely convinced	UE

TRITON Conclusions

Expert	UE	US	UR	UP	N	SP	SR	SS	SE	Tot.
UE	-	4	9	2	-	-	-	-	-	21
US	-	5	10	6	-	-	-	-	-	21
UR	-	2	14	8	1	2	-	-	-	27
UP	-	1	5	5	-	8	-	-	-	19
N	-	-	-	-	-	2	-	-	-	2
SP	-	-	-	5	-	5	-	-	-	10
SR	-	-	-	3	-	-	-	-	-	3
SS	-	-	-	-	1	2	-	-	-	3
SE	-	-	-	-	-	-	-	-	-	0
Tot.	0	12	38	29	2	19	0	0	0	100

h) Comparison of the Certainties attached to the Conclusions of TRITON against Actual Data

(See previous page for key)

	TRITON Conclusions									Tot.
	UE	US	UR	UI	N	SI	SR	SS	SE	
Present*	-	-	2 <sup>A</sup>	-	-	6	-	-	-	8
Marginal*	-	2 <sup>A</sup>	7 <sup>B</sup>	9	2	12	-	-	-	32
Absent*	-	10	29	20	-	1	-	-	-	60
Total	0	12	38	29	2	19	0	0	0	100

\* - Note that the classification of 'present', 'marginal' and 'absent' relies entirely on the presence/absence of certain stages of the crested newt population, as recorded by the pond recorder. These are:

1) Present - this classification applies to ponds where either; any metamorphs are recorded, or; where a pond has eggs, larvae and adults present.

2) Marginal - This is where individuals are present, but do not fall into the two groupings identified in (1) Present.

3) Absent - Where no individuals are recorded.

Note that the classification does not rely in any way on other aspects of the pond.

A - With these 4 sites, the conclusions of TRITON do not correspond closely to the actual data. However, it should be noted that the pond came to either the same conclusion/certainty as the expert (in 3 cases), or with a more conservative judgement than the expert (in the remaining 1 case).

B - In these 7 cases, TRITON was reasonably convinced that these ponds were unsuitable to support crested newts, although some individuals were present. It should be noted that in 6 of the cases, TRITON agreed exactly with the expert's conclusions/certainties, or came to a slightly more conservative conclusion. In the remaining one case, TRITON was slightly more convinced that the pond was unsuitable than the domain expert.

Part 5: Correctness of Assessment of a Specific Pond  
by the Student Group

The student evaluators were asked to fill in a pond assessment form (see Appendix F) for a particular site. This site was picked specifically as untypical, being in a stage of transition between unsuitability and suitability for crested newts. The domain expert (Dr Rob Oldham) considered the pond to be suitable for crested newts. The students were then asked to use HEX to interpret the data they recorded, and the conclusions generated by HEX are summarised below.

Of 55 students:

	No. of Students
Pond Suitable:	10
Pond Suitable, as long as fish absent:	31
	—
Total 'Suitable':	41
	—
Pond Unsuitable because of;	
Size/depth	4
Nutrient levels	4
Waterfowl	2
Toxic Pollution	1
Size/Depth & Toxic Pollution	2
Size/Depth & Nutrient levels	1
	—
Total 'unsuitable'	14
	—

Part 6: Details of Noted Criticisms and Suggestions

Presented here are the written criticisms and suggestions given by 15 experts examining HEX and TRITON, 55 students examining HEX, and 20 students examining TRITON. Comments by the domain expert (Dr Oldham) are also noted. If HEX or TRITON are not specifically mentioned, then it may be assumed the comment relates to both systems.

(I) Criticisms

**Reasoning**

- (i) Pond depth as a means of 'suggesting' suitability is questionable (1 expert), because of categories used (domain expert).
- (ii) 'Steep slope' is only of minor importance in decision making (domain expert)
- (iii) 'Access to animals' is only of minor importance in decision making (domain expert)
- (iv) 'fewer species than expected' is a meaningless entity (domain expert)
- (v) To properly assess macrophytes, the time of the year must be stated (domain expert)
- (vi) Degree of isolation of pond must be considered (relative to other ponds/potential recruitment sources) (domain expert, 1 expert)
- (vii) The absence of pH as a factor (6 experts)
- (viii) It may not be correct to assume a pond is absent whenever fish are present (3 experts)
- (ix) Waterfowl a big problem - does question mean living/nesting in the pond, and what about occasional visitors (2 experts, 1 student)
- (x) Importance of barriers overestimated (2 experts)
- (xi) Is consideration of a 'barren slope' necessary? (2 experts)
- (xii) Altitude should be considered (2 experts)
- (xiii) Absence of pond age as a factor (1 expert)
- (xiv) Surely garden ponds are a special category (1 expert)
- (xv) Moorlands are a type of 'suitable' habitat (1 expert)
- (xvi) >25% pond shading rendering a pond unsuitable within TRITON is questionable (1 expert)

- (xvii) Is south shading necessary? (1 expert)
- (xviii) Are 'floating plants' significant? (1 expert)
- (xix) shouldn't human disturbance be considered? (1 expert)
- (xx) Should include the effect of pond slope on newt movement (1 expert)
- (xxi) Include rainfall levels (1 expert)
- (xxii) Questions about amount of shallow area in a pond (1 expert)
- (xxiii) Buildings may have a positive effect (1 expert)
- (xxiv) Geology should be considered (1 expert)
- (xxv) Turbidity also caused by fish and clay (1 expert)
- (xxvi) Access by cattle not necessarily a bad feature (as stated by HEX) - tadpoles feed on dung (1 expert)
- (xxvii) Frequency of road use should be included (1 expert)
- (xxviii) If the depth is <0.3m, HEX assumed that drying was frequent, even when it was stated that drying was 1-4 times per decade (1 expert)
- (xxix) HEX assumed fish were present in larger ponds, when user stated 'unknown' to fish presence (1 expert)
- (xxx) TRITON seems to use convictions too strongly in stressing unsuitable sites, and too weakly in stressing suitable sites (1 expert)
- (xxxi) Seems useless outside England (1 expert)
- (xxxii) A depth <0.3m does not necessarily make a pond impermanent (1 expert - however, it should be noted that this is not taken as so by either system, often there must be corroborating evidence).

#### **Explanation Facilities**

- (i) Both positive and negative relations are expressed in any order within TRITON - the explanations were equivocal, and perhaps confusing. (3 experts, 2 students)
- (ii) The term 'South of Wales' is not clear (1 expert)
- (iii) The final explanations concerning drawdown do not always come up (2 students)
- (iv) In HEX, the 'hypertext' explanations were self-referencing in a circular way (2 students)
- (v) Explanations of TRITON in poor English (2 students)

(vi) With HEX, the reasons for 'unsuitability' seemed inadequate (1 student)

#### **Final Conclusions/Explanations**

(i) TRITON seemed to cut off very quickly, not asking potentially important information (3 experts)

(ii) In TRITON, when finished with an object, menu should not return to top of list, but stay with that object (2 experts)

(iii) Need more scope in interrogation of final conclusions in HEX (1 expert)

(iv) In TRITON, have to hit key twice to get out of conclusion (1 expert)

(v) Final screen difficult to use (4 students)

(vi) In TRITON, the 'what-if' facility should be alphabetical (2 students)

#### **Questions/Answers**

(i) 'Species in reduced numbers' - needs proper definition or prior knowledge (3 experts)

(ii) With TRITON, choices on screen sometimes do not tally with assessment form (3 experts)

(iii) Plant Cover - what does this means? (1 expert)

(iv) In TRITON, default 'what-if' questions confusing (1 expert)

(v) Some questions unclear ('large' in 'large, dry expanses of land' needs defining, 'with holes' in 4(f), etc) (5 students)

#### **Wording Mistakes**

(i) For HEX, the screen about barriers should say 'the questions you are about to answer...' (1 student)

#### **Definitions**

(i) The question relating to 'access by large animals' needs some more specific definition (1 expert)

#### **Handbook**

(i) Handbook assumes the use of a colour monitor (1 expert)

(ii) Note that 'Number Lock' should be turned off when using INS and DEL keys (1 student)

## **Overall**

- (i) 'Tricky'/unusual ponds not evaluated properly (4 experts)
- (ii) TRITON had too strict a cut-off procedure (3 experts)
- (iii) Systems are unwieldy, as there are too many 'exceptions' to rules to make the systems function realistically (1 expert)
- (iv) Judgements tend to be too conservative and too absolute (1 expert)
- (v) The system can only be properly used at certain times of year (1 expert)
- (vi) Absence of 'conclude' facility in HEX (1 expert)

## (II) Suggestions/Comments

### **User Interface**

- (i) Use pictures to make more user friendly (1 student)

### **Explanations**

- (i) In HEX, perhaps a further tier of explanation, with even more detail, would be useful (2 students)
- (ii) The presentation of explanations by TRITON may be best done graphically, via arrows. (1 student)

### **Final Screen**

- (i) Liked 'what-if' facility in TRITON (1 expert, 3 students)

## **Overall**

- (i) Ability to go back to amend errors/change values would be useful (4 experts, 15 students)
- (ii) Print-out of questions/conclusions would be useful (2 experts, 1 student)
- (iii) Management advice would be useful (1 expert, 1 student)
- (iv) 'Do you really want to exit?' would be useful (1 expert)
- (v) Facility to edit answers may be useful (1 expert)
- (vi) Both systems (especially HEX) seem useful as educational tools (2 students)

**APPENDIX B**  
**THE DEVELOPMENT OF THE PERSEUS TOOLKIT**



## Introduction

The requirement for a higher-level tool, or 'shell', with which to build second-generation knowledge-based systems first arose in considering the requirements of a second generation knowledge-based system for an ecological domain (the evaluation of pond habitats with respect to an endangered species, the great crested newt). The unique features of using 'convictions' as a measure of belief, combined with the use of these convictions as the focus of ordering of reasoning, are not present in existing higher-level tools or shells. Existing tools are therefore not suitable for the construction of a second-generation system within this research. In this research, the shell developed was called *PERSEUS*.

The development of the *PERSEUS* toolkit, like any piece of software, underwent several interlinked stages, and these stages are collectively called a "life cycle". Figure B1 illustrates the typical life cycle of a piece of software.

Within this life cycle, each step was completed and then reviewed. Any errors, problems, and other difficulties required repetition of some or all parts of that stage, or recourse back to an earlier stage. In addition, these stages overlapped to some degree.

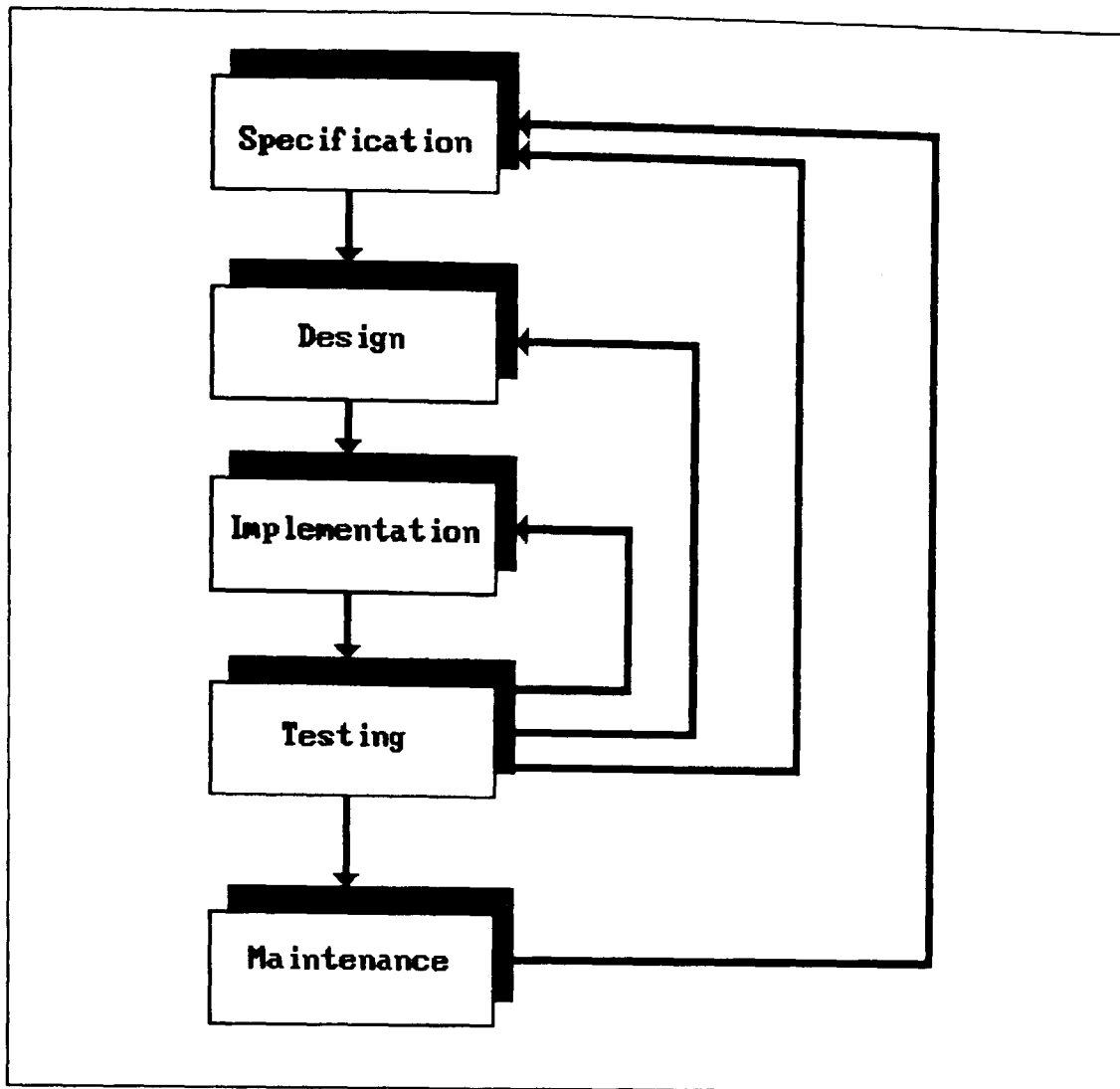


Figure B1: The Software Life Cycle  
(after Parsaye and Chignell, 1988)

## 1. Specification

Software development begins with consideration of the feasibility of building a piece of software (ie considering whether the software will meet potential needs), coupled with suitable planning. Absence of proper planning in any software/system development often leads to dissatisfaction with the system, and may result in users rejecting the system (Kendall and Kendall, 1992). Planning also lends structure to the development process, and helps to make the process more efficient and systematic. In considering the feasibility of the PERSEUS system, a number of aspects were considered;

i) The inclusion and integration of second generation knowledge based systems facilities that are required for this research (Table 12 in main thesis [p.100] describes these requirements).

ii) The likely future users of the specific knowledge-based system within this research (ie a system that performs pond evaluation with respect to the great crested newt), in terms of their;

- requirements,
- previous experience within the domain,
- access to hardware,

as well as the more general aspects of the user interface that enhance usability.

It was considered that one very significant group of users, naturalists and field workers, are most likely to have access to a computer via the IBM-compatible personal computer (PC). The ubiquity of the PC means that potential users (of all types) are very likely to have access to hardware.

iii) Likely degree of effort involved in construction, and the resources and potential approaches and tools available. This includes effort *per se*, and a comparison of the effort needed to construct a shell and knowledge-based system relative to the construction of a stand-alone system. This involved initial consideration of which available tools were the most appropriate for implementing the PERSEUS system. The AI language PROLOG was selected (see Section 4.5.2 of main thesis).

In judging the relative merits of building shell and system, relative to a one-off system, the functionality of the proposed (TRITON) system was considered. By definition, a number of abilities not commonly found in first-generation systems had to be available in a second-generation system. For example, a more flexible approach to acquiring inputs from users was desirable (eg machine-driven questioning or user-driven assertion of values). Such higher-level facilities, often associated with second-generation systems, are most easily implemented and adjusted by using a higher-level shell, and the ability to change such facilities may be useful at the testing stage. A shell would also simplify improvements to the knowledge base (given deficiencies and errors are likely to be present initially), and allow the PERSEUS architecture to

be used for different systems/goals. Such considerations led to the conclusion that the extra effort in building a shell was justified given the possible benefits.

## 2. Design

In designing the PERSEUS toolkit, typical parameters of software design were observed. Initial design concerned the consideration of the goal of the finished system. This naturally fragmented into a number of subgoals (eg a shell needs some access to the operating system; a shell requires some means to retrieve and store data given by the system builder or user; and so on). These subgoals are natural groupings that share common functions or conceptual similarities, and may be possibly further fragmented into specific tasks, or further subgoals. This process is initially carried out without consideration of the programming language, using both a subset of everyday language (as a 'structured narrative'), and a graphical method to represent the structure/flow of data in the finished system. As the individual functions/processes emerged, implementation of functions in a suitable programming language was a natural progression. Figure B2 illustrates this progressive breakdown used in the design of PERSEUS.

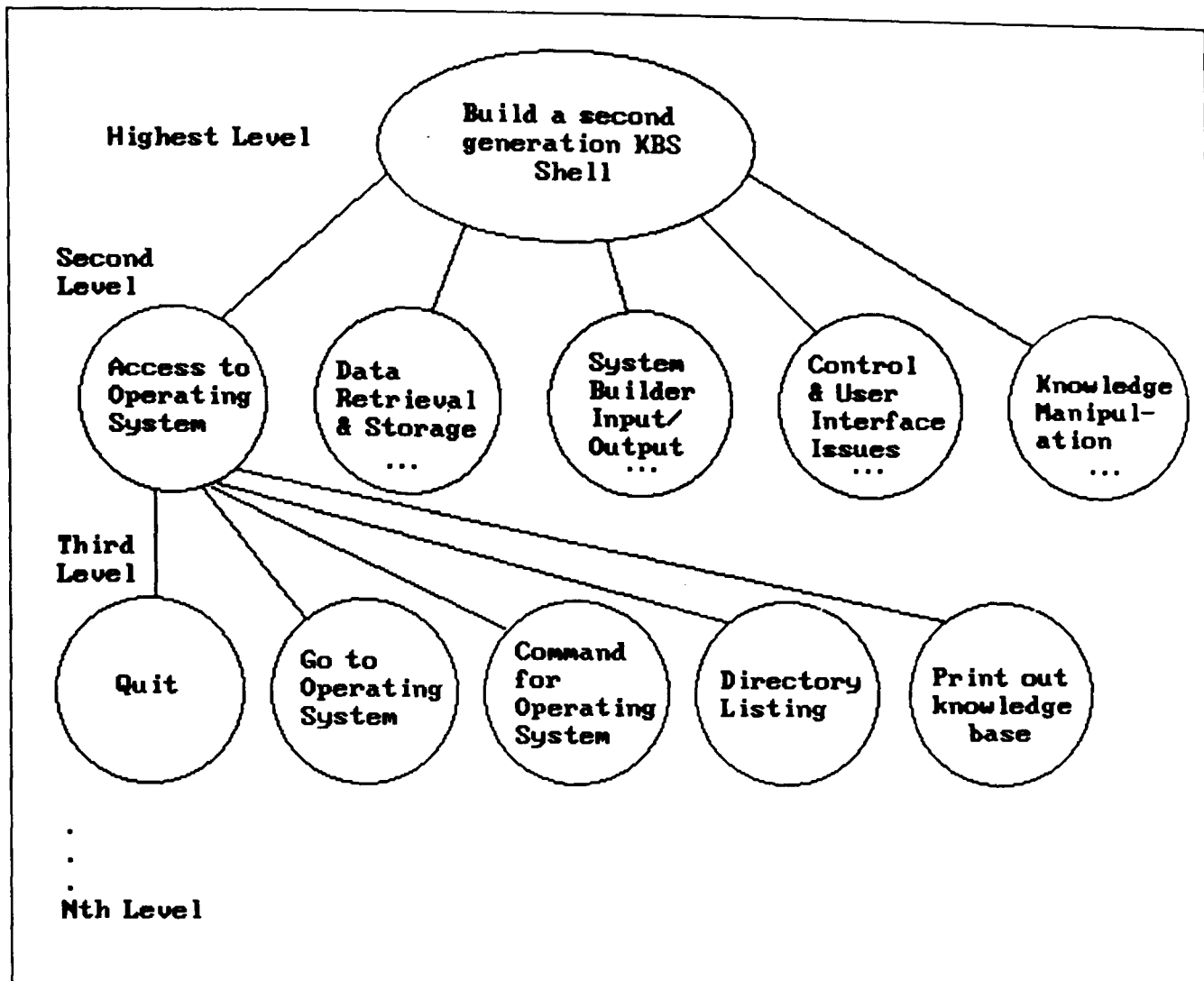


Figure B2: Stepwise Refinement of the PERSEUS Shell Design

The approach used in this work is referred to as *stepwise refinement* (Bratko, 1990). The passage from rough description of overall goal or goals to a set of specified functions/processes goes through sets of 'refinements', with each step containing the same general information, but in progressively more detail. Using this process, the options available within the PERSEUS system were specified.

A further aspect of design was the careful consideration of the user interface. The user interface is of prime importance, and needs to be designed to help users interact with the system for the users' maximum benefit (Kendall and Kendall, 1992). Provision of powerful functionality without suitable interface abilities is no longer considered a sufficient end-point in software development (eg Richards et al., 1986). Screen layout, background and foreground colour matching, amount of text on screen, user tools available (eg mouse), single button-press response, and various other ergonomic and aesthetic

aspects were considered in designing the interface.

In the PERSEUS system, a menu-driven interface was selected as the most suitable. This allows a uniform approach to using the various facilities of the system, without burdening the user with the need to remember special command words (as with command-driven systems). Menu-driven programs are common, and are familiar to most non-novice computer users. They also require less processing power than other types of interface (eg graphical user interfaces), and may therefore be used on a greater range of machines (ie may be used on machines that have insufficient processing power to efficiently handle graphical interfaces).

### 3. Implementation

On deciding that the means of development would centre on building a second generation knowledge based system shell, using PROLOG, for the IBM-compatible PC, various PC-based dialects of PROLOG were considered by literature review, and use of available packages. PDC-PROLOG was decided upon, as it addressed adequately issues of cost, flexibility, and efficiency. It also had present a range of features that facilitated the building of the PERSEUS shell - for example, satisfactory user interfaces were easily built in this language.

Each of the subgoals was coded and tested independently of the whole system, keeping development *modular* (Kreutzer and McKenzie, 1991). Testing occurred by using suitable and unsuitable responses, and by deliberately trying to 'crash' the function/subgoal. Once tested and thought to be satisfactory, the new modules or functions were integrated with the existing program, and further tested for any problems or errors in interaction. PDC-PROLOG aided in this approach greatly by (i) allowing modular development as a standard approach, and (ii) having extensive testing/debugging facilities with which to trap and find the source of errors. PDC-PROLOG enhanced the final system greatly by being able to convert programs to a compiled form that is very quickly processed

by the computer, speeding up the running time considerably, and generally improving efficiency (compilation is not a usual feature of many PROLOG dialects).

More general rules of coding were maintained at this stage, such as making the program transparent (ie easy to understand and read), easy to modify, robust to incorrect inputs, and names of functions were selected so as to properly explain the task of the function.

#### 4. Testing

Whilst testing of the code occurred during the implementation stage, other types of testing were required for the whole system. This involves testing along two sets of criteria; the first focussing on the shell; the second on the knowledge based systems that are created using this shell. Testing of TRITON has taken place, primarily to make sure the knowledge base was correct, and to ensure the system was robust (ie not prone to crashes). Prior to evaluation by end-users, the expert was asked to review the contents of the knowledge base, and the working system, for errors in output/judgement. Several errors did occur, and were corrected by the knowledge engineer and expert. This process occurred iteratively until the knowledge base was fully checked, and no more knowledge base or system errors were apparent.

Whilst full testing has not yet taken place on PERSEUS at this point in time (as this research has been more concerned with TRITON than the full functionality of PERSEUS), it should be noted that those aspects relating to the research have been properly tested by running hundreds of tests through the system, followed by careful checking. The focus of this testing has been to ensure;

(i) that the QCC works as specified,

(ii) that the processes of knowledge senescence/renascence have occurred as specified,

(iii) other aspects that are relevant to the end-user evaluation of TRITON are working as specified (eg conclusions are correct, explanations properly presented, etc), and

(iv) the TRITON system running under PERSEUS is robust.

Further testing and evaluation of PERSEUS is required prior to its use by other knowledge engineers. One such area requiring such evaluation is an assessment of the usability of the PERSEUS system by potential second-generation knowledge-based system builders.

## 5. Maintenance

Maintenance is undertaken for a number of reasons. It is conducted to improve the existing software, either in response to identification of limitations and/or errors, or to keep track of the changing requirements of users. Other reasons include a desire to improve the efficiency of a program (Kendall and Kendall, 1992). It must be part of system builder's repertoire to ensure that there are adequate channels through which users/organisations may provide feedback. Other foci of maintenance may include improvement of the user interface, and porting the system to other hardware/software (Parsaye and Chignell, 1988).

There has been no need for maintenance of PERSEUS to date, as it has only been used to construct TRITON for the present research.



**APPENDIX C**  
**REFERENCE GUIDE FOR THE PERSEUS TOOLKIT**

## PERSEUS - An Overview

The PERSEUS toolkit is a shell within which knowledge-based system can be constructed and run. It comes in two separate programs. The first, PERSEUS, is the tool used to construct a second-generation knowledge-based system, and as part of such a system;

- determine various aspects of the user interface and control parameters,
- compile the knowledge given into a more efficient form so that real-time reasoning may occur,
- specify the types of relationship that are allowed within the constructed knowledge-based system.

The second, PERSRUN (for PERSEUS-Runtime), is the package used to run finished systems, taking the knowledge base and other parameters specified by PERSEUS, and presenting users with a finished, working system.

Note that in this text, the builder of knowledge-based systems using PERSEUS is called the 'system developer', whilst the end user of a knowledge-based system developed under PERSEUS is called a 'user'.

### How to Get Started

The PERSEUS shell may be entered by;

- i) Selecting the correct drive (eg type CD\PERSEUS, followed by <return>)
- ii) If no particular knowledge base is required, then type 'PERSEUS', followed by <return>. If you wish to load a particular knowledge base, for example 'tester.kbs', then type 'PERSEUS TESTER' (entry of the .KBS suffix is optional), and press <return>. The 'tester' knowledge base will be automatically loaded.

On entering the PERSEUS environment, you will encounter the initial screen (see Figure C1).

Select with arrows or use first upper case letter

Figure C1: The Initial PERSEUS screen

## 1. PERSEUS

Figure C1 shows that the initial opening screen of PERSEUS has a set of words or expressions running along the top. These terms are a menu by which commands or further menus of commands or directives can be invoked. They may be selected in two ways; (i) by pressing the first/highlighted letter of the choice eg 'F' for **F**iles, (ii) by the use of the arrow keys (**←**, **→**) to move to different terms (shown by a moving highlight), and selecting a choice by hitting **<return>**.

Each of these main menu terms invokes a process or a further menu when chosen. To cancel any of these subsequent menus/choices, hitting the **<esc>** will typically return the system developer to the highest-level choices. Note that the bottom bar of the screen offers brief information to the system developer about making/cancelling choices.

Brief details of the highest-level choices are given on the following page. More detailed information is given in the following subsections.

**System** - This invokes a menu that subsequently allows a variety of system-level selections (eg quitting the PERSEUS shell).

**Files** - This invokes a further menu that allows a retrieval or storing of both knowledge-base files (suffixed .KBS) and text (ASCII) files, as well as renaming knowledge-base files, and the clearing of the PERSEUS workspace.

**Edit** - This invokes a text editor that allows PERSEUS terms to be entered into, removed from, or altered within, the knowledge base.

**Objects** - This allows access to object-attributes created by PERSEUS during 'compilation' of terms. These are listed, and may be accessed to allow the addition of material for user interface purposes (eg addition of 'definition' to the object-attribute).

**Build** - This allows compilation of knowledge from natural terms (entered via the editor or in file loading) to a form that is efficient for searching and reasoning purposes. It also allows the removal of redundant object-attributes.

**User Interface** - This allows various aspects of the user interface to be selected/expressed, including a title for the working system, an introductory and finishing text, selection of colours for various parts of the finished knowledge-based system, the user level (eg novice, near-expert, etc), and the initiative level (ie machine-controlled questioning, facts asserts by the user, or a mixture of the two).

**Control** - This allows a number of higher-level parameters to be set by the system developer, include the setting of the system's goal, whether the runtime should have repeated runs (so that users may use it for more than 1 time without having to return to the operating system), and so on.

## 1.1 Menu Options

When a selection from the main menu is made, the system developer is often presented with a further, pull-down menu. A selection from this menu can be made by either;

- (i) pressing the first letter from each option (eg 'q' for 'Quit'), or
- (ii) by using the (↓,↑) keys, and hitting <return> when desired selection is highlighted.

To cancel the menu, and return to the main menu, simply press <esc>.

Sometimes the choice has further menus, or other input requirements - this is denoted by an option being followed by two periods (..). For example, under the **System**, main choice, the option '*Command for OS..*' has these two periods. This denotes further menus, or some other action that involves the system developer submitting further information (in this case, entry of the command in question).

### 1.1.1 System

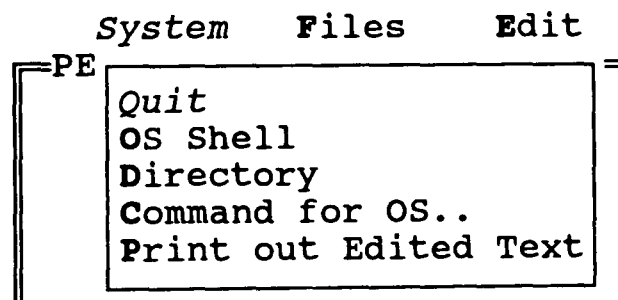


Figure C2: The 'System' Menu

**System-Quit** - This directs the PERSEUS shell to shut down. If you have been doing work on a knowledge base and have not saved the most recent changes, it will ask if you wish to save this knowledge base ({y/n} denoting yes or no). If you do, and a copy of the file already exists, it will also ask if you wish to overwrite the existing copy. In this way, the system ensures a knowledge base that is wanted is not written over by mistake, or lost by not being saved.

**System-OS shell** - 'OS' stands for 'operating system'. If this option is taken, then the PERSEUS system remains in memory, whilst you are placed in the operating system environment. You have not left the PERSEUS system, but are temporarily placed outside its bounds. To return to the PERSEUS shell, type EXIT <return>.

NB: It is a common error to forget you are still effectively in the PERSEUS shell, and by switching off the computer at this point, you may lose valuable work. In addition, PERSEUS is still loaded in memory in this mode, and so only a small amount of processor memory is available for use at this time. Only small programs will be able to run, and to use larger programs PERSEUS must be properly vacated for such programs to work.

**System-Directory** - This gives a listing of the current directory. A system developer may also move down the directory tree (ie down one level towards the root directory) by selecting '..\'', or to subdirectories of the present directory by selecting <'Name of subdirectory>\'. This is a browsing facility, and does not permit any operations to occur on highlighted files.

**System-Command for OS..** - This allows the system developer to undertake a one-off DOS command from within PERSEUS. A system developer may select this option, and be confronted with a means to enter the command. If this is not necessary, the system developer may select <esc>. Otherwise commands can be entered, followed by <return>. (Example: del \*.bak). If many operations are to be undertaken via the operating system, the system developer is advised to either quit PERSEUS, or 'shell out' of PERSEUS (using **System-OS Shell**).

**System-Print out edited text** - This prints the contents of the Edit text onto paper (also checking that your computer is attached to a waiting printer). If you wish to inspect the file before printing, do so in Edit, or copy the text to an ASCII file using 'Files-Write 'edit' text to ASCII' (section 1.1.2). This ASCII file may then be inspected/changed using a conventional text processor.

### 1.1.2 Files

**Files-Load Knowledge Base** - If this is selected, then a box pops up on screen, inviting entry of the name of a knowledge base to load. The system developer may then instigate either of the following;

i) enter the name of a knowledge base. If the knowledge base exists in the present directory, then it is loaded. If it does not exist, then the system developer is given an appropriate error message, and returned to the main screen.

ii) Press <return>. If any knowledge bases exist in the current directory (recognised by the suffix .KBS), PERSEUS gives a listing of these knowledge bases. A system developer may choose to load one the knowledge bases by highlighting the choice, and pressing <return>. If no knowledge bases exist, the box for entering a name is maintained. It can only be cancelled by pressing <esc>.

Note that a system developer may enter a name which includes directory details and wild-cards (eg ..\tryout.kbs, temp\\*.kbs).

**Files-Save Knowledge Base** - If this is selected, then a number of occurrences may take place, depending on the current state of the system. These are;

i) If a knowledge base has previously been loaded or saved, then the last name given is presented. The system developer may select this without change, but will be warned that this means the old knowledge base will be overwritten, and given the option to cancel.

ii) If a new (unnamed) knowledge base is being constructed, or if there is no knowledge base, then the system developer is presented with a blank box, and given the opportunity the name the knowledge base. If the name of an existing knowledge base is used, the system developer is warned and given an opportunity to change the name.

Note that directory details may be included in the name, and the knowledge base will be saved to the specified directory (if it exists).

**Files-Rename Current Knowledge Base** - This allows the system developer to change the name of the knowledge base currently loaded into PERSEUS. If the system developer selects a name that already exists, there will be an opportunity to cancel the operation.

**Files-Empty for New File** - This clears the 'Edit' text, and all user interface and control settings, reverting back to the 'default' settings used by PERSEUS. System developers are given a clear warning of what will occur prior to emptying.

**Files-Write 'Edit' text to ASCII..** - This copies the contents of the 'Edit' option to an ASCII (text) file. This file may then be read, edited, etc independent of PERSEUS. If there is nothing in the 'Edit' text, then the system developer is informed of this, and an ASCII file is not created. If the name given to the ASCII file is already being used, then the system developer is warned of this.

**Files-Get ASCII for 'Edit' text..** - This reads an ASCII file into the PERSEUS system, and can be accessed via the 'Edit' option. If the 'Edit' option already has text present, the system developer is given the opportunity to save that text.

**Files-UI\Control Setup Store\Retain..** - This option allows the system developer to retain or load user interface and control options (described under 'User Interface' and 'Control') independently of the 'edit' text (ie the knowledge base). Selection of this option prompts a further menu (figure C3):



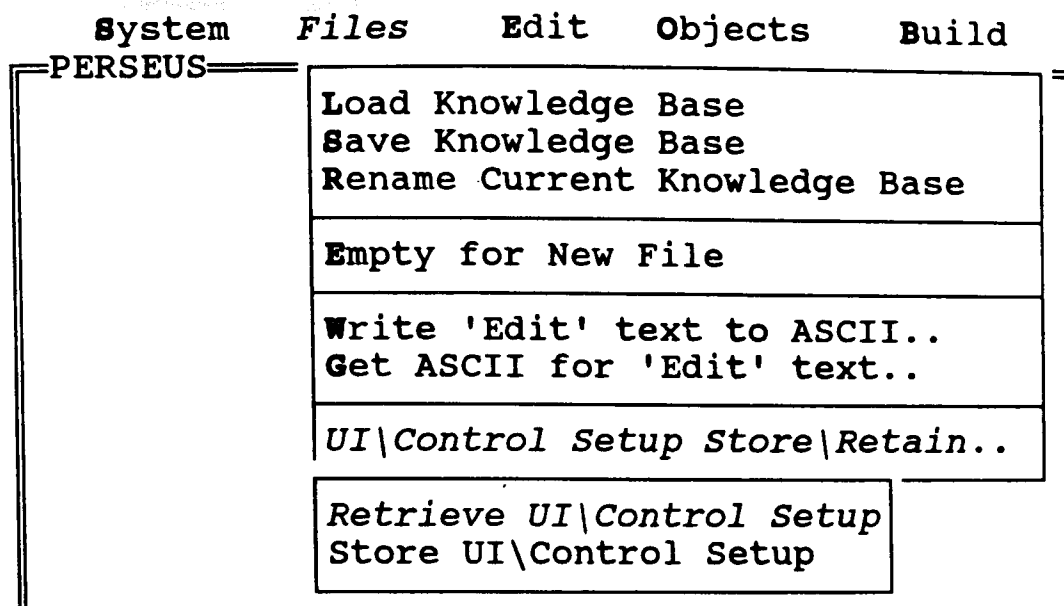
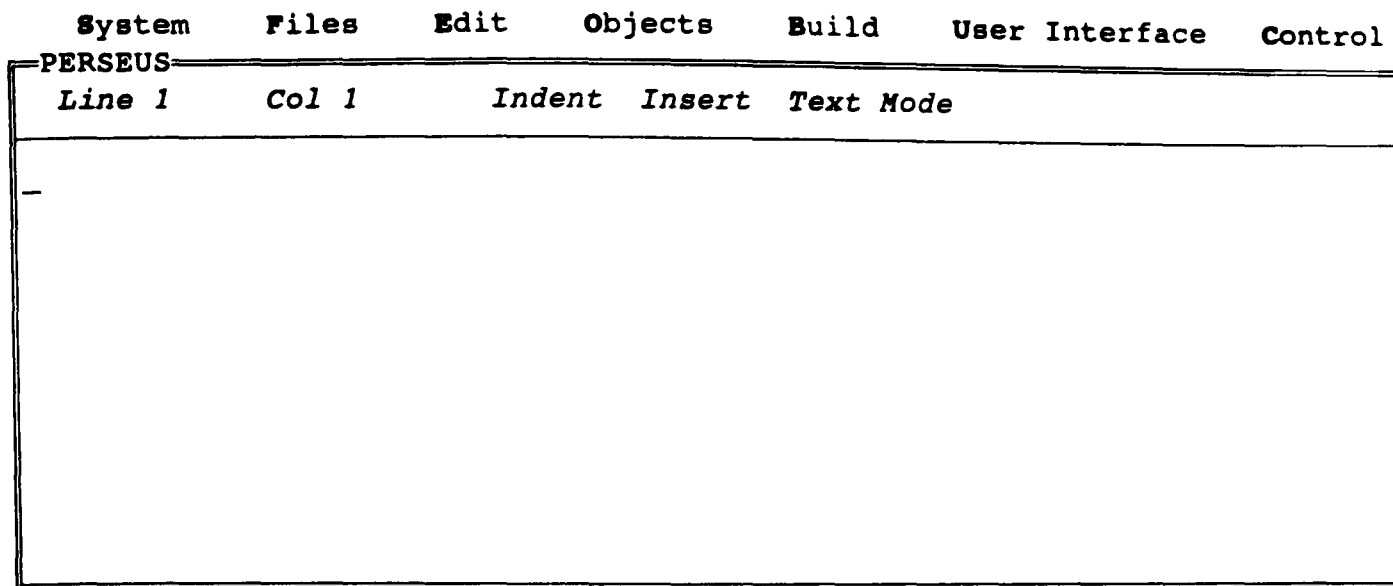


Figure C3: Files-UI\Control Setup Store\Retain Submenu

i) **Retrieve UI\Control Setup** - This allows the system developer to retrieve any existing setup file, denoted by an '.STP' suffix. The default setup saved by PERSEUS is PERSEUS.STP (see p.C-15 'User Interface-Automatic Storage of UI\Control..').

ii) **Store UI\Control Setup** - This allows the system developer to store the present values for the user interface and control choices, and is stored with an '.STP' suffix. The default name is PERSEUS.STP, and system developers may change this as they wish. If system developers select a name that is already used, they are warned of this duplication, and have the option to change the name.

1.1.3 **Edit** - This gives the system developer access the knowledge base. If no knowledge base has been loaded, then new text may be entered. Figure C4 shows the 'edit' screen, with no knowledge base present.



F1-Help F2-Save F3-Load F5-Zoom F6-Next F7-Xcopy F8Xedit F10-End  
Figure C4: The 'Edit' screen

The facilities of the editor are numerous, and are fully covered by the editor itself (using <f1> within the editor). Given here is a brief description of each of the f-keys, and other important keys.

<esc> - This aborts any changes to the text. If changes have been made, system developers are asked if they are sure they wish to abort.

<Ins> - This toggles the editor from 'insert' mode (ie entered letters are placed between existing letters/spaces), to 'overwrite' mode (ie entered letters write over existing letters).

<f1> - A help key. Can be pressed at any time, and provides a full description of the facilities of this editor.

<f10> - This saves text changes, and returns to the main PERSEUS menu.

<f2> - This copies current text into an ASCII file. The mode of operation is the same as the main menu option 'Files-Write 'Edit' text to ASCII..'

<f3> - This reads an external ASCII file into the editor. The mode of operation is the same as the main menu option 'Files-Get ASCII for 'Edit' text..'

<f5> - This toggles the editing space between the 'window' shown in figure C2, and the full screen.

<f6> - No function in this system

<f7,f8> - The f7-key allows the editing of other text files outside the PERSEUS system. f8 allows entry into these other files, and copying some or all of these files into the current 'edit' text. This allows the system developer to merge different ASCII files within the editor.

The knowledge base can contain three types of phrases. Filler words may occur anywhere in the phrases (see 'Control-Filler words');

i) Statements - These have the generic form:

<Object<sub>1</sub>> <Attribute<sub>1</sub>> <Value<sub>1</sub>> <relation> <Object<sub>2</sub>>  
<Attribute<sub>2</sub>> <Value<sub>2</sub>> <conviction>.

Eg: Ecosystem<sup>(O1)</sup> diversity<sup>(A1)</sup> being high<sup>(V1)</sup>  
implies<sup>(rel)</sup> ecosystem<sup>(O2)</sup> productivity<sup>(A2)</sup> is medium<sup>(V2)</sup>,  
in many cases<sup>(con)</sup>.

(The superscript notations denote the significant words/terms. The remainder are 'fillers', which help maintain sensible interpretation of complete terms by system developers and users.)

ii) Inheritances - Of the generic form:

<Object<sub>1</sub>> <inheritance term> <Object<sub>2</sub>>.

so that Object<sub>1</sub> inherits features of object<sub>2</sub>.

Inheritance terms are devised under 'Control-Inheritance Terms'.

eg: Ponds<sup>(O1)</sup> are kinds<sup>(inh)</sup> of ecosystem<sup>(O2)</sup>.

iii) Facts - Of the generic form:

<Object> <Attribute> <Value> <Conviction>

eg: Pond<sup>(O)</sup> diversity<sup>(A)</sup> is high<sup>(V)</sup>, in some cases<sup>(con)</sup>.

1.1.4 Objects - This choice gives access to object-attributes created in compiling the knowledge base (see p.C-13 'Build-Compile'). If this option is selected but no compilation has yet occurred, a message will occur informing the system developer of this mistake. If compilation has occurred, then a menu of object-attributes will be given. The system developer may scroll up and down this menu, and select appropriate object-attributes by pressing <return>. The menu can be left by choosing <esc>, which returns the system developer to the main menu.

If the system developer chooses an object attribute, a screen similar to the one below will occur:

System	Files	Edit	Objects	Build	User Interface	Control
--------	-------	------	---------	-------	----------------	---------

OBJECT ATTRIBUTES

Object: <name of object>  
Attribute: <name of attribute>  
Values: <set of values used for this object attribute in knowledge base>  
Object Equivalents: <set of possible equivalents/other words for this>  
Attribute Equivalents: <set of possible equivalents/other words for this>

Question Level: 1

Added Values: \_\_\_\_\_

Question: \_\_\_\_\_

Definition:

<esc> Abort      <Up/Down Arrows> to Move      <f10> Save

Figure C5: A generic Object-Attribute Screen

The first five rows of information (from 'Object:' to 'Attribute Equivalents:') are for viewing only, and cannot be changed within this window.

The row starting 'Object Equivalents:' gives details of other words that are used to represent the object in question. For example, 'canine', 'hound', 'pooch' may all be used to mean 'dog'. The setting of these equivalents occurs under 'Control-Equivalents-Objects'. A similar explanation is true for 'Attribute Equivalents:', where other terms may be used to represent the attribute in question. For example, 'turnover' may be an equivalent to 'productivity'. These attribute equivalents are set under 'Control-Equivalents-Attributes'.

The row of 'Question Level:' permits the system developer to attach a numeric weighting to the object-attribute (default 1). These numbers may be used to specify the level of user to which the question may be properly asked. For example, level 1 is for novices in the field to which the knowledge base is geared, level 2 for people with some but not extensive knowledge, and so on. The system developer can have up to 9 levels. The compiled system can then be directed to use questions of a certain number or below (under 'User Interface-User Level..'). Systems set to user level 1, will only ask questions of 'question level 1' (ie suitable for novices), whilst systems of level 2 will be able to ask questions of 'question level 2' or below (assuming that people with some experience would be capable of answering questions that are directed at novices). In this way, a system developer can construct a single knowledge base, and easily modify this knowledge base to ask suitable questions to users of different levels of ability.

The row of 'Added Values:' allows the system developer to perform two main tasks;

i) Add values to the existing values identified in the compilation process, so that the menu presented to users is complete.

ii) Rearrange existing values into a preferred order (eg 'high,medium,low', rather than 'medium,low,high'.)

Both of these abilities are available simultaneously.

The row of 'Question:' allows the system developer to state a question about the object-attribute of current interest, which will override the default question given

by PERSEUS (which is 'What is <Object> <Attribute>?').

The rows of 'Definition:' allow the system developer to attach a definition, or other useful text, to the object-attribute. This text is then available to users.

If any changes are made to the object-attribute, then these may be saved by selecting <f10>, returning the system developer to the object-attribute menu. If changes are made, and wish to be cancelled, <esc> is pressed. The system will ask for confirmation of this, to ensure wanted material is not lost.

#### 1.1.5 Build

**Build-Compile** - This takes the contents of the 'edit' text, and the contents of certain 'Control' values (eg filler words, inheritance terms, types of relationship, equivalents), and compiles this information into a form that may be ultimately used by PERSRUN. Any errors that occur are flagged, and compilation is discontinued until these are corrected.

**Build-Purge** - This removes any object-attributes previously created, but no longer used in the compiled knowledge base.

#### 1.1.6 User Interface

**User Interface-Title** - This allows the system developer to specify a heading that occurs in the running knowledge base. If no title is entered, the default heading of 'PERSEUS' is used.

**User Interface-Introduction Screen** - This allows the system developer to input one or more screens that are presented to users prior to using the system.

**User Interface-Finishing Screen** - This allows the system developer to input one or more screens that are presented to users prior to leaving the system.

**User Interface-Colours..** - This allows the system developer to change the colours of certain areas of the screen of systems running under PERSEUS, namely: (i) Pull-down Menus (ii) Frames (ie borders of windows) (iii) Text, and (iv) Status Line (ie help line at base of screen). These four are presented as choices on a menu. When one is selected from this menu, the developer can select new default colours for this option chosen by moving around a palette, and pressing <return> when a desired colouring is highlighted (or cancelling with <esc>). On cancelling or making a selection, the developer is returned to the 'User interface' menu.

**User Interface-User Level** - This allows the system developer to select the questions used by a knowledge-based system built under PERSEUS. The default level is 1. In the 'Objects-Attribute' subsystem, each object-attribute may have a question level associated with it. This may be 1 to 9. 1 indicates the question is suitable for those users with the most basic level of knowledge (eg novices), and higher values represent higher levels of knowledge (the developer may have 9 levels, if necessary). The setting of the 'user level' determines which questions will be asked. If it is left at the default level (1), it will ask questions of level 1 (ie novice-level questions). If it is set at a higher value (eg 3), it asks questions of object-attributes with this or lower levels attached (as you would expect more knowledgeable individuals to be able to answer questions that a novice may answer).

Using this method of 'gating' questions, it would be possible for a system developer to use the same knowledge base to produce several systems, geared to users of different abilities or levels of experience.

**User Interface-Mode of Initiative..** - This allows the system developer to direct a working knowledge-based system to work in different modes. The mode may be selected by menu choice. These modes are as follows:

i) **Machine Driven** - This is typical of the way most knowledge-based systems work. The system determines what knowledge it requires to make a decision, and gets

necessary information by asking the user questions. The user cannot direct the system to focus on certain values that the user may feel are significant.

ii) User Driven - This is where the user selects what information is given to the knowledge-based system, and the system must come to some conclusion using this information only.

iii) Combined Mode - This is where the user provides information, and asks the system to come to some conclusion. The system can supplement this knowledge by asking the user further questions that it deems important in decision-making.

User Interface-Automatic Storage of UI\Control - This allows the system developer to direct PERSEUS to automatically save (or not save) UI/Control settings, independent of the knowledge base (by selecting 'yes' or 'no'). 'Yes' is the default value for this, and the information is saved in 'PERSEUS.STP' (the user has the option to rename this setup file).

#### 1.1.7 Control

Control-Goal.. - This allows the system developer to set the object-attribute that is the goal of the reasoning process. The changing of the goal has no affect on the compilation process, and therefore the goal of a system built under PERSEUS is very easily modified (changing the goal of a first-generation knowledge based system often requires a major rewrite). On selecting this option, a box appears as in figure C6, on the following page.



System    Files    Edit    Objects    Build    User Interface    Control	
PERSEUS	Goal..
	Cutoff conviction
Goal	
	Equivalents..
	Types of Relation
	Inheritance Terms
	Filler words
	Parameter Changes

<esc> Abort    <f10> Save

Figure C6: The 'Control-Goal..' screen

The box with the heading 'Goal' allows the system developer to enter the object and attribute that is to be the goal of the knowledge-based system. There are two fields in this box (the underline denotes the 'object' field), and these can be swapped by the system developer selecting <return> or appropriate left/right arrow keys. If a previous object-attribute goal is present, then this will be presented in the fields. To make changes, the names of the desired object-attributes are entered, and <f10> should be selected. To abort the operation of goal setting, <esc> should be selected.

Control-Cutoff Conviction.. - This selection allows the system developer to decide at what level of conviction the search to satisfy a stated goal should be discontinued. In larger knowledge bases, and/or when a decision can only be acceptable under more 'convincing' evidence, this allows the system to come to a conclusion more quickly.

On making this selection, the system developer is presented with a menu and question (Figure C7).

<i>No level of cutoff</i>
<i>in all cases</i>
<i>in virtually all cases</i>
<i>in most cases</i>
<i>in many cases</i>
<i>in about half of the cases</i>
<i>in some cases</i>
<i>in few cases</i>
<i>in very few cases</i>

*At what level of conviction should the search be discontinued?*

Figure C7: Control-Cutoff Conviction Submenu

The default selection is 'No level of cutoff', directing the search for a goal to continue until all of the knowledge base has been considered. The remainder of the selections reflect the possible convictions attached to terms in the knowledge base. If one of these is selected, for example, 'in some cases', then the search for a goal will not continue if the remaining evidence for/against a goal-value is at or below this conviction level (ie at the less convincing levels of 'in few cases' and 'in very few cases'). This means that the system developer may direct search to look for the most convincing evidence only. It should be noted that the system developer can choose to select 'in all cases' - by doing this, the developer would be making the entire knowledge base unavailable for consideration, and the system would always come to an inconclusive outcome. For this reason, if any restriction is to be designated by the system developer, it should be at the lower levels of conviction. The entire list of convictions is presented in this menu for the sake of completeness, and it is the responsibility of the system developer to utilise this feature properly.

**Control-Repeated Runs..** - This facility allows the system developer to designate whether the finished knowledge based system allows the end user the chance to engage in further consultations after the first run. On selecting this option, the system developer chooses from a yes/no menu (default 'yes').

**Control-Object Adjustment..** - This facility allows the system developer to designate whether the finished knowledge based system allows the end user to adjust the values of any object-attributes during any part of the consultation. In this way, the user not only answers questions presented by the system, but can give details that he/she thinks may be relevant to decision-making. On selecting this option, the system developer chooses from a yes/no menu (default 'yes').

**Control-Switch Unknowns..** - This facility allows the system developer to designate whether the finished knowledge based system automatically adds 'unknown' to the selection of answers a user may give to any particular question. On selecting this option, the system developer chooses from a yes/no menu (default 'yes').

**Control-Equivalents** - This facility allows the system developer to designate equivalent words for any terms/words used in the knowledge base. These are categorised into objects, attributes, values, and convictions. For example, the system developer may designate the word 'ponds' and equivalent to the object 'pond'. In compiling the knowledge base, the word 'ponds' is then treated as the word 'pond'. Similarly, an attribute 'productivity' or value 'low' may have equivalents of 'production' or 'poor' respectively. Using these equivalents for objects, attributes, and values, the expressions used in the knowledge base, and those presented to the user in explanations, can be more natural than would otherwise be possible.

Similarly, convictions may have alternative expressions; for example 'in all cases' may be replaced by 'always'. This would allow the phrase 'pond anglers being present always indicate pond fish are present', instead of the more stilted 'pond anglers being present indicate pond fish are present, in all cases'. It must be noted however, that the system developer must be careful to maintain the integrity of the meaning of the original convictions. This facility, however, allows the system developer to create his own expressions of qualitative uncertainty.

Control-Types of Relation.. - Using this facility, the system developer may specify the words/terms used in expressing the relationship between one object-attribute-value (O-A-V) and another. On selecting this option, a further menu occurs, with two options; 'Asymmetrics' and 'Symmetrics'. Asymmetric terms are directional. They are used when the value of the first object-attribute may be used to instantiate the value of the second object-attribute. The default term is 'implies' (eg '<O-A-V> implies <O2-A2-V2>, in some cases'). Symmetric terms are used when the instantiation can occur in either direction, and the default term is 'is equivalent to'. For example, given '*pond productivity being medium is equivalent to pond diversity being medium, in most cases*', it may be possible to infer a value for diversity given a value for productivity, or vice versa.

On selecting either of the options, the system developer is presented with an edit-screen. The default values will be present if the consultation is new; the asymmetric choice will show 'implies', and the symmetric will show 'is equivalent to'. These may be deleted or retained, and further terms may be added. Each new term should be placed on a new line. Examples of possible terms may include the asymmetric terms 'causes, cause, indicates, indicate, means', and symmetric terms 'equals, is the same as'. It is the responsibility of the system developer to retain the properties of asymmetry and symmetry in the terms chosen to represent the relationships between O-A-V's.

Control-Inheritance Terms - On selecting this option, the system developer will be presented with an edit screen containing the default (or previously written) inheritance terms used in the target knowledge base. The default inheritance terms are; '*kinds, kind, types, type*'. Others that may be reasonably used may include '*sort(s)*', '*variant(s)*' etc. Individual terms must be put on a new line. It is the responsibility of the system developer to maintain the properties of 'inheritance' in terms used.

**Control-Filler words** - This allows the system developer to specify filler words that are ignored in compilation, but are retained in explanation. Use of these words may enhance the clarity of explanations.

On selecting this option, the system developer enters an edit screen. If the system is being built from scratch, the developer will have present a set of default filler words. These include 'at, to, the, a, an', and a variety of other words. Also included are full stops and commas. The defaults may be deleted, or new words added. Each word/term must be placed on a new line.

**Control-Parameter Changes..** - This presents the system developer with the option to allow the end user the facility to change many of the user interface/control options (eg switch unknowns on/off) during a consultation. The developer is presented with a yes/no menu for selection (default 'no'). It is recommended that this option is only used in development, when the system developer is considering what options should be available to end-users.

## 2 PERSRUN

A compiled knowledge base (denoted by <name>.kbs) must be properly converted to a suitable form in order to run under the PERSRUN system. The conversion is done by the following means;

- 1) Make sure the machine is in the correct directory, and is in the operating system environment.
- 2) Type 'make2run name(.kbs)', and press <return>.

If the knowledge base is in a suitable (compiled) form, this program will produce a runtime knowledge base, recognised by the full name '<name>.kbr'. To then run this compiled knowledge base, the following is done:

- 3) Type 'persrun name(.kbr)', and press <return>.

The PERSRUN system operates in a variety of ways, according to what user interface/control parameters have been specified. If the initiative level have been set to 'user-driven' or 'combined mode', then the user will first encounter a list of object-attributes (eg pond-productivity, ducks-present, crested newt population-present) which may be chosen in order for the user to assert values for them. On leaving this listing, a user-driven knowledge based system will state its conclusions from the information given - it operates on the assumption that no other values are known. A combined-mode system will also draw conclusions, but does not rely on the assumption that all other values for object-attributes are unknown. It may ask the user questions to augment existing knowledge for two possible reasons; (i) to establish a more definite (ie more certain) value for the goal; (ii) to make sure that contradicting evidence is not available/present which may otherwise invalidate conclusions. The third possible setting for the initiative mode, 'machine-driven', the system gathers knowledge by asking the user questions, selecting the most suitable questions for arriving at the most certain value for the object-attribute that is the goal of the knowledge-based system.

At any point, a user may ask a number of questions about the object-attribute that is the focus of the present question. Users may ask for a definition by selecting <f2> (that is stated by the system developer under the PERSEUS 'Objects' option). A user may also enquire 'why' a question is being asked (ie what relationship it has with the goal object-attribute) by selecting <f3> - the explanation given is derived dynamically from the deep knowledge present within the system. Users may also stop the questioning at any time, and ask for conclusions given present knowledge (by <f4>), or quit the session completely (<f10>).

Depending on what has been specified as being available in the PERSEUS system, other facilities may also be available as f-keys. These are presented on the next page.

<f5> Adjustments - This is available if the system developer selected 'yes' for the PERSEUS option 'Control-Object Adjustment'. This allows the viewing/changing of values of the object-attributes present within the knowledge base. The use is presented with a menu listing of object-attributes, which may be selected using <return>. On selecting an O-A, the user may then assert a value (via menu selection) for the object-attribute.

<f6> UI Chngs (Changes) - This is available if the system developer selected 'yes' for the PERSEUS option 'Control-Parameter Changes..'. Note that this facility should only be used by the system developer in deciding the full format of the working knowledge-based system, as it is likely to prove counterproductive if available to end users.

On pressing <f6>, the user is presented with a menu that allows different user interface/control options to be altered. The menu contains the following options (which are then explained:

*Goal Details* - This allows the goal of interest to change. It gives a listing of different object-attributes. The user may select a new goal by pressing <return>, or cancel by pressing <esc>.

*User Level* - Equivalent to User Interface-User Level (see under section C-1.1.6)

*Initiative Level* - Equivalent to User Interface-Mode of Initiative (see under section C-1.1.6)

*Cutoff Conviction* - Equivalent to Control-Cutoff Conviction.. (see under section C-1.1.7)

*Repeated Runs* - Equivalent to Control-Repeated Runs.. (see under section C-1.1.7)

*Toggle Unknowns* - Equivalent to Control-Switch Unknowns (see under section C-1.1.7)

*Parameter Changes* - Equivalent to Control-Parameter Changes.. (see under section C-1.1.7)

*Object Adjustment* - Equivalent to Control-Object Adjustment.. (see under section C-1.1.7)

The running knowledge-based system comes to conclusions given information by the user. On reaching a conclusion, the system may be questioned to determine how that conclusion has been reached, and its explanations are derived from the deep knowledge present within the system. The user may also decide to change any number of object-attributes, and request the system to come to a conclusion given this new information (using <f5> "What-if?").

A variety of other facilities may also come into operation in the knowledge-based system running under PERSRUN, given proper cues by the system developer within the PERSEUS system. These include selection of colours used, the title used, the presence of an introductory and closing screen (where desired), and whether the user can go through repeated runs (ie at the end of one consultation, the user may continue with another without recourse to the operating system).



**APPENDIX D**  
**USER GUIDE FOR HEX**

# **A GUIDE AND INTRODUCTION TO HEX: HABITAT EVALUATION EXPERT SYSTEM**

BY MARK F. CAIN, 1992

SCHOOL OF COMPUTING AND MATHEMATICAL SCIENCES

DE MONTFORT UNIVERSITY LEICESTER,  
THE GATEWAY,  
LEICESTER,  
LE1 9BH.

## 1. An Introduction to HEX

The habitat evaluation expert system, HEX, is a computer-based system that assesses the habitat of a pond and its surroundings. It performs this task in order to decide whether or not the site is able to support a colony of the great crested, or warty, newt (*Triturus cristatus*). It tackles this problem by asking questions of the computer user, so as to acquire information from which it can decide a status for the pond's suitability.

In addition to asking questions and deciding upon the status of a pond site; the HEX system can explain to the user why any particular question is being asked, and why certain features are of relevance to the crested newt. Definitions of common ecological terms are also embedded within the explanation facilities.

The easiest way to become familiar with the HEX system is by using it. However, this guide is intended as an introduction to its facilities, and covers all aspects of the system. The guide should only be used in conjunction with the HEX system, and reading the guide separately from the HEX system will be of little benefit. Immediately after using this guide, the user should experiment and 'play' with the system to become more familiar with its operation. Note that if any problems occur, hitting the <Esc> or <f2> key (both usually near the top and to the left of the keyboard) will extricate the user from any difficulties.

Good luck!

NOTE: The following symbols have certain meanings:

<word> means a key on the computer keyboard, where *word* is the letter, number or expression on that key.

Where there are letters in **bold**, this denotes an answer that you, the user, should give.

## 2. Installation

If HEX has already been installed, then go to section 3.

If HEX has not yet been installed on your computer, then your computer must have a hard disk with 1.5 megabytes of space free.

To install, follow this procedure:

- 1) Make sure the machine is on, and in the home directory (usually indicated by a 'C:\>'). If it is not in the home directory, type 'CD\'', followed by a <Return> (↵).
- 2) Type 'A:' followed by a <Return>.
- 3) If 'A:\' or some similar prompt appears on the computer screen, then go onto instruction (4). If a message such as 'Not ready reading drive A - Abort, Retry, Fail?' appears, it means that the drive in which the installation disk has been placed is not called 'A:'. In this case, type A, and then repeat instruction 2, but typing 'B:' instead of 'A:'.
- 4) Type 'installh' followed by a <Return>.
- 5) Installation will now take place. When it has finished, you will be automatically placed in the correct directory to run HEX.

### 3. Beginning a Consultation with HEX

To begin a consultation with HEX, you must be in the correct directory. This will usually be called C:\HEX>. If you are unsure of whether or not you are in the correct directory, just type HEX, followed by a <Return> or <Enter>. If you are in the wrong directory, then a message will appear: "Bad command or file name". To get into the right directory, type 'CD\HEX', followed by a <Return>.

Assuming you are in the correct directory, type HEX followed by a <Return> (↵). A message will flash on the screen telling you that you are in the LEONARDO environment (LEONARDO is the computer tool with which the HEX system was built). This is followed by a welcome screen:

---

HEX: Habitat Evaluation Expert System

---

Welcome to HEX, a system for  
the evaluation of pond sites  
with respect to the Great  
Crested Newt.

Written by Mark Cain, 1992

---

Hit any key to continue

---

You are now in HEX and, as the instructions say, you can hit any key to continue.

#### 4. Answering the questions

After hitting a key, the next screen you will encounter after the opening screen will ask you a question. The screen looks like this:

---

HEX: Habitat Evaluation Expert System

---

<f2> EXIT SYSTEM  
<Enter> Choice  
↑↓ Scroll Choice

---

Do you require an introduction to HEX?

---

yes
no

The choice menu containing **yes** and **no** will be the values from which you choose. You will notice that **yes** is dark grey letters on a light grey background, whilst **no** is white lettering on a blue background. Press either of the arrow keys (<↑>, <↓>). The grey background will move accordingly, highlighting the choice. Once you have decided on your choice, press <Return> and the computer will proceed accordingly. In this case choose **yes**, and read the 2 screens of text that briefly introduce HEX.

NOTE: If you hit any wrong keys accidentally, and details appear on the screen that are unwanted, try following guides given with these details, or try pressing <Esc> to get back to the point you require. Alternatively, you can try <f2>. However, this may lead to you leaving the system completely, so only try <f2> after <Esc> has failed. THIS INFORMATION APPLIES THROUGHOUT THE SYSTEM.

## 5. Exiting the HEX System

It is possible to leave the HEX system at many points, especially when a question is being asked, by pressing the <f2> key. You should now be presented with a question about whether you require a complete check of the pond habitat. Try pressing <f2> now, and see what happens.

On pressing <f2> (EXIT SYSTEM), the following screen menu should appear:

```
Yes, QUIT - abandon the system
Don't QUIT - continue with this consultation
Create a CHECKPOINT file on the consultation
Restart using an existing CHECKPOINT file
```

Ignore the bottom two options (they allow you to stop the consultation at any time, save it at that point, and continue with it at a later time).

The top two options are the ones of interest. The top option is the way of leaving the system at whatever point you choose. If you choose the top option, this is effectively the same as ending the consultation, and you will return to the standard operating environment of the computer. If you choose the second option (**don't QUIT**), then HEX returns to the same point where you chose to leave the system.

For now, choose **don't QUIT** - this will take you back to the screen asking if you require a complete check of the pond habitat.

## 6. Question Screens - other features

You have already encountered one question: 'Do you require an introduction to HEX?'. The next question you meet is in the screen:

---

HEX: Habitat Evaluation Expert System

---

<f2> EXIT SYSTEM  
<Enter> Choice  
↑↓ Scroll Choice

---

Do you require a complete check of the suitability of the habitat?

---

yes no	All of the major habitat features will be considered, step by step.
-----------	---

Note the large box on the right of the choice menu. When present, this contains extra information on the choices available. This information may be a further description of each choice, or it may be a guide to the way to answer the given question. Try changing the highlighted option in the menu from **yes** to **no** using an arrow key. You will notice that the information in the box changes to:

yes no	The system will continue assessing until it encounters a habitat feature that is unsuitable.
-----------	--

Notice that the details in the box can be sensitive to the menu option.

Choose **yes**.



7. Getting explanation about why a question is being asked - The 'Why?' Facility

The following screen will appear:

---

HEX: Habitat Evaluation Expert System

---

<f2> EXIT SYSTEM  
<f7> Why?  
<Enter> Choice  
↑↓ Scroll Choice

---

1) Where in the UK is the pond located?

---

Parts of England other than south west  
Scotland  
Wales  
south west England

Those parts of England  
that are not directly  
south of Wales.

Note the question 'Where in the UK is the pond located?' is preceded by a '1'. The number that precedes any question relates to the number of the question when using the standard 'Pond Assessment Form', a questionnaire filled in whilst in the field by HEX users. You may not have encountered this form yet. Do not worry, you will become familiar with such forms when going into the field to do a pond assessment. HEX does not ask questions in a fixed format (current questions are dictated by previous answers), and may 'jump' about relative to the Pond Assessment form. Questions are numbered when asked by HEX so that it is easy to find your answer on the Pond Assessment Form. You may safely ignore this aspect for the time being.

If you move between the choices in the menu, you will notice that the details within the large box change accordingly. Also, you may notice that a new option is available: <f7> Why?. This allows the user of HEX to find out why this question is being asked, and what relationship this question has with the presence/absence of the crested newt. You will find that this option is available for all of the question screens that will follow.

Assume that you want to know why the location in the UK is of importance to the crested newt. Press <f7>. You will now go to a completely new screen headed: 'THE RELATIONSHIP BETWEEN LOCATION AND THE CRESTED NEWT'. You

will see text that tells you about this relationship, and you will notice that several words are highlighted in red and white, and that one of them is flashing ('highland'). These highlighted words (called "buttons") can be selected by the user to give further information. To move between the highlighted words you can use the arrow keys. To select a particular button, press <Enter>.

In this example, choose **soil type** by using the arrow keys to make it the flashing button, and then press <Enter>. A new screen will overlay the previous one, and this will be headed 'UNDERLYING SUBSTRATE AND THE GREAT CRESTED NEWT'. You will notice that it is similar to the previous screen, being a page of text with some words highlighted in red. You can choose between these buttons by using the arrow keys. Select **nutrients**.

A third screen will come up, headed 'NUTRIENTS'. This will tell you what nutrients are.

You will notice that it is different from the previous screens in that it is enclosed within a grey box, and the text is in yellow. The previous type of screen was one that contained information that stated how particular features of the environment affect the crested newt, whereas this type of screen (in a grey box, and with yellow text) is a definition of some ecological entity, idea or concept.

To move back to your original question screen you must move back through each screen you have chosen. You do this by hitting <f2> (QUIT). However, there is no need to go straight back to the question screen. Feel free to explore the 'Why?' facility. NOTE THAT IS PARTICULARLY IMPORTANT THAT YOU EXPLORE THE 'Why?' FACILITY THROUGHOUT YOUR INITIAL FAMILIARISATION WITH HEX.

Remember, the only keys you need to use are the arrow keys to choose between the highlighted buttons, the <Enter> key to select a button, and the <f2> key to quit from a screen.

When you do go back to the original question screen ('Where in the UK is the pond located?'), choose **Parts of England other than south west**.

What follows is a series of question screens, and suitable answers are given below. Feel free to explore the <f7> Why? facility at any point, but please follow the answers given EXACTLY.

On choosing **Parts of England other than south west**, further questions will be asked:

↓  
Is there any suitable habitat within 500 metres of the pond? Choose **yes**.

↓  
Are there any barriers to crested newt movement within 500m of the pond? Choose **no**.

What area of woodland is present within 500m of the pond?  
Choose "> 4000m<sup>2</sup>"

↓

What % of the pond surface is filled by emergent plants?  
Choose "25-50"

↓

What remainder of the % pond surface has floating plants covering it? Choose "25-50"

↓

What remainder of % pond surface has submerged vegetation beneath it? Choose "0"

↓

Is the water very cloudy? Choose **no**

↓

Are there trees/plants shading the pond to any extent?  
Choose **yes**

↓

What is the % of shading of the pond? Choose "26-75"

↓

What is the area of the pond (in metres squared)? Enter 200 (When inputting a large number, remember not to include commas, as the system will not understand this).

↓

What % of the south quarter of the pond is shaded? :  
Choose "50 or greater"

↓

How often does complete drying of the pond occur? Choose **never**

↓

Has a pond dip been performed? Choose **no**

↓

Does a mat of bacteria cover most of the bottom of the pond? Choose **no**

↓

Does there seem to be a reduced number of species in this pond? Choose **unknown**

↓

Do ducks, geese or swans live in, and use, the pond?  
Choose **yes**

↓

How many adult ducks/geese/swans are present in the pond?  
Enter 8

↓

What is the maximum depth of the pond (metres)? Choose "0.3-1"

↓

Are fish present or absent in the pond? Choose **absent**

↓

All features of the habitat have now been considered.

## 8. Conclusions from the Information given

From the information you have given, you will find that HEX concludes that the site is unsuitable. You will find a screen that looks like this:

---

HEX: Habitat Evaluation Expert System

---

The site is unsuitable.

The following features were deemed unsuitable:

degree of shading duck numbers Aquatic habitat
--

If you require further information about why these features are not suitable, then select one or more of them using:
--

<p>&lt;Ins&gt; Add to List &lt;Del&gt; Remove from List</p>
---

The box on the upper right contains the list of features that have been judged unsuitable by HEX. For example, duck numbers are considered unsuitable because you input the answer 200 to question 9: 'What is the area of the pond (in metres squared)?', and answered yes to question 12: 'Do ducks, geese or swans live in, and use, the pond?' The presence of any water fowl in such a small pond is likely to have an effect on any potential newt population.

As explained in the lower, left-hand box, if you wish for further information about the features judged unsuitable, then you can choose them by moving the grey highlight over your choices, and pressing <Ins>. Choices made will flash, and if you want to cancel any choices then move the highlight over that choice, and press <Del>. If you do not wish for any further explanation, just hit <Return> without selecting any choices from the menu.

Depending on your answers in future consultations, you will find that the system will judge a pond as 'suitable' or 'unsuitable' or 'provisionally suitable' (ie judged as suitable, as long as certain criteria are satisfied, such as fish are absent), and give a list of parameters that have been used to assign this status.

For now, choose any two (or more) from the list using <Ins>, and press <Return>. This will be followed by two (or more) screens that explain how your choices affect the crested newt. After these explanations, you will meet the final HEX screen.

## 9. Continuing the Consultation

You will now encounter the final HEX screen. This will ask the question: 'Do you wish to continue with this consultation?' If you choose **continue**, then the HEX system will start over again, and will perform as before, but without the introduction screens. If you choose **stop**, then you will leave the HEX system, and return to the computer's main operating system.

For now, choose **stop**.

## 10. Removing HEX from the Computer

If you wish to remove HEX from your computer, it is possible to do so using the installation disk, following these instructions:

1) Insert the installation disk into the machine, and type 'A:' (or if this does not work, 'B:'), followed by <Return>.

2) Type 'remhex', followed by <Return>.

After several seconds, the HEX system will be removed.

You are now familiar with the complete workings of HEX.

**APPENDIX E**  
**USER GUIDE FOR TRITON**

# **A GUIDE AND INTRODUCTION TO TRITON:**

## **A POND EVALUATION EXPERT SYSTEM**

BY MARK F. CAIN, 1992

SCHOOL OF COMPUTING AND MATHEMATICAL SCIENCES

DE MONTFORT UNIVERSITY LEICESTER,

THE GATEWAY,

LEICESTER,

LE1 9BH.

## 1. An Introduction to TRITON

The pond evaluation expert system, TRITON, is a computer-based system that assesses the habitat of a pond and its surroundings. It performs this task in order to decide whether or not the site is able to support a colony of the great crested, or warty, newt (*Triturus cristatus*). It tackles this problem by asking questions of the computer user, so as to acquire information from which it can decide a status for the pond's suitability.

In addition to asking questions and deciding upon the status of a pond site, the TRITON system can explain to the user why any particular question is being asked, and why certain features are of relevance to the crested newt.

The easiest way to become familiar with the TRITON system is by using it. However, this guide is intended as an introduction to its facilities, and covers all aspects of the system. The guide should only be used in conjunction with the TRITON system, and reading the guide separately from the TRITON system will be of little benefit. Immediately after using this guide, the user should experiment and 'play' with the system to become more familiar with its operation.

Good luck!

NOTE: The following symbols have certain meanings:

<word> means a key on the computer keyboard, where word is the letter, number or expression on that key.

Where there are letters in **bold**, this denotes an answer that you, the user, should give.



## 2. Installation

If TRITON has already been installed, go to section 3.

If TRITON has not yet been installed on your computer, then your computer must have a hard disk with about half a megabyte space free.

To install, follow this procedure:

- 1) Make sure the machine is on, and in the home directory (usually indicated by a 'C:\>'). If it is not in the home directory, type 'CD\'', followed by a <Return> (↵).
- 2) Type 'A:' followed by a <Return>.
- 3) If 'A:\' or some similar prompt appears on the computer screen, then go onto instruction (4). If a message such as 'Not ready reading drive A - Abort, Retry, Fail?' appears, it means that the drive in which the installation disk has been placed is not called 'A:'. In this case, type A, and then repeat instruction 2, but typing 'B:' instead of 'A:'.
- 4) Type 'installt' followed by a <Return>.
- 5) Installation will now take place. When it has finished, you will be automatically placed in the correct directory to run TRITON.

### 3. Beginning a Consultation with TRITON

To begin a consultation with TRITON, you must be in the correct directory. This will usually be called C:\TRITON>. If you are unsure of whether or not you are in the correct directory, just type TRITON, followed by a <Return> or <Enter>. If you are in the wrong directory, then a message will appear: "Bad command or file name". To get into the right directory, type 'CD\TRITON', followed by a <Return>.

Assuming you are in the correct directory, type TRITON followed by a <Return> (↵). You will then encounter the welcome screen:

TRITON		
<table border="1"><tr><td>Welcome to TRITON, a system for the evaluation of pond sites with respect to the great crested newt.</td></tr><tr><td>Written by Mark Cain, 1992</td></tr></table>	Welcome to TRITON, a system for the evaluation of pond sites with respect to the great crested newt.	Written by Mark Cain, 1992
Welcome to TRITON, a system for the evaluation of pond sites with respect to the great crested newt.		
Written by Mark Cain, 1992		
Page 1		

Hit any key to continue...

You are now in TRITON. Hit any key to continue.  
You will now encounter an introduction screen, saying:

Introduction to TRITON
<p>The great crested newt (TRITURUS CRISTATUS) is currently an endangered species, threatened by pollution, the use of intensive methods of agriculture, and the loss of pond sites.</p> <p>This expert system is geared to the evaluation of a pond and its surrounding habitat, and its general aim is to assess if a site is one suitable for the crested newt.</p> <p>The quickest way to become familiar with the workings of this system is by using and experimenting with it. If you are a complete novice, then it is advisable to use the user guide/introduction to TRITON in conjunction with the system.</p>

Hit another key to continue.

#### 4. Answering the questions

After the introduction screen, there will a few moments to wait while data is loaded into the system. This will be indicated by a red flashing message, saying 'Please wait while data loads'. The next screen you will encounter will ask you a question. The screen looks like this:

TRITON
--------

10) Are fish present in the pond?			
<table border="1"><tr><td>present</td></tr><tr><td>absent</td></tr><tr><td>unknown</td></tr></table>	present	absent	unknown
present			
absent			
unknown			

<f2> Define    <f3> Explain    <f4> Conclude    <f10> Quit

The choice menu containing **present**, **absent** and **unknown** will be the values from which you choose. You will notice that the word **present** consists of light blue letters on a light grey background, whilst **absent** and **unknown** is white lettering on a blue background. Press the 'down' arrow key <↓>. The grey background will move accordingly, highlighting the lower choices. To move up the list, you can use the 'up' <↑> arrow.

Note the question 'Are fish present in the pond?' is preceded by a '10'. The number that precedes any question relates to the number of the question when using the standard 'Pond Assessment Form', a questionnaire filled in whilst in the field by TRITON users. You may not have encountered this form yet. Do not worry, you will become familiar with such forms when going into the field to do a pond assessment. TRITON does not ask questions in a fixed format (current questions are dictated by previous answers), and may 'jump' about relative to the Pond Assessment form. Questions are numbered when asked by TRITON so that it is easy to find your answer on the Pond Assessment Form. You may safely ignore this aspect for the time being.

Once you have decided on your choice, press <Return> and the computer will proceed accordingly. In this case choose **unknown**.

## 5. Exiting the TRITON System

It is possible to leave the TRITON system at many points, especially when a question is being asked, by pressing the <f10> 'Quit' key. You should now be presented with a question about whether the pond dries up completely. Try pressing <f10> now, and see what happens.

On pressing <f10> (QUIT), the following question should appear:

Do you wish to go through another run?

If you select *yes* ('y'), then the consultation begins again. Selecting *no* ('n') takes you out of the TRITON environment.

For now, choose 'y' - this will take you back to the screen asking if fish are present in the pond.

## 6. Question Screens - other features

You should now have in front of you the screen below:

TRITON
--------

10) Are fish present in the pond?			
<table border="1"><tr><td>present</td></tr><tr><td>absent</td></tr><tr><td>unknown</td></tr></table>	present	absent	unknown
present			
absent			
unknown			

<f2> Define    <f3> Explain    <f4> Conclude    <f10> Quit

Note that at the bottom of the screen, there are several options available with 'f' keys (the 'f-keys' usually run along the top of the keyboard). Try pressing <f2> 'define'.

On pressing this, the 'define' option will provide extra information about the subject of the question. This is not necessarily restricted to a definition. In this case, for example, there is advice about how to assess if fish are present or not. Other sorts of information that may be given include more precise descriptions of the choices available for each question, a definition of the subject of the question, and so on.

Read the 'definition' about fish presence, and see if you think it may be helpful.

When you have finished, hit any key to continue.

You may also note that there is an <f4> 'conclude' option. With this option, you are asking the system to come to some conclusions using only the information it already has. This means it will ask no further questions. For now, choose the <f4> option.

You will find that TRITON will state that it can not come to a reasonable value from the information given. This is not surprising, as you have not provided TRITON with any information yet!

Press a key to continue. The system will again ask if you wish to go through another run. Choose yes ('y').

7. Getting explanation about why a question is being asked - The 'Explanation?' Facility

You should still be on the screen asking about fish presence. If you are not on this screen, press <f10>, followed by 'y', and you will find yourself there after a few moments.

A further facility that is available with the questions is the <f3> 'Explain' facility. On selecting this, the system will explain the relevance of the question to assessing the suitability of a pond to support a crested newt colony. Choose <f3> 'explain'.

TRITON now provides a description of the relationships that exist between fish and the suitability of a pond to support crested newts.

For example, the first screen will note that, in some cases, the presence of fish implies that a pond is suitable to support a crested newt population, as their presence indicates pond productivity is high enough to support crested newts. It goes on to note that in many cases, however, fish predate crested newt larvae to such an extent that they exclude a viable crested newt colony.

The explanation provided may often go over several screens. However, it is often worthwhile to read these screens, so as to understand the interactions that are likely to occur in a real pond site.

You press any key to move forward a screen. Eventually, you will return to the original question screen.

What follows is a series of question screens, with appropriate answers. Please use the <f2> *define* and <f3> *explain* facilities as much as possible, to see if they aid in selecting the answers or explaining why questions are being asked. Please follow the answers given EXACTLY.

Are fish present in the pond? Choose **absent**.

↓

Does the pond ever dry up completely? Choose **1-4 times per decade**.

↓

Is there any suitable habitat within 500 metres of the pond? Choose **present**.

↓

Is agricultural runoff going into the pond? Choose **present**.

↓

Is the pond next to a steep slope barren of vegetation? Choose **present**.

↓

Are there signs of sewage dumping/organic rubbish in the pond? Choose **unrecorded**.

↓

What is the size of duck numbers relative to pond size? Choose **large**.

↓

The system should now reach a conclusion.

## 8. Conclusions from the Information given

From the information you have given, you will find that TRITON concludes that there is strong evidence that the site is unsuitable. You will find a screen that looks like this:

TRITON

There is strong evidence to suggest that pond habitat is unsuitable to support a crested newt population.

Contributing evidence includes:

Barren slope adjacent to the pond has the value "present".

Pond duck numbers has the value "large".

Agricultural runoff into the pond has the value "present".

<f2> Define <f3> Explain <f5> "What-if" (Change Values)

The three parameters given are a set of accumulated evidence that the pond is not suitable to support a viable crested newt colony. You will note that there are three 'f-key' facilities available at this point.

<f2> 'define' will further explain details about a pond habitat's suitability for crested newts.

<f3> 'explain' will explain how each of the parameters relates to a pond's suitability to support a viable crested newt colony. (eg a barren slope indicates that runoff from the slope will occur, causing pond turbidity, and reducing pond productivity). Use the <f3> facility now, and read how the system came to its conclusions. On finishing, you will return to the above screen.

The "what-if" facility <f5> is a means to further investigate the pond site. It allows users to adjust the values of certain elements that are considered within TRITON (eg pond location, shading, etc). Select <f5>.

Notice that a menu comes up, with several selections. Press, and keep pressing, the down arrow (↓). You will see that there are many possible elements that are available for change. Go to the top of the list (using the 'up' <↑> arrow), and select the first element "pond shading", by highlighting it, and pressing <Enter>.

You will be asked "What is the percentage shading of the pond?". Choose "> 75%". On making the selection, you will be returned to the larger menu of elements. You may change any number of these, but for now, leave the facility by pressing <Esc>.

The machine will take several moments to reformulate its



information, and then come to a conclusion based upon the new evidence. In this case, there will be no change in the original conclusions.

To leave the session, hit any key. You will encounter the question 'do you wish to go through another run?', to which you answer 'no' ('n'). You will then leave the TRITON environment.

#### 9. Removing TRITON from the Computer

If you wish to remove TRITON from your computer, it is possible to do so using the installation disk, following these instructions:

1) Insert the installation disk into the machine, and type 'A:' (or if this does not work, 'B:'), followed by <Return>.

2) Type 'remtrit', followed by <Return>.

After several seconds, the TRITON system will be removed.

You are now familiar with the complete workings of TRITON.

**APPENDIX F**  
**THE POND ASSESSMENT FORM USED WITH HEX AND TRITON**

### **Pond Assessment Form: Introduction**

This pond assessment form is for use with HEX/TRITON. It contains a questionnaire to be filled, on site.

When using this questionnaire, it is suggested that the user take a copy of an ordinance survey map of the area around the pond (at 1:10,000 or 1:25,000), upon which there should be drawn a circle of radius 500 metres, which centres on the pond. This circle can be shaded/drawn over, to act as an aid in estimating areas of suitable habitat (ie woodland, scrub, rough grass), and in identifying significant land features (eg busy roads, wide or fast moving rivers, walls, etc). An asterisk (\*) is placed before those questions where shading/sketching is appropriate.

**POND ASSESSMENT FORM**

Name of Recorder/Date:

Name/ID of pond (if any):

OS Coordinates (if available):

**(1) Location**

Parts of England other than south west	<input type="checkbox"/>	GO TO (3)
Scotland	<input type="checkbox"/>	
Wales	<input type="checkbox"/>	GO TO (3)
South west England (parts of		
England directly south of Wales)	<input type="checkbox"/>	GO TO (3)

**(2) Location in Scotland**

North Scotland	<input type="checkbox"/>
West Scotland	<input type="checkbox"/>
Other part of Scotland	<input type="checkbox"/>

**(3) Suitable Terrestrial Habitat**

Is there any suitable terrestrial habitat within 500 metres of the pond? (Suitable terrestrial habitat includes woodland of any type, scrub, rough grass, hedges, and ditches with vegetation present)

Yes	<input type="checkbox"/>	
No	<input type="checkbox"/>	GO TO (6)
Unknown	<input type="checkbox"/>	

**(4) Barriers Present**

Are there any potential barriers to crested newt movement within 500 metres of the pond? (Such barriers include roads, walls, streams or rivers, tarmaced areas, ploughed fields, buildings, cliffs, salt water, and so on)

Yes	<input type="checkbox"/>	
No	<input type="checkbox"/>	GO TO (5)
Unknown	<input type="checkbox"/>	

Are there any of the following within 500 metres of the pond:

**\* (4a) Roads?**

Yes	<input type="checkbox"/>	—————>	4a.2) Are any motorways, A-roads, or roads wider than 50 metres present that run lateral to the pond?
No	<input type="checkbox"/>		

Yes	<input type="checkbox"/>
No	<input type="checkbox"/>

**\* (4b) Walls?**

Yes	<input type="checkbox"/>	—————>	4b.2) Are any of these walls likely to be impenetrable to the crested newt (such as walls made of bricks
No	<input type="checkbox"/>		

and  
gaps,  
must  
Dry

mortar or the like, with no  
doorways, or holes: such walls  
also be lateral to the pond).

stone walls are penetrable.

Yes	<input type="checkbox"/>
No	<input type="checkbox"/>

**\* (4c) Insurpassable Rivers/streams?**  
Yes [ ] (Insurpassable means those likely to be too  
No [ ] fast/large for an adult crested newt to be  
able to successfully cross, and which run  
laterally to the pond)

**\* (4d) Buildings?**  
Yes [ ]  4d.2) Are there any buildings that a  
No [ ] crested newt would not be able to  
penetrate or circumnavigate?  
Yes [ ]  
No [ ]

**\* (4e) Recently-ploughed fields?**  
Yes [ ]  4e.2) In traversing a ploughed field,  
No [ ] would the newt have to cover >100m?  
Yes [ ]  
No [ ]

**\* (4f) Large, dry expanses to land that are not prone to  
puddle formation?**  
Yes [ ]  In traversing a dry expanse of land,  
No [ ] would the newt have to cover >50m?  
Yes [ ]  
No [ ]

**\* (4g) Any other land features that would act as barriers  
to crested newt movement (such as cliffs, salt water  
inlets, high banks, etc)**  
Yes [ ]  Note what such barriers are, and how  
No [ ] many of each there are:

**(5) Terrestrial Habitat**

Please state the area/length of each of the following  
terrestrial habitat features that are available to the  
crested newt colony of the pond. 'Available' means  
(i) within 500 metres of the pond's edge, and (ii) not  
made inaccessible by barriers noted in right hand side of  
4a-4g (if any exist). The best method of performing such  
estimates is by sketching a 500m radius circle drawn  
around the pond on a scale map (from the centre of the  
pond).

NB: As an aid to estimation, note that the area of a  
tennis court is about 260m<sup>2</sup>, whilst a small (park)  
football pitch is about 4,000m<sup>2</sup>.

**\* (5a) What area (m<sup>2</sup>) of woodland is available?**  
>4,000m<sup>2</sup> [ ] GO TO (6)  
1-4,000m<sup>2</sup> [ ]  What area is available?  
0 [ ]

\* (5b) What area (m<sup>2</sup>) of scrub is available (scrub is vegetation that includes grass, herbs, shrubs, and possibly bushes and scattered trees)?

>4,000m<sup>2</sup> [ ] GO TO (6)  
1-4,000m<sup>2</sup> [ ] \_\_\_\_\_> What area is available?  
0 [ ]

\* (5c) What area (m<sup>2</sup>) of rough grass is available (rough grass usually has clumps of uneven vegetation present)?

>4,000m<sup>2</sup> [ ] GO TO (6)  
1-4,000m<sup>2</sup> [ ] \_\_\_\_\_> What area is available?  
0 [ ]

\* (5d) What length of suitable hedges (ie with base going to ground, or with uncropped vegetation beneath) is available (m)?

>=2000m [ ] GO TO (6) \_\_\_\_\_> What length is available?  
1-2000m [ ]  
0 [ ]

\* (5e) What length of vegetated ditches is available (m)?

>=2700m [ ] GO TO (6)  
1-2700m [ ] \_\_\_\_\_> What length is available?  
0 [ ]

## (6) Shading

(6a) Is there shading of the pond by trees?

Yes [ ] GO TO (6c)  
No [ ]

(6b) Is there shading of the pond by any objects?

Yes [ ]  
No [ ] GO TO (7)

(6c) What is the percentage shading of the pond's total surface?

< 25% [ ] GO TO (7)  
26-75% [ ]  
> 75 [ ] GO TO (7)

(6d) What percentage of the southern quarter of the pond's total surface is shaded (NB: the sun is in the east in the morning, south at midday, and west in the evening)?

< 50 % [ ]  
≥ 50 % [ ]

**(7) Macrophyte Complement of the Pond**

(7a) What percentage of the pond surface is filled by emergent vegetation (to nearest 5%)? (NB emergent vegetation has significant aerial parts, such as reed beds)

(7b) What remainder (after considering emergent vegetation) of the pond surface is covered by floating plants (to nearest 5%)? (Examples of floating plants include duckweed and lilies)

(7c) What remainder (after that occupied by emergent and floating vegetation) of the pond surface has submerged vegetation beneath it (to nearest 5%)?

**(8) Pond Depth and Size**

(8a) What is the maximum pond depth?

- < 0.3 metres [ ]
- 0.3-1 metres [ ]
- > 1 metres [ ]
- Unknown [ ]

(8b) What is the area of the pond (m<sup>2</sup>)?

(Note: The area of a tennis court is about 260m<sup>2</sup>)

For more or less elliptical ponds, the area is given by:

$$\frac{\text{length} \times \text{width} \times 3.14}{4}$$

**(9) Drawdown**

Does the pond ever dry up completely?

- Never [ ]
- 1-4 times per decade [ ]
- > 4 times per decade [ ]
- Unknown [ ]

**(10) Fish Presence**

Are fish present?

- Present [ ]
- Absent [ ]
- Unknown [ ]

**(11) Duck Presence**

(11a) Do ducks, swans, or geese live in, and use, the pond?

- Yes [ ]
- No [ ] GO TO (12)

(11b) How many adult ducks, swans, or geese present?

**(12) Turbidity**

(12a) Is the water very cloudy (so that the bottom cannot be seen except in very, very shallow parts)?

Yes [ ]  
No [ ] GO TO (13)

(12b) Do any large animals have access to the pond?

Yes [ ]  
No [ ]

(12c) Is the pond next to a steep slope that is relatively barren of vegetation?

Yes [ ]  
No [ ]

**(13) Pollution**

(13a) Has a pond dip been performed by you?

Yes [ ]  
No [ ] GO TO (13c)

(13b) Are any of the following species present in the pond dip - gammarids, mayfly larvae, or dragonfly larvae?

Yes [ ] GO TO (14)  
No [ ]

(13c) Is a bacterial mat lying suspended over much of the bottom of the pond? (This is often coloured, and possibly translucent, and makes the water appear milky)

Yes [ ]  
No [ ] GO TO (13g)

(13d) Are dead leaves in the pond?

No [ ]  
In reasonable amounts [ ]  
In large amounts [ ]

('Large amounts' means that the leaves form a barrier which obscures 40% or more of the pond's bottom)

(13e) Are there signs of sewage dumping in the pond?

Yes [ ]  
No [ ]  
Unknown [ ]

(13f) Do any large animals have access to the pond?

Yes [ ]  
No [ ]

(13g) Does there seem to be a fewer species in this pond than you would expect?

Yes [ ]  
No [ ] GO TO (14)  
Unknown [ ]



(13h) Is any crop agriculture in close proximity to the pond where there is a downward slope into the pond, with no barrier of vegetation, or known drainage from more distant agricultural sources?

Yes [ ]

No [ ]

(13i) Is there a road within 10 metres of the pond?

Yes [ ]

No [ ]

**(14) This form is complete.**

**APPENDIX G**  
**EVALUATION QUESTIONNAIRES**

Part 1 - Questionnaires for HEX/TRITON provided to  
Student/Expert Evaluators

Where specific questions were only given to students or experts, the question is followed by brief details.

Evaluation form for HEX/TRITON  
To be filled in by the User

*This questionnaire is designed to provide feedback on your thoughts of the HEX/TRITON computer system. Your answers will be of much value for the purposes of evaluation and improvement of this system, and any comments you may wish to make, however minor you may feel they are, will be greatly appreciated. Any personal details given here will be kept in the strictest confidence, and will only be used in the research at hand, and in the improvement of the HEX/TRITON system for future users.*

Thank you for your time and effort.

Section A: Personal Details

Date:

Name:

Age (years):

Place of Work/Address:

Job Description:

Telephone Number (Daytime & Evening):

(1) How familiar are you with using a computer (please tick appropriate box)?

I use computers frequently	[ ]
I use computers occasionally	[ ]
I use computers rarely	[ ]
I never use computers	[ ]

(2) Qualifications in ecological/environmental science:

None	[ ]	
GCSE/'O'level/CSE standard only	[ ]	
'A' level/HND/BTEC standard only	[ ]	
Currently doing Degree	[ ]	—> Which year?
		[ ]
Post-Graduate	[ ]	

If your educational qualifications extend beyond a first degree, please give brief details (continue overleaf, if necessary):

(3) Do you have any experience in ecological/environmental management or assessment?

No [ ]

Yes [ ]

If 'yes', give details (continue overleaf, if necessary):

Section B: Questions about HEX/TRITON (Select one choice per question)

**(1) Handbook**

Did the handbook used in conjunction with the HEX/TRITON system give a good overview and introduction to HEX/TRITON?

- Definitely [ ]
- Mostly [ ]
- For a reasonable part [ ]
- Only some of the time [ ]
- Not at all [ ]

**(2) User Interface**

(2a) Do you feel that the various screen layouts you encountered were acceptable?

- Definitely [ ]
- Mostly [ ]
- For a reasonable part [ ]
- Only some of the time [ ]
- Not at all [ ]

(2b) Do you feel it was easy to examine the information within HEX/TRITON?

Definitely	[ ]
Mostly	[ ]
For a reasonable part	[ ]
Only some of the time	[ ]
Not at all	[ ]

(2c) Was the HEX/TRITON system easy to learn?

Definitely	[ ]
Mostly	[ ]
For a reasonable part	[ ]
Only some of the time	[ ]
Not at all	[ ]

(2d) Did HEX/TRITON require a high degree of concentration to use?

Definitely	[ ]
Mostly	[ ]
For a reasonable part	[ ]
Only some of the time	[ ]
Not at all	[ ]

(2e) Would you rank the overall 'usability' of HEX/TRITON as high?

Definitely	[ ]
Mostly	[ ]
For a reasonable part	[ ]
Only some of the time	[ ]
Not at all	[ ]

### (3) Justification/Explanation Facilities

(3a) Did you find the justifications of questions (The '<f7> Why?/<F3> Explain?' facility), when you used them, were clear and understandable?

Definitely	[ ]
Mostly	[ ]
For a reasonable part	[ ]
Only some of the time	[ ]
Not at all	[ ]

(3b) Were the details sometimes given to the right of your list of choices/in '<F2> Define' selection for each question useful in making your choice?

Definitely	[ ]
Mostly	[ ]
For a reasonable part	[ ]
Only some of the time	[ ]
Not at all	[ ]

(3c) Did you find the final explanations of pond suitability or non-suitability for the crested newt clear and understandable?

Definitely	[ ]
Mostly	[ ]
For a reasonable part	[ ]
Only some of the time	[ ]
Not at all	[ ]

(3d) Do you feel that the overall system gave useful information?

Definitely	[ ]
Mostly	[ ]
For a reasonable part	[ ]
Only some of the time	[ ]
Not at all	[ ]

(3e) Do you feel the level of detail was suitable?

Definitely	[ ]
Mostly	[ ]
For a reasonable part	[ ]
Only some of the time	[ ]
Not at all	[ ]

#### (4) Education

Do you feel you have gained any insight into the pond ecosystem whilst using HEX/TRITON?

Definitely	[ ]
Mostly	[ ]
For a reasonable part	[ ]
Only some of the time	[ ]
Not at all	[ ]

#### (5) Accuracy

(5a) Do you think that the assessment of whether the pond was suitable or unsuitable was acceptably accurate?  
(Experts Only)

Yes	[ ]
Yes, with notable exceptions	[ ]
No	[ ]

If 'yes, with notable exceptions', or 'no', please give details (continue overleaf, if necessary):

(5a.ii) Do you think the strength of conviction that the system expressed in the conclusion (eg 'strong evidence for...', 'some evidence for...', etc) was correct (**Experts evaluating TRITON Only**) :

Yes	[ ]
Yes, with few exceptions	[ ]
No, overconfident in conviction	[ ]
No, too weak in conviction	[ ]
No, too strong and too weak at different times	[ ]
Don't know	[ ]

If 'yes, with few exceptions', or 'No,...', please give brief details (continue overleaf, if necessary):

(5b) Do you think the explanations of the final conclusions were correct? (**Experts Only**)

Yes	[ ]
Yes, with notable exceptions	[ ]
No	[ ]
Didn't understand them	[ ]
Don't know	[ ]

If 'yes, with notable exceptions', or 'no', please give details (continue overleaf, if necessary):

(5c) Do you think the explanations of the final conclusions were clear? (**Experts Only**)

Yes	[ ]
Yes, with notable exceptions	[ ]
No	[ ]

If 'yes, with notable exceptions', or 'no', please give details (continue overleaf, if necessary):

(5d) Do you think the explanations of the final conclusions were clear? (**Experts Only**)

Yes [ ]  
Yes, with notable exceptions [ ]  
No [ ]

If 'yes, with notable exceptions', or 'no', please give details (continue overleaf, if necessary):

**(6) Robustness**

Do you feel that the system is able to handle a wide variation of cases? (**Experts Only**)

Yes [ ]  
Yes, with notable exceptions [ ]  
No [ ]

If 'yes, with notable exceptions', or 'no', please give details (continue overleaf, if necessary):

**(7) The Suitability of the Questioning Strategy**

(7a) Do you feel that the overall ordering of questions is acceptable for gathering information for the evaluation of a pond in terms of suitability for the crested newt? (**Experts Only**)

Yes [ ]  
Yes, with notable exceptions [ ]  
No [ ]  
Don't know [ ]

If 'yes, with notable exceptions', or 'no', please give details (continue overleaf, if necessary):



(7b) Do you feel all of the questions are worded in an understandable way? (**Experts Only**)

Yes [ ]  
Yes, with notable exceptions [ ]  
No [ ]

If 'yes, with notable exceptions', or 'no', please give details (continue overleaf, if necessary):

(7c) Do you think that all of the questions are relevant (to pond evaluation with respect to the crested newt)? (**Experts Only**)

Yes [ ]  
Yes, with notable exceptions [ ]  
No [ ]

If 'yes, with notable exceptions', or 'no', please give details (continue overleaf, if necessary):

**(8) Errors and Omissions**

(8a) Are there any obvious omissions of relevant questions in the system? (**Experts Only**)

No [ ]  
Yes [ ]

If 'yes', please give details (continue overleaf, if necessary):

(8b) Are there any incorrect facts/details given anywhere within the system? (**Experts Only**)

No                    [   ]  
Yes                    [   ]

If 'yes', please give details (continue overleaf, if necessary):

(8c) Is any incorrect advice/explanation given by the system? (**Experts Only**)

No                    [   ]  
Yes                    [   ]

If 'yes', please give details (continue overleaf, if necessary):

(8d) Are there any errors not already noted in the previous sections within the system? (**Experts Only**)

No                    [   ]  
Yes                    [   ]

If 'yes', please give details (continue overleaf, if necessary):

**(9) Improvements**

Have you any suggestions for improvements to the system, including criticisms of the present system, possible additional facilities you would like to see, and so on?

No           [    ]  
Yes           [    ]

If 'yes', please give details (continue overleaf, if necessary):

Section C: Final Comments

If you have any comments about the HEX/TRITON system, however minor, please feel free to state them in the box below (continue overleaf, if necessary). This may include criticisms or praise of the system that could not be properly expressed in the questionnaire, suggestions for extending the system for a greater range of tasks, and any comments about the handbook and this questionnaire:

Thank you for your time and effort in filling out this questionnaire. The information is gratefully received, and will be utilised to improve the HEX/TRITON system for future users.

Part 2 - Questionnaires for Recording Expert Evaluator  
Opinions of Default Explanations of HEX and TRITON

The Evaluation of Explanations

Name:

Place of work/address:

Please imagine you are using several computer systems that evaluate ponds to assess whether they are suitable to support a viable colony of crested newts. Each system may give different information, and this exercise is to see how well these systems can explain themselves.

Note the focus of this part of the evaluation process is not to test or assess your own abilities, but to evaluate the success of different approaches to explanation, which may be utilised in the future. We are attempting to assess whether these explanations are acceptable to target users. The target users for these systems fall into two main groups:

(1) **Educational** - students that are using these systems to acquire ecological and other knowledge.

(2) **Practical** - people using these systems to evaluate ponds for practical purposes. Examples of these would be conservation workers, habitat managers, pond recorders, and so on.

These groupings are referred to throughout the evaluation.

At the end of this stage of the evaluation, please note any comments you may wish to make in the box below, and continue overleaf if necessary.

Thank you for your participation.

## Section 1

You are asked 'Where in the UK is the pond?'. You wish to know how the pond's location relates to the pond's suitability to support crested newts, and request some explanation of why the above question is being asked. Using two different computer systems, you are presented with the following two sets of information:

---

### **Explanation A**

*if the pond's location is in Scotland  
then the pond location is unsuitable*

*if the pond location is unsuitable  
or the terrestrial habitat is unsuitable  
or the aquatic vegetation within the pond is unsuitable  
or the pond's depth and size is unsuitable  
or the fish status of the pond is unsuitable  
or the duck status of the pond is unsuitable  
or pond shading is unsuitable  
or pond permanence/frequency of drying is unsuitable  
or pollution levels are unsuitable  
then the pond is not suitable to support crested newts*

---

### **Explanation B**

*The pond location being Scotland implies that the pond habitat is unsuitable to support a crested newt population, in about half of the cases. This is derived from:*

*A pond located in Scotland indicates that the pond's mean temperature is low, in most cases.*

*The pond's mean temperature being low indicates pond productivity is low, in most cases.*

*Pond productivity being low indicates the pond habitat is unsuitable to support a crested newt population, in many cases.*

---

i) **Comprehensibility:** From the details given in the explanations, do you understand how the pond's location and the pond's suitability to support a viable crested newt colony are related (tick one box from each column)?

	A	B
Definitely	[ ]	[ ]
Mostly	[ ]	[ ]
For a reasonable part	[ ]	[ ]
Partially	[ ]	[ ]
Not at all	[ ]	[ ]

ii) **Clarity:** How clearly do you think each of these explanations expresses the relationship between the pond's location and the pond's suitability to support a viable crested newt colony?

	A	B
Extremely clear	[ ]	[ ]
Very clear	[ ]	[ ]
Reasonably clear	[ ]	[ ]
Not very clear	[ ]	[ ]
Not clear at all	[ ]	[ ]

iii) **Accuracy:** Do you consider that these explanations are accurate (in considering the pond's location and pond suitability to support crested newts)?

	A	B
Definitely	[ ]	[ ]
Mostly	[ ]	[ ]
For a reasonable part	[ ]	[ ]
Partially	[ ]	[ ]
Not at all	[ ]	[ ]
Don't know	[ ]	[ ]

iv) **Utility:** How would you rank the utility of the information given by each of the explanations for *educational* and *practical* users?

Educational Users			Practical Users		
	A	B		A	B
Definitely useful	[ ]	[ ]	Definitely useful	[ ]	[ ]
Mainly useful	[ ]	[ ]	Mainly useful	[ ]	[ ]
Reasonably useful	[ ]	[ ]	Reasonably useful	[ ]	[ ]
Only partly useful	[ ]	[ ]	Only partly useful	[ ]	[ ]
Not at all useful	[ ]	[ ]	Not at all useful	[ ]	[ ]

v) **Detail:** Do you think the explanations are of sufficient detail to enable *educational* and *practical* users to understand the complexities of the relationship between the pond's location and pond suitability to support crested newts?

Educational Users			Practical Users		
	A	B		A	B
Definitely	[ ]	[ ]	Definitely	[ ]	[ ]
Mostly	[ ]	[ ]	Mostly	[ ]	[ ]
In the main	[ ]	[ ]	In the main	[ ]	[ ]
Partially	[ ]	[ ]	Partially	[ ]	[ ]
Not at all	[ ]	[ ]	Not at all	[ ]	[ ]

vi) **Preference:** Which explanation do you prefer?

A	[ ]
B	[ ]
Neither	[ ]

Section 2

You are asked 'What is the area of the pond (m<sup>2</sup>)?'. You wish to know how the area of the pond relates to the pond's suitability to support crested newts, and request some explanation of why the above question is being asked. Using two different computer systems, you are presented with the following two sets of information:

---

**Explanation C**

The pond's area being <20m<sup>2</sup> implies that the pond habitat is unsuitable to support a crested newt population, in about half of the cases. This is derived from:

The pond's area being <20m<sup>2</sup> indicates the pond's likelihood of drying up is greater than 4 times per decade, in about half of the cases.

The pond's likelihood of drying up being greater than 4 times per decade indicates the pond is not suitable for crested newt larvae, in virtually all cases.

The pond being unsuitable for crested newt larvae indicates the pond habitat is unsuitable to support a crested newt population.

---

**Explanation D**

if the pond's area is less than 20m<sup>2</sup>  
and the maximum depth has any value  
then the pond's depth and size is unsuitable

if the pond location is unsuitable  
or the terrestrial habitat is unsuitable  
or the aquatic vegetation within the pond is unsuitable  
or the pond's depth and size is unsuitable  
or the fish status of the pond is unsuitable  
or the duck status of the pond is unsuitable  
or pond shading is unsuitable  
or pond permanence/frequency of drying is unsuitable  
or pollution levels are unsuitable  
then the pond is not suitable to support crested newts

---

i) **Comprehensibility:** From the details given in the explanations, do you understand how the area of the pond and the pond's suitability to support a viable crested newt colony are related (tick one box from each column)?

	C	D
Definitely	[ ]	[ ]
Mostly	[ ]	[ ]
For a reasonable part	[ ]	[ ]
Partially	[ ]	[ ]
Not at all	[ ]	[ ]

ii) **Clarity:** How clearly do you think each of these explanations expresses the relationship between the area of the pond and the pond's suitability to support a viable crested newt colony?

	C	D
Extremely clear	[ ]	[ ]
Very clear	[ ]	[ ]
Reasonably clear	[ ]	[ ]
Not very clear	[ ]	[ ]
Not clear at all	[ ]	[ ]

iii) **Accuracy:** Do you consider that these explanations are accurate (in considering the area of the pond and pond suitability to support crested newts)?

	C	D
Definitely	[ ]	[ ]
Mostly	[ ]	[ ]
For a reasonable part	[ ]	[ ]
Partially	[ ]	[ ]
Not at all	[ ]	[ ]
Don't know	[ ]	[ ]

iv) **Utility:** How would you rank the utility of the information given by each of the explanations for *educational* and *practical* users?

Educational Users			Practical Users		
	C	D		C	D
Definitely useful	[ ]	[ ]	Definitely useful	[ ]	[ ]
Mainly useful	[ ]	[ ]	Mainly useful	[ ]	[ ]
Reasonably useful	[ ]	[ ]	Reasonably useful	[ ]	[ ]
Only partly useful	[ ]	[ ]	Only partly useful	[ ]	[ ]
Not at all useful	[ ]	[ ]	Not at all useful	[ ]	[ ]

v) **Detail:** Do you think the explanations are of sufficient detail to enable *educational* and *practical* users to understand the complexities of the relationship between the area of the pond and pond suitability to support crested newts?

Educational Users			Practical Users		
	C	D		C	D
Definitely	[ ]	[ ]	Definitely	[ ]	[ ]
Mostly	[ ]	[ ]	Mostly	[ ]	[ ]
In the main	[ ]	[ ]	In the main	[ ]	[ ]
Partially	[ ]	[ ]	Partially	[ ]	[ ]
Not at all	[ ]	[ ]	Not at all	[ ]	[ ]

vi) **Preference:** Which explanation do you prefer?

C	[ ]
D	[ ]
Neither	[ ]



### Section 3

You are asked 'What percentage of the pond's surface area is occupied by emergent vegetation?'. You wish to know how the level of emergent vegetation in the pond relates to the pond's suitability to support crested newts, and request some explanation of why the above question is being asked. Using two different computer systems, you are presented with the following two sets of information:

---

#### **Explanation E**

if the percentage of pond surface area <95%  
then the aquatic vegetation within the pond is suitable

if the pond location is suitable  
and the aquatic vegetation within the pond is suitable  
and the pond's depth and size is suitable  
and the fish status of the pond is suitable  
and the duck status of the pond is suitable  
and pond shading is suitable  
and pond permanence/frequency of drying is suitable  
and pollution levels are suitable  
then the pond is not suitable to support crested newts

---

#### **Explanation F**

The pond's surface area being occupied by 50-75% emergent vegetation implies the pond habitat is unsuitable to support a crested newt population, in some cases. This is derived from:

The pond having a percentage surface area occupied by 50-75% emergent vegetation indicates the pond is at a stage of succession that is very late, in about half of the cases.

The pond being at a stage of succession that is very late indicates the pond is not suitable for aerobic, aquatic species, in some cases.

The pond being unsuitable for aerobic, aquatic species indicates the pond is not suitable for crested newt larvae.

A pond that is not suitable for crested newt larvae indicates the pond habitat is unsuitable to support a crested newt population.

---

i) **Comprehensibility:** From the details given in the explanations, do you understand how the level of emergent vegetation and the pond's suitability to support a viable crested newt colony are related (tick one box from each column)?

	E	F
Definitely	[ ]	[ ]
Mostly	[ ]	[ ]
For a reasonable part	[ ]	[ ]
Partially	[ ]	[ ]
Not at all	[ ]	[ ]

ii) **Clarity:** How clearly do you think each of these explanations expresses the relationship between the level of emergent vegetation and the pond's suitability to support a viable crested newt colony?

	E	F
Extremely clear	[ ]	[ ]
Very clear	[ ]	[ ]
Reasonably clear	[ ]	[ ]
Not very clear	[ ]	[ ]
Not clear at all	[ ]	[ ]

iii) **Accuracy:** Do you consider that these explanations are accurate (in considering the level of emergent vegetation and pond suitability to support crested newts)?

	E	F
Definitely	[ ]	[ ]
Mostly	[ ]	[ ]
For a reasonable part	[ ]	[ ]
Partially	[ ]	[ ]
Not at all	[ ]	[ ]
Don't know	[ ]	[ ]

iv) **Utility:** How would you rank the utility of the information given by each of the explanations for educational and practical users?

Educational Users			Practical Users		
	E	F		E	F
Definitely useful	[ ]	[ ]	Definitely useful	[ ]	[ ]
Mainly useful	[ ]	[ ]	Mainly useful	[ ]	[ ]
Reasonably useful	[ ]	[ ]	Reasonably useful	[ ]	[ ]
Only partly useful	[ ]	[ ]	Only partly useful	[ ]	[ ]
Not at all useful	[ ]	[ ]	Not at all useful	[ ]	[ ]

v) **Detail:** Do you think the explanations are of sufficient detail to enable educational and practical users to understand the complexities of the relationship between the level of emergent vegetation and pond suitability to support crested newts?

Educational Users			Practical Users		
	E	F		E	F
Definitely	[ ]	[ ]	Definitely	[ ]	[ ]
Mostly	[ ]	[ ]	Mostly	[ ]	[ ]
In the main	[ ]	[ ]	In the main	[ ]	[ ]
Partially	[ ]	[ ]	Partially	[ ]	[ ]
Not at all	[ ]	[ ]	Not at all	[ ]	[ ]

vi) **Preference:** Which explanation do you prefer?

E	[ ]
F	[ ]
Neither	[ ]