UNIVERSITY OF SOUTHAMPTON

Knowledge Acquisition for Knowledge-Based Systems: An Empirical Comparison of Two Methods

Clive Norman Washington Nicholson

Submitted for the degree of
Doctor of Philosophy

Department of Accounting & Management Science

September 1992

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF SOCIAL SCIENCE

Doctor of Philosophy

Knowledge Acquisition for Knowledge-Based Systems:

An Empirical Comparison of Two Methods

by Clive Norman Washington Nicholson

In the search for more efficient ways of developing accurate knowledge bases, many methods have been used to acquire knowledge. Some writers have conjectured about which methods are likely to be most successful for different problem-solving tasks. But few studies have tried to predict, and then test hypotheses about, the differential efficacies or efficiencies of the methods. This research focused on the repertory grid technique and knowledge acquisition from a minimal set of examples, compared their external features, then examined their probable effects on the mind of the knowledge source. This analysis allowed some hypotheses to be stated about the performance of the two methods.

To test these hypotheses, a single-factor within-subject experiment was designed, and the SCENIC knowledge-acquisition tool developed. Volunteers used the tool to elicit, by both methods, their own knowledge of a domain. The subjects also classified exemplars, which were then used to evaluate the knowledge bases generated. A multivariate analysis of variance on the data collected in the experiment supported some of the hypotheses. Those not supported highlight opportunities for improving the efficiency of the repertory grid technique; and some ideas are expressed as to how this improvement might be achieved.

But little do we perceive what solitude is
Or how far it extendeth.
For a crowd is not company;
And faces are but a gallery of pictures;
And talk a tinkling cymbal,
Where there is no love.

- Francis Bacon (*On Friendship*)

Contents

Chapter 1. The Knowledge Acquisition Problem 1

Introduction 1

Critical Task 4

Large Amounts of Knowledge 7

Slow Process 9

Making Knowledge Acquisition Manageable 12

A Deluge of Words 16

Conclusions 23

Chapter 2. Learning Without Case Records: a mapping of the repertory grid technique onto knowledge acquisition from examples 26

Introduction 26

Eliciting Examples 27

The Repertory Grid Technique 32

Deriving Heuristic Knowledge 40

Conclusions 54

Chapter 3. Some Implications of Cognitive Psychology for Knowledge Acquisition 57

Introduction 57

Divisions in Memory 59

Retrieval of Information 63

Implications for Knowledge Acquisition 70

Conclusions 83

Chapter 4. Evaluation Measures 85
Introduction 85
Certification for Operation 86
Evaluation for Research 87
Evaluation: What is Involved 88
Possible Influences 101
Conclusions 108

Chapter 5. The Controlled Experiment in Knowledge-Acquisition Research 109
Introduction 109
Experiments 110
Conclusions 119

Chapter 6. Design of an Experiment to Compare two Knowledge-Acquisition Techniques 123
Introduction 123
Hypotheses 124
Variables 125
Method 127
Subjects 129
Knowledge Domain 133
Some Requirements for the Apparatus 138
Data to be Collected 139
Conclusions 141

Chapter 7. Design of SCENIC: a CAKE Tool for Empirical Work 143
Introduction 143
Structure of the Tool 147
The RGT Function 148
The KAMSE Function 151
The Performance System 165
Tracing 166
Conclusions 167

Chapter 8. How Technique Affects Knowledge Acquisition: a Controlled Experiment 168
Introduction 168
Experiment Design 170
Results 173
Conclusions 187

Chapter 9. Discussion of the Results 191
Introduction 191
Implications of the Results 193
Internal Validity 199
External Validity 201

Chapter 10. Overall Conclusions 202
Lessons from the Research 202
Limitations of the Research 206
Value of the Research 209

Appendix A. Entry strategy selection: a broader view 213

Appendix B. Instructions for KAMSE 217

Appendix C. Instructions for the Repertory Grid Technique 221

Appendix D. Knowledge Units Elicited 226

Knowledge from the Repertory Grid Technique 226

Knowledge from KAMSE 227

References 230

Index 263

Figures

1. Four normative processes for building a knowledge-based system 5
2. Three perspectives on knowledge acquisition methods 13
3. A training set (set 1) for machine learning. 27
4. The partial domain model implicit in the examples 28
5. Boose's application tasks 38
6. Sets 1.1 and 1.2 48
7. Sets 1.1, 1.2.1, and 1.2.3 49
8. Two routes to a single objective 55
9. Memories, and types of information thought to be stored in them. . . 61
10. Modes of retrieval from long-term memory 64
11. An example of declarative knowledge 65
12. Tentative productions constructed from declarative knowledge 66
13. Example of a compiled production 67
14. Retrievability of long-term memory. 69
15. Stages in the repertory grid technique and KAMSE 72
16. An inference matrix for classification of a strategic business unit . . . 78
17. Possible relationship between knowledge quantity and KB performance 102
18. Possible relationship between knowledge quality and KB performance 103
19. A possible context for the two observations of Michalski & Chilausky 104
20. Factors and effects in building a knowledge base. 120
21. Variables involved in the hypotheses 125
22. The knowledge acquisition stages of the two methods implemented in SCENIC 144
23. Functions of the SCENIC CAKE tool 148
24. Subfunctions of the RGT function 149

25. Subfunctions of the KAMSE function 152
26. The panel used to elicit elements 154
27. One of the panels used to elicit constructs 155
28. The construct-elicitation panel with a separated triad 156
29. The panel that elicits the opposite pole of a construct 157
30. The rating panel 158
31. The panel used to elicit classes 159
32. The panel used to elicit attributes 160
33. The panel used to elicit examples 161
34. Structure of the performance system in SCENIC 166
35. Omnibus MANOVA tests of knowledge-acquisition method 175
36. Mean number of minutes to acquire elements and classes 176
37. Univariate analysis of variance in time used to acquire elements or classes 176
38. Mean number of minutes for each method to acquire constructs or attributes 177
39. Analysis of variance of time used to acquire constructs and attributes 178
40. Mean number of minutes to acquire ratings or examples under each method 179
41. Analysis of variance of time used to acquire ratings or examples . . . 180
42. Mean accuracy of knowledge bases generated under each method . . 181
43. Analysis of variance of knowledge-base accuracy 182
44. Mean number of minutes to classify evaluation exemplars 183
45. Analysis of variance of time used to classify evaluation exemplars . . 183
46. Number of minutes used at different stages of each knowledge-acquisition method 184
47. Mean number of minutes to classify evaluation exemplars 186

48. Univariate analysis of variance in time to classify evaluation exemplars 187
49. Summary of support for the hypotheses 188
50. Contribution of the research 210

Preface: what this thesis is about

When a knowledge-based system is delivered to its users and incorporated successfully into their routine operations, it is generally the product of a sequence of actions in which prospective users, knowledge engineers, and domain experts participated. Such a system would normally be accepted for routine use only when it has been shown to work with a high degree of accuracy. Some of this accuracy can sometimes be achieved by refining the knowledge base (Politakis, 1985; Ginsberg, 1988). But the size of the refinement task, and even the necessity for refinement, can be reduced if the knowledge acquisition process is itself capable of producing highly accurate knowledge bases.

Knowledge acquisition can be a difficult problem; and several methods have been used to try to solve it. Some of these methods are variations of other methods; even some apparently new methods are simply variations of old ones. Some of the methods have been implemented in knowledge-acquisition tools. Chapter 1, "The Knowledge Acquisition Problem" on page 1 discusses these issues to provide a background for the rest of the work.

The method used for knowledge acquisition can have profound effects on the pace and outcome of the process. Indeed, when a domain expert agrees to have his knowledge elicited by a knowledge engineer, it is crucial that this elicitation be as effective and efficient as possible. Such efficacy and efficiency are unlikely to be achieved unless the elicitation process closely matches the cognitive processes of the domain expert. Few pointers exist for someone faced with choosing among the tools or indeed among the methods. Boose (1989) has tried to position several of these tools in a multi-dimensional space. However, both the dimensions and the positions of individual tools on them partially reflect Boose's own biases.

This research, however, is not concerned with selecting among all possible techniques, but singles out two: the repertory grid technique and knowledge acquisition from a minimal set of examples. When the repertory grid technique (Kelly, 1955; Shaw, 1980) is used for knowledge acquisition, the process can be subdivided into six stages. Knowledge acquisition from a minimal set of examples can also be subdivided into six stages, which are parallel with those of the repertory grid technique. At the second of these stages, the repertory grid technique elicits constructs while knowledge acquisition from a minimal set of examples elicits attribute descriptors and values. Chapter 2, "Learning Without Case Records: a mapping of the repertory grid technique onto knowledge acquisition from examples" explores the external differences and similarities of the two methods, and tries to determine why using one might be more advantageous than using the other.

But it is difficult to explain why one knowledge-acquisition method might be more efficient or efficacious than another, without considering how the knowledge underlying a cognitive skill is represented in the mind and how it is retrieved. For example, in trying to retrieve an expert's knowledge, the repertory grid technique constrains the expert to generate constructs by considering three elements at a time. In knowledge acquisition from a minimal set of examples, the expert also follows the analogous process of generating attribute descriptors and values by considering the classes. This method, however, does not constrain the expert to consider any specific number of classes at a time. Instead, the expert is given the freedom to consider classes in any groupings that seem natural or expedient.

Some cognitive theorists assert that a domain expert cannot access the compiled productions that govern his or her performance. What can be retrieved is declarative knowledge, but that is generally sufficient to produce accurate knowledge bases, if combined with appropriate meta-knowledge. In

the same way as a novice can produce effective productions from declarative knowledge, meta-knowledge embodied in a knowledge-acquisition tool can operate on declarative knowledge to fashion productions that exhibit expert performance. Such considerations give rise to hypotheses that can be tested empirically. Chapter 3, "Some Implications of Cognitive Psychology for Knowledge Acquisition" on page 57 discusses these issues. Here is the essence of the questions raised by the hypotheses: Do experts find it easier to express classification knowledge by describing examples or by making distinctions between examples?

It is also difficult to compare the two methods and to articulate objective theories without measures of the various variables of interest. How have researchers defined and measured these variables? And how are they correlated? Chapter 4, "Evaluation Measures" on page 85 surveys the literature concerned with this.

Although researchers in this field are fond of case studies and benchmarks; the controlled experiment appeared ideal for hypothesis testing: it might be appropriate if sufficiently strong reasons could be found for using it. This is not the usual method of doing research on knowledge acquisition, but a few controlled experiments have been used, each having lessons for others to learn. Chapter 5, "The Controlled Experiment in Knowledge-Acquisition Research" on page 109 surveys the literature on using this approach for this purpose.

Against this background, it was necessary to design an experiment that would provide the data to test the stated hypotheses. Chapter 6, "Design of an Experiment to Compare two Knowledge-Acquisition Techniques" on page 123 discusses the design of the experiment. To perform the experiment, it was also necessary to design and build a vital piece of apparatus: a knowledge acquisition tool embodying the two methods mentioned above. Chapter 7,

"Design of SCENIC: a CAKE Tool for Empirical Work" on page 143 describes the design of the tool. The act of proceeding with this design and development helped to refine both the objectives of the experiment and the nature of the tool.

With all the preparations completed, the time came when it was appropriate to involve volunteer subjects in the experiment. The data collected was subjected to a multivariate analysis of variance to identify significant effects of independent variables. Chapter 8, "How Technique Affects Knowledge Acquisition: a Controlled Experiment" on page 168 is an analysis of the data collected.

The analysis shows that, whereas the two methods produce equally accurate knowledge, the domain expert needs to expend more effort when the repertory grid technique is used. This increased effort is at two stages of the process. At one of these stages, shortcomings of the repertory grid technique are evident, but are probably capable of being remedied so as to make the technique more efficient.

Chapter 9, "Discussion of the Results" on page 190 is an extended discussion of the results of the experiment. The thesis ends with Chapter 10, "Overall Conclusions" on page 201, which assesses the worth and significance of the research, its shortcomings, and the further work that it invites.

Acknowledgements

I am grateful to a number of people who contributed to this thesis by reviewing parts of it. These individuals all made useful suggestions and asked searching questions, which helped improve the quality of the final version.

Geoff Cooper, John Horswill, and Pete Stretton, reviewed Chapter 2, "Learning Without Case Records: a mapping of the repertory grid technique onto knowledge acquisition from examples" at various stages of its development. Valerie Harris reviewed Chapter 3, "Some Implications of Cognitive Psychology for Knowledge Acquisition." Clare Jackson and Melita Rustage reviewed an earlier draft of Chapter 5, "The Controlled Experiment in Knowledge-Acquisition Research." Dave Reynolds read an earlier draft of Chapter 10, "Overall Conclusions."

I am also deeply grateful to Sue High for her advice on the appropriate statistical treatment of the experimental data.

Several anonymous reviewers, who commented on papers that I submitted for publication in journals, provided welcome reassurance that I was not totally out of synchrony with the rest of the knowledge-acquisition community.

I am also indebted to Chris Woodford, who was always ready with stimulating comments at the merest mention of cognitive psychology, for not only reviewing some of the papers that I submitted to journals, but also for reading the thesis from cover to cover. In the latter endeavour, he provided comments of all kinds: proof-reading, editorial, aesthetic, and technical. Under some of these headings, Vanessa Parnaby also contributed.

Throughout the research, my supervisors at Southampton (Con Connell and Jonathan Klein) read everything that I put in front of them. They provided encouragement, thought-provoking discussion, and helpful suggestions to keep

me heading in productive directions. Without their help, I would surely have laboured much longer over this work.

To Phillip Powell I would like to express appreciation for a shot at his MSc students. My thanks also go to the experimental subjects themselves (volunteers every one) who gave freely of their time, some just for the experience, others to play their part in pushing back the frontiers of knowledge. The empirical work would have been impossible without them.

Ultimately, it was my employers, IBM United Kingdom Laboratories Ltd, who made it all possible by providing financial help under their Tuition Support Programme. My managers (Margaret O'Donnell when I started, and Jim Geraghty when I finished) helped by supporting my applications for sponsorship. They also displayed faith in my ideas by allowing me to apply them to real-world problems. Perhaps most crucially, the staff of the IBM library at Hursley were very helpful indeed in finding information whenever I needed it.

My wife Helene deserves special mention for her extraordinary understanding and patience, allowing me to attend to the research at unsocial hours and at times when better initiated spouses would have been expected to be mowing grass. She also read a number of the chapters, and provided valuable feedback. If parts of this thesis are intelligible, she helped make them so. On the other hand, if anything really unclear remains, the responsibility is entirely mine.

Chapter 1. The Knowledge Acquisition Problem

Abstract

This chapter develops the background for the rest of the work by looking at the problem of knowledge acquisition and some of the approaches taken to its solution. It argues that these approaches are aimed at making the process more manageable, finding methods powerful enough to cope with the challenges, and finding efficient ways of building systems. Interviews of various kinds, protocol analysis, and ways of coping with the copious textual material they typically generate, are discussed. Some other methods, e.g., the repertory grid technique and knowledge acquisition from a minimal set of examples elicit rather less text and more knowledge units that are directly usable in knowledge bases. In spite of the range of approaches available, there is little evidence to inform the choice among the methods and tools.

Introduction

Computer programs have been developed to model various kinds of decision-making activities. The simplest of these decisions involve considering a number of factors in order to make a choice from a set of possible consequents. Some kinds of diagnosis and classification tasks are like this. More complex tasks involve a series of interrelated simple decisions, the outcomes of which are used to assemble and tailor a plan or design (Garg-Janardan & Salvendy, 1988), or a skeletal report (Klinker, Bentolila, Genetet, Grimes, & McDermott, 1987).

The potential of these models has been evident ever since a system to infer the partial structure of substances from their mass spectra began to take shape in 1965. Since then, numerous other knowledge-based systems have been developed for a wide range of tasks. In recent years, the deployment of these systems (KBSs) has progressed from primarily analysis to synthesis problems. These systems have proved to be useful in a variety of applications from risk

classification (e.g., Shaw & Gentry, 1990) to process planning (e.g., Joseph & Davies, 1990).

Diagnostic systems are also increasingly being based on models of the physical system rather than merely models of the expert's decisions.

Increasingly KBSs are being developed for imbedding in traditional systems (see, e.g., Freundlich, 1990) rather than for stand-alone use. In such situations, many of the questions from the KBS are addressed not to a user, but to a database or to a program that seeks the judgement of the KBS. Often the result of the inference is also not displayed for a person, but rather passed to another program or used to initiate a process.

However, despite the growing range of their application, KBSs depend for their development on eliciting knowledge from a source — often a difficult task, which Feigenbaum (1977) described as the "critical bottleneck". The knowledge-acquisition bottleneck has become a cliche in the field of knowledge-based systems development (see, e.g., Mitchell, 1983; Wielinga, Bredeweg, & Breuker, 1988; McGraw, 1989; Rowley, 1990; Agarwal & Tanniru, 1990).

Three project-management factors, susceptible to managerial action, appear to contribute to the existence of the bottleneck. Firstly, knowledge acquisition often occupies a critical position in the sequence of tasks for building a knowledge-based system. As Hart (1986) has noted, "all the knowledge must be acquired before it can be represented". Secondly, large amounts of knowledge are usually required to build meaningful systems (see, e.g., Jacobson & Frieling, 1988). And thirdly, the process of acquiring knowledge is often slow (see, e.g., Smith, 1984; Shalin, Wisniewski, & Levi, 1988). These factors are discussed in the sections that follow.

Various approaches have been brought to a concerted assault on the problem. A growing number of tools that have been called knowledge-support

systems, knowledge-acquisition tools, knowledge-engineering workbenches, and computer-assisted knowledge-engineering (CAKE) tools, have been developed. Boose (1989) lists 65 of these tools; and several others have emerged since then. These tools share the aim of solving some of the problems evident in transferring knowledge from knowledge sources[1] to knowledge bases.

These tools attempt to provide computer support at various stages of the knowledge acquisition process, and seek to make building a knowledge-based system "a piece of cake". Most of these tools are designed to facilitate the acquisition of knowledge for what Kitto & Boose (1989) refer to as "analysis tasks", although a few of these tools, for example, SALT (Marcus, 1987; Stout, Caplain, Marcus, & McDermott, 1988; Marcus & McDermott, 1989) and CGEN (Birmingham, 1988), have tackled typically more complex "synthesis tasks".

A knowledge-acquisition tool generally models a particular style of interaction between a knowledge engineer and a domain expert. Behind the interaction, the tool organises and interrelates the information being obtained.

A systematic appraisal of the problem of transferring knowledge into a knowledge base is an essential foundation for understanding the task faced by knowledge-acquisition techniques and tools. This chapter therefore discusses these difficulties, and offers a perspective from which they may be viewed. Three reasons are suggested for the existence of the bottleneck, which is being attacked by several techniques and tools.

---

[1] Throughout this thesis, the term "knowledge source" is used to mean the person or artifact possessing the knowledge to be elicited. The term has been used in other senses in the literature (see, e.g., Clancey, 1983; Wielinga, Schreiber, & Breuker, 1992).

Critical Task

It has long been known (see, e.g., Koontz & O'Donnell, 1972) that certain activities are difficult to control, because they tend to reach 85 or 90% completion and stay there while time continues to elapse, and costs continue to be incurred. One solution has been to subdivide such activities into smaller, more manageable, units which can be monitored and perhaps even resequenced. These task sequences also enable explicit consideration of choices between reducing costs and shortening the schedule by selectively allocating additional effort (see, e.g., Buffa, 1972).

Several writers have proposed normative sequences of tasks to perform in building a knowledge-based system. Figure 1 on page 5 shows some contrasting prescriptions for the process. It is evident that the approach which Forsyth recommends is a casual one, perhaps suited to small systems in which the builder is archiving his or her own knowledge with the aim of becoming familiar with tools for developing knowledge-based systems. Weiss & Kulikowski, on the other hand, propose a more structured process, while Bowyer, Markowitz & Yusko recommend a process along the lines of some system-development methodologies. Even so, there are clear similarities between the different processes.

Where knowledge acquisition fits in the process depends to some extent on how the former is defined; and there is some disagreement as to what knowledge acquisition involves. While some writers (e.g., Buchanan, Barstow, Bechtal, Bennett, Clancey, Kulikowski, Mitchell & Waterman, 1983) see knowledge acquisition as the entire process of building a knowledge-based system from scratch, others (e.g., Hart, 1986) define it simply as elicitation. Waterman's (1985) definition of knowledge acquisition as "the process of

<table>
<tr><th>Forsyth (1984)</th><th>Weiss & Kulikowski (1984)</th><th>Bowyer, Markowitz & Yusko (1987)</th><th>Wielinga, Schreiber & Breuker (1992)</th></tr>
<tr><td rowspan="4">1 Purchase a shell

2 Prototype until you know what you want

3 Write production version in another language</td><td>1 Define problem (constraints, goals, roles, participants, resources)</td><td>1 Define system objectives
2 Define subsystems</td><td rowspan="2">1 Knowledge identification
a) Collect data
b) Identify tasks, concepts, and relations</td></tr>
<tr><td>2 Conceptualize
a) Interview an expert

b) Abstract characteristics</td><td>3 Create Cause Tables
4 Write knowledge-engineering document

5 Pareto analysis
6 Build control flow model
7 Verify knowledge-engineering document
8 Define skill parameters</td><td rowspan="3">2 Knowledge modelling
a) Collect data

b) Select interpretation model
c) Define domain schema
d) Build domain structures
e) Assemble model
f) Validate model
g) Differentiate model
h) Construct model bottom-up</td></tr>
<tr><td>3 Computer representation

4 Build prototype using a general purpose tool</td><td>9 Code knowledge base</td></tr>
<tr><td>5 Test, refine, specialize, & generalize knowledge base</td><td>10 Establish test cases

11 Test & validate</td></tr>
</table>

Figure 1. Four normative processes for building a knowledge-based system

extracting, structuring, and organizing knowledge from some source, usually human experts, so it can be used in a program" typifies this latter view.

From his perspective on medical diagnosis systems, Politakis (1985) argues that they are developed in a process the first stage of which is "specification of the diagnostic conclusions and the findings", but that real

knowledge acquisition "consists of formulating the rules that relate the findings to the conclusions".

The first three processes shown in Figure 1 on page 5 assume the acquisition of heuristic knowledge for analysis tasks. But some researchers, e.g., Breuker & Wielinga (1987) and Sykes (1987), argue that systems based on heuristic knowledge have the following shortcomings in their operation:

- The explanations that they give of their reasoning are often shallow.
- They tend to arrive abruptly at the limits of their knowledge.
- The knowledge they contain is difficult to transfer to other problem-solving tasks.

A growing number of researchers think that these difficulties can be resolved by modelling the physical system to which the knowledge relates. The KADS methodology assumes a model-based paradigm (see, e.g., Voss, 1990). Knowledge-acquisition tools, e.g., CAUSA (Dilger & Moller, 1990), have also been used to acquire knowledge for such systems.

Whether heuristic or model-based knowledge is used, it would clearly be useful to have more flexible methodologies, allowing development to be expedited by assigning increased resources, or allowing several tasks to proceed in parallel. The value of prototypes has long been recognised as a means of demonstrating design concepts and of visualising what an emerging product looks like. Weiss & Kulikowski (1984) recommend building a "prototype as soon as possible" because it provides something tangible around which knowledge acquisition can progress. Forsyth (1984, p 17) also argued for using an expert-system shell "as a prototyping tool till you know what you want".

Rapid prototyping seems to have the potential for enabling builders of knowledge-based systems to break free from the constraints imposed by a serial methodology such as that of Buchanan *et al* (1983) or Weiss & Kulikowski

(1984). Rapid prototyping effectively repositions knowledge acquisition in a knowledge-based system building project, because stages after elicitation (i.e., representation and testing) can proceed apace in a cycle of elicit-build-test. With the growing emphasis on integrating knowledge-based systems with traditional systems (see, e.g., Freundlich, 1990), this approach is especially important when the knowledge-based system being built is part of a larger system. An early prototype of the knowledge-based system can be used to test the rest of the system or even to refine its design.

Knowledge-acquisition tools can provide support for the rapid prototyping approach (see, e.g., Gutwald & Wallace, 1987; Shaw, 1988a and b; Boose, 1988; Gaines, 1988; Whipple, Davis, Kam, & Needham, 1989). Moreover, a knowledge-based system can be prototyped in pieces, which are later fitted together as a complex whole, thus allowing the possibility of expedited completion by concurrent development. According to (Boose & Bradshaw, 1987), the techniques embodied in AQUINAS "combine to make it a powerful testbed for rapidly prototyping portions of many kinds of complex knowledge bases". One insight to emerge from this is that tools like AQUINAS might not be very effective for tackling complex problems.

However, rapid prototyping is not without its detractors: Neale (1987, p 60) has cited criticisms by Breuker & Wielinga (1987 and 1983) that constantly referring the "emerging system to the expert for comments" is a waste of "the expert's valuable time".

Large Amounts of Knowledge

From as early as 1965, when DENDRAL was being developed, it had already started to become evident that large amounts of knowledge were required to build significant knowledge-based systems. According to Shirai & Tsujii (1985), DENDRAL has "a large number of rules for inferring the partial structure of a

substance" from the substance's mass spectra. Weizenbaum (1976) similarly describes MACSYMA as "an enormously large program for doing symbolic mathematical manipulations".

According to Partridge (1986), "intelligent behaviour within the ill-defined empirical world ... is founded upon vast amounts of information". Future systems are expected to be even more knowledge-hungry than past and present ones. As Wilkins (1987) has noted, "autonomous computer systems of the future will need far more knowledge than humans can explicitly transfer". However, it is difficult to determine when the amount of knowledge in a knowledge base is optimal for its intended purpose. Wilkins (1987) has asserted that

> extant techniques for reasoning under uncertainty for expert systems lead to a sociopathic knowledge base ... [that is,] there exists a subset of the knowledge base that gives better performance than the original knowledge base.

But if data is collected in a form that can be fed into an inductive or other learning process, a distillation can take place, thus reducing the sociopathy of the knowledge base. This distillation is discussed in Chapter 2, "Learning Without Case Records: a mapping of the repertory grid technique onto knowledge acquisition from examples" on page 26: a mapping of the repertory grid technique onto knowledge acquisition from examples": a mapping of the repertory grid technique onto knowledge acquisition from examples": a mapping of the repertory grid technique onto knowledge acquisition from examples" on page 26. Indeed, some knowledge-acquisition methods seek to minimise the input to the learning process. This is appropriate where the data is not already available.

Another approach to the problem of large amounts of knowledge is to avoid it if possible, by selecting appropriate domains that produce significant

payback from small amounts of knowledge. The validity of this approach is demonstrated by several small knowledge-based systems, built quickly, that have provided their users with considerable payback (Nicholson, 1988; Department of Trade and Industry, 1992a and 1992b). These small systems are often well structured problems which are the best candidates for knowledge-based systems, according to Partridge (1986), who asserts that

> if we take most of the vagaries and ill-structure of everyday life out of the picture we are left with realistic and serious potentially tractable ... grist of the expert system mill.

Because large knowledge bases can be difficult to update, Hart (1986, p 26) advises against building systems containing knowledge that is likely to change often. According to her, systems containing volatile knowledge "will need updating if [they are] to retain [their] expertise". Naylor (1983) argues that if the task to be modelled "can be reduced to a series of judgements ... you have a good chance of building an expert system to do it".

Recognizing that the problem of knowledge acquisition becomes even more daunting as the scope of a potential domain expands, Partridge (1986) argues that "the success of expert systems rests largely on the very restricted and specialised nature of the domains in which they operate".

Slow Process

Knowledge acquisition is widely recognised to be an inherently slow process. Buchanan et al (1983) assert that "manual methods for acquiring strategic knowledge push the limits of human cognitive abilities".

Some writers argue that the process is slow because it is not well understood. For example, Smith (1984) asserts that

while tools and methodologies have emerged to provide considerable aid in this activity, the process of eliciting, representing, and refining the knowledge utilized by the domain expert remains ill-defined and time consuming.

Whether the remarks discussed above are valid or not, one reason that the process is slow is (see, e.g., Johnson-Laird, 1983; Anderson, 1982) that, in general, human experts are poor knowledge sources because they find it difficult to explain how they make their decisions.

The difficulty in articulating the thought processes behind expert performance is often attributed to tacit knowledge. Tacit, or implicit, knowledge contrasts with explicit knowledge. Examples of explicit and tacit knowledge given by Nickerson (1977) are:

Explicit	**Tacit**
o 2 + 4 = 6.	o The Mississippi flows downhill.
o Whales are mammals.	o Julius Caesar had a mother.

Tacit knowledge is not stored for the individual items to which it relates, but can be inferred or computed. Retrieval of tacit knowledge takes more time than that of explicit knowledge (Camp, Lachman, & Lachman, 1980), but, as discussed in Chapter 3, both are subject to conscious control.

Anderson (1982) has offered a theory explaining why experts have difficulty communicating their expert knowledge. Anderson's ACT* (adaptive control of thought) theory about memory and expertise asserts that there are two forms in which people store knowledge: declarative and procedural. Both explicit and implicit knowledge of facts is declarative. A person with an understanding of a domain, but with no experience of applying the knowledge to tasks, is armed with only declarative knowledge. As the knowledge is

exercised repetitively in performing some task, a process of compilation takes place, creating a second version of the knowledge in a procedural form, which the expert cannot retrieve consciously. Findings of empirical studies (e.g., Lundell, 1988), are consistent with ACT* theory.

Some knowledge acquisition methods take knowledge compilation into account by seeking to acquire the knowledge underlying expert performance without asking directly for it. While experts may have difficulty explaining how they perform their expert task, they are still good at actually performing the task in new (or even hypothetical) situations. Protocol analysis, task observations, and forward scenario simulation attempt to take advantage of these abilities. Kolodner (1983) argues that experts are also good at recounting previously handled cases. Both the repertory grid technique and knowledge acquisition from examples draw on these abilities.

But according to Anjewierden (1987, p 29), "AI has very little to say about methods or techniques that could be used to alleviate [the knowledge-acquisition bottleneck].

Some writers (e.g., Neale, 1987) see this lack of effective techniques as caused by the fact that people who build knowledge-based systems do not spend enough time reflecting on their own methods, documenting these methods, and generally trying to develop insights into which methods work well under which circumstances. Neale has also indicted knowledge engineers for being too ready to accept explicit knowledge while ignoring tacit knowledge, which is difficult to access without the use of well developed techniques from psychology. On the other hand Anderson (1982) argues that the expert does not have conscious access to the knowledge that produces the skilled behaviour; and attempts by knowledge engineers to access the inaccessible may also explain the slowness of the process.

Other writers (e.g, Forsythe & Buchanan, 1989) argue that the process is slow because of the techniques, the tools, and even the people involved in knowledge acquisition. But knowledge-acquisition technology is still evolving; and some researchers feel that the labour-intensive process of knowledge engineering is unnatural and goes against the trends in computing (see, e.g., Shaw & Gaines, 1987b). They argue that the process can probably be made more efficient by creating tools capable of organising and analysing the information being obtained more efficiently than people can.

Even so, Forsythe and Buchanan (1989, p 435) have criticised the CAKE approach as

> glossing over detailed questions of how to gather the material [in favour of a] focus on higher-level issues such as classifying information collected from the expert or insuring that this information is complete.

Making Knowledge Acquisition Manageable

Reitman Olson & Rueter (1987) have described seven knowledge-acquisition methods as direct and five as indirect (see Figure 2 on page 13). In general, the direct methods are approaches for obtaining information from a knowledge source whereas the indirect methods are techniques for analysing and organising the data obtained into forms that reduce the "representational mismatch" between knowledge source and knowledge base.

Interviewing

First among the direct methods listed by Reitman Olson are interviews. Essentially, the different kinds of interview are all conversations in which a knowledge engineer asks questions and a domain expert tries to answer them. Neale (1987) has listed fourteen types of interviews used for knowledge acquisition. As Figure 2 on page 13 also indicates, LaFrance (1987) has

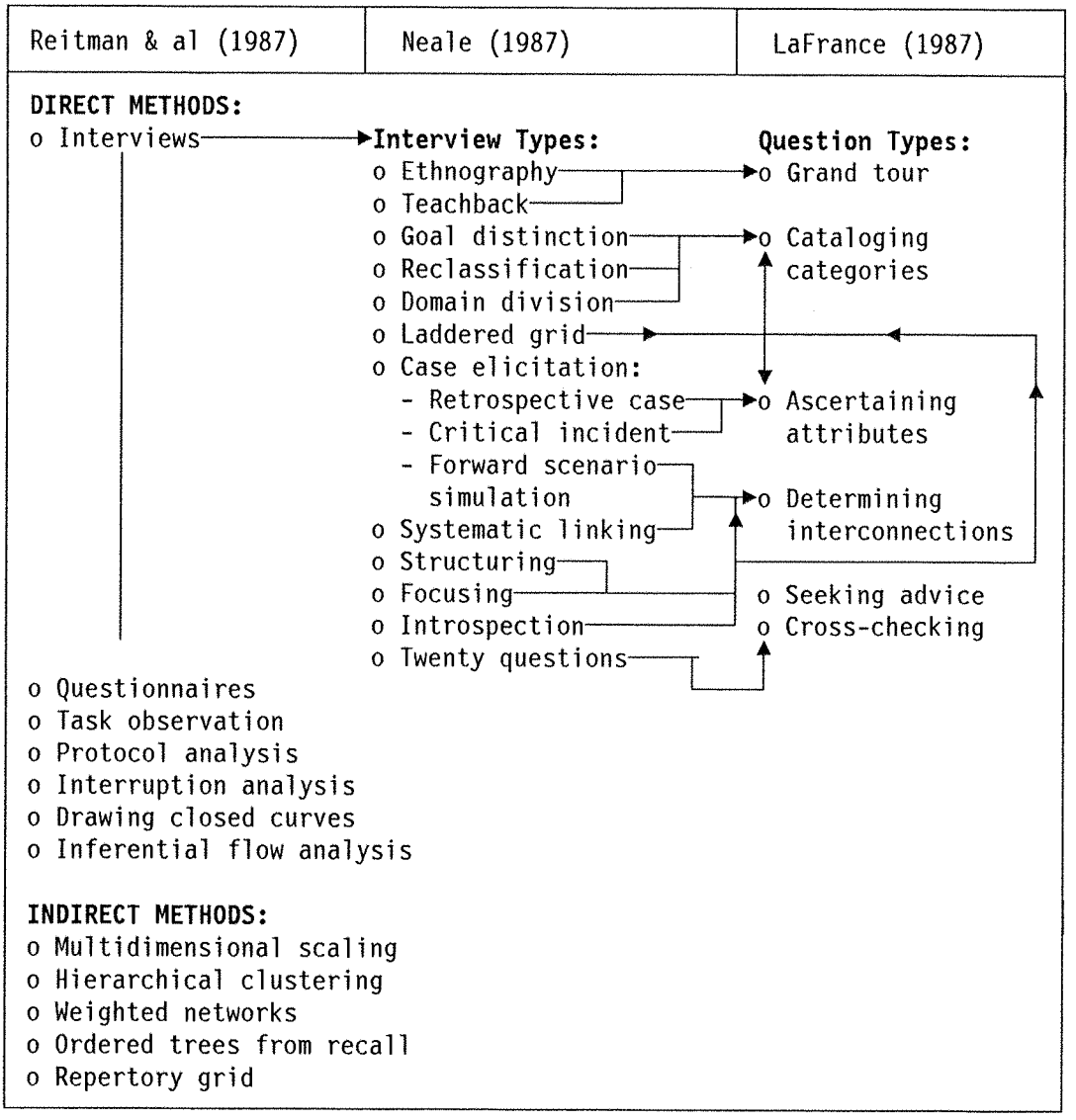

Figure 2. Three perspectives on knowledge acquisition methods

identified six types of questions that a knowledge engineer might ask, depending on the "form of knowledge" being sought.

Although it might be expected that different types of questions would be asked in a single interview, some of the interview types listed by Neale (1987) can be shown to involve certain aims primarily and can therefore be linked to certain question types. These links are also shown in Figure 2. It is also

interesting to note that three of Neale's interview types are aimed at eliciting examples. These three interview types can all be used in knowledge acquisition from a minimal set of examples (KAMSE, see page 29), one of the two methods with which this research is chiefly concerned.

Three of LaFrance's question types are involved in several knowledge-acquisition methods, including the repertory grid technique (see page 32), the other method with which this research is mainly concerned. It should also be noted that although Reitman Olson classifies the repertory grid technique as an indirect method, it is a style of interviewing coupled with a method of analysis.

Evidently, the developers of CAKE tools can choose from a wide range of techniques for elicitation and analysis of domain knowledge. Typically, these tools simply model the interaction used in a particular interview method combined with one or more techniques for analysing and organising the knowledge elicited.

So far, none of these tools is universally applicable to all knowledge domains for which their use might be contemplated. As Clancey (1991) puts it, there is no "big switch" tool, which is one reason that comparative evaluations like Boose (1989) are useful. One of Boose's primary objectives seems to be to draw conclusions about which tools are suited to which application tasks.

Of all the dimensions on which Boose compares the tools, the one to which he attaches foremost significance is the application task for which a tool is suited. This is a pragmatic choice, since a tool user would be primarily interested in finding a tool that can handle the problem for which its use is being contemplated.

A prospective tool user will also want to know how easy a particular tool is to learn and use, whether it runs on the equipment available, and whether it delivers knowledge bases to an appropriate performance system.

Boose used seventeen dimensions to compare the 26 tools, and he presents scatter tables involving a few of these dimensions (domain dependence, application task, degree of automation, training needed, and life cycle support).

Even so, Boose's primary concern seems to be classifying knowledge-acquisition tools according to the application tasks for which they are suited; and very few tools have been developed that are suitable for applications that do both analysis and synthesis. For example, no knowledge-acquisition tools for expert instruction systems have been reported (see, e.g, Tompsett, 1989); but it is clear that such tools are more difficult to build, because they have to handle a dimension of complexity not present in the modelling of simple decisions. They may also have been slow to appear because they have to embody two problem-solving methods: heuristic classification and heuristic construction (Clancey, 1986).

One important test of any knowledge acquisition tool must be how well it enables its user to capture, record, and exploit domain knowledge. Whether one tool is more suitable than another for a given purpose has to be determined empirically. But AQUINAS is primarily a knowledge acquisition tool; its purpose is to elicit knowledge and create a knowledge base. If Boose's distinctions between the tools are accurate, the knowledge base generated might be useful, during domain definition, in advising on a suitable tool for the problem.

Builders of knowledge-acquisition tools are therefore beginning to include components that assess the suitability of tool for domain. For instance, Kitto & Boose (1989) have modified the "dialog manager" in AQUINAS: by adding "the automated guidance facility [which] provides advice on strategies and tools based on application characteristics" (p 149). Stout *et al* (1988) have also changed SALT so that it now recognizes when a domain is not quite suited to itself. A further step might be to generate a suitable tool, which can then be

used to elicit the knowledge required. PROTEGE (Musen, 1989) uses this approach. Yet another approach is taken by the ACKnowledge project (see, e.g., Shadbolt, 1990), which tries to integrate a number of different tools within the framework of a knowledge-based workbench.

It appears that, to be really effective, such front ends and generators of knowledge-acquisition tools need to be based on the kind of information that Boose (1989) presents. It is therefore quite useful to try and place tools within some comparative framework that highlights their differences and similarities and gives some indication of which approaches or tools are most likely to be successful when applied to a given knowledge-acquisition situation.

A Deluge of Words

While using the methods discussed above may make the knowledge engineer more confident, and the domain expert more talkative, some of these methods, particularly protocol analysis and various types of interviewing, can generate a surfeit of words. One research focus therefore centres on obtaining knowledge from text. This section briefly reviews these methods.

Protocol Analysis

Verbal protocol analysis, or thinking aloud, (see, e.g., Ericsson & Simon, 1984; Hart, 1986; Neale, 1988) is a method that produces at least a transcript of the expert's protocol (what s/he claims to be paying attention to while performing the expert task). This protocol can be developed at the same time as the task is performed, or later when the videotape recordings are being viewed. KRITON (Diederich, Ruhmann & May, 1987), in addition to using other knowledge-acquisition methods, analyses verbal protocol transcripts, as discussed later in "Knowledge Acquisition from Text" on page 18.

Natural-language processing does not have to be automatic, it can be done by careful analysis. But this analysis is likely to be more productive if the

relationships being found can be represented in the text stored in the computer. Hypertext has been used as a way of helping knowledge engineers analyse textual information. According to Storrs (1989),

> hypertext, in its simplest form, is a set of nodes connected together by undifferentiated links. Each node is an unstructured piece of text or graphics (or both) and each link is a uni-directional association between two nodes.

These text nodes are viewed on a computer screen, and the reader can request any of the other nodes linked to the one being viewed. The reader thus makes his or her own non-linear path through the text. Use of hypertext to analyse a verbal protocol is not an automatic process. The knowledge engineer studies the protocol transcript, chunks it, and loads it into a hypertext system. S/he then creates links between concepts thought to be related, and makes annotations as s/he does the analysis. One of the KPT tools (Anjewierden, 1987), the Protocol Editor (PED), is a hypertext editor that enables a knowledge engineer to analyse the transcripts of verbal protocol analysis sessions.

Gaines & Linster (1990) used "the hypermedia tool HyperCard a general purpose knowledge acquisition tool for unstructured material in the form of text and diagrams". According to them, "the annotation and explanation captured in the hypermedia system were available as context sensitive help to the user of the expert system".

Some writers (e.g., Storrs, 1989) argue that the product of protocol analysis in hypertext ought to be usable, not just as help text, but as a knowledge base in its own right. According to Storrs, hypertext and semantic networks resemble each other in important respects, with a richly interconnected hypertext document being equivalent to a semantic network. It

may therefore be possible to use hypertext in place of a semantic network, which presumably could be used in a knowledge-based system if it were surrounded by appropriate meta-knowledge.

Knowledge Acquisition from Text

Some knowledge-acquisition methods generate large amounts of textual information. Even knowledge engineers who intend to use an interview method are advised to read existing texts on the domain before seeing the experts. Both kinds of text generally contain knowledge units which can be identified only by careful time-consuming analysis. The need for this analysis highlights a mismatch (identified by Buchanan *et al*, 1983) between the form in which an expert uses knowledge and the form in which it is required for building a knowledge base. One facet of this mismatch is discussed in Chapter 3. Another facet is that the expert may express domain knowledge in natural language while the knowledge-based system requires it as rules, frames, or some other formalism. This mismatch is seen by Gruber (1988) as "a fundamental problem in knowledge acquisition".

Natural-language understanding, which has long promised to enable people to communicate with computers without using a formal language, is also seen as a way to find meaning in large bodies of text. Both these goals, if they can be achieved, would relieve much of the drudgery of acquiring knowledge from interview transcripts, thinking-aloud protocols, and other textual sources.

Roskar (1988) argues that "the difficulty of transferring expert knowledge to the computer ultimately depends ... on the way in which the knowledge is represented within the computer". Buchanan et al (1983) see the "representation mismatch" as being overcome by two approaches: "learning by being told", and natural-language conversation with the expert combined with an "English-like" representation. The second approach was used by Tanyi &

Linkens (1989), who included in their expert-system shell "a module, KAM, which allows rules to be constructed in a pseudo natural language, FKRL". However, this approach still requires either the expert to express knowledge in this pseudo-English, or the knowledge engineer to translate the expert's natural language into the pseudo-English.

Even pseudo-English rules may have to be translated into a more concise formalism (in the system described by Tanyi & Linkens, the rules expressed in FKRL are translated into PROLOG). This translation involves the same mechanisms used in natural-language processing (e.g., Sager's, 1981, "Linguistic String Project"), i.e., the following components:

- Rules of grammar for the language
- A parser to do syntactic analysis on sentences
- A lexicon containing words in the language
- "Procedures for transforming string parse trees" into a semantic representation.

However, much of the work in this area seeks to determine the semantic content of individual sentences, rather than paragraphs or larger chunks of text. Various devices have been used to understand these larger chunks of specific kinds of text (e.g., Schank, 1975, used scripts to guide story understanding).

Schank argued that humans reduce sentences heard or read into a semantic representation, which has two ways of enriching what is being said. One device involves making explicit the implicit concepts embodied in the words and sentences. The other involves inferring what has not actually been said, but what may have been hinted at in previous sentences or what would normally be expected. The usefulness of Schank's theory is in its potential for converting one representation to another via the intermediate semantic representation or semantic dependency graphs. Clearly, the target

representation can be either another natural language or a knowledge-representation formalism.

But even tools without strong semantic capabilities can generally acquire some knowledge from text. Gettig (1989) describes a tool (KAM) that analyses textual data and extracts rules and facts, some of which may be suitable for direct use in knowledge bases. The text is scanned for certain words (e.g., if and when), which act as cues for the presence of condition/action pairs. Text in the vicinity of each cue is analysed more closely to try and isolate a rule. The analysis and extraction are guided by heuristics which the user can specify. This (and the fact that the system only partially generates some rules) means that the system is not completely automatic.

Moreover, there is no guarantee that any rules will be extracted, those extracted may not be relevant, and some of them may contain either ambiguous pronouns, or nouns incorrectly inferred from pronouns. These problems, according to Gettig (1989), are being addressed.

According to Rau, Jacobs, & Zernik (1989), storing text in a semantic representation (or "conceptual format") is unusual, but makes it easier to access text via natural language. They also assert that lexicons are not rich enough, i.e., "lack of extensive linguistic coverage is the major barrier to extracting useful information from large bodies of text". They therefore propose that natural-language processing systems should be tolerant to unknown words, and should "acquire lexical information automatically from the texts". Their SCISOR prototype information retrieval system, implements both recommendations.

Channier & Fournier (1988) have pointed out problems with automatic processing of technical texts:

- Scarcity of "general tools to recognize the constituents of the texts"

- Representing their contents to enable users to "modify and check them for consistency".

The ACTES project of Channier & Fournier was aimed at extracting "rules for an expert system simulating ... processors responsible for managing ... alarms". The project also aimed at finding a formalism to define grammars, and an intermediate semantic representation of information gathered from processing texts.

Reimer (1990) used a system called WIT, which does semantic analysis on technical texts, and "builds up representations of the concepts described in the text". Introductory material and textbooks are not used, because, Reimer argues, they are often out of date, and parsing mechanisms cannot yet cope with them. Reimer's system acquires knowledge of the terminology used in the domain, and proposes a hierarchy of concepts. This hierarchy is then restructured by "inductive generalization". This approach does, however, requires "small domain specific" knowledge before it can do any processing. If this approach (which is not based on any deep understanding of the text) works, it may speed text analysis in the early stages (domain orientation) of knowledge acquisition.

Schmidt & Schmalhofer (1990) also point out that the rules found in textbooks may be inaccurate, incomplete, conflicting, or open to interpretation. They argue that these problems can be solved by "enriching" the knowledge acquired from text, by eliciting "records of solved cases" from a domain expert.

Cognitive Mapping

The methods discussed in the preceding sections elicit text-rich information. Some other methods elicit information rich in knowledge units rather than text; the data they elicit are readily transformed into knowledge bases. One of these methods is cognitive mapping, a by-product of the ideas of Kelly (1955). The

technique is used in operations research by consultants (also called facilitators) to help clients (also called problem owners) define their problem space (see, e.g., Klein & Cooper, 1982; Eden, Jones & Sims, 1984; Morecroft, 1988; Eden, 1988).

The problem space is represented by "word-and-picture maps, algebraic 'sentences', models and simulations" (Morecroft, 1988, p 316-317). These representations are used to indicate "relationships that are perceived to exist among attributes and/or concepts" in the problem space (Zhang, Chen & Bezdek, 1989, p 31). Eden, Jones & Sims (1984) developed a tool called COPE to store, manipulate, analyse, and display cognitive maps. COPE's analysis of the concepts and the relationships between them highlights clusters, hierarchies, and loops. Facilitators focus on loops in particular, because loops indicate "vicious circles" in the problem owner's thinking, which must be resolved if progress with the problem can be made.

Morecroft (1988) argues that the fields of knowledge acquisition and cognitive mapping have enough in common to be able to influence each other's progress. In particular, knowledge acquisition may be able to contribute to cognitive mapping a "better [understanding of] how to elicit and reconstruct policymakers' broad business knowledge".

Improvements in the technology used for cognitive mapping may also make it feasible to use the same maps and models, either as an operational or an intermediate representation for knowledge bases. Zhang *et al* (1989) have developed a tool called Pool2, which gathers knowledge of a problem space from multiple experts. If developed further, Pool2 could, Zhang *et al* argue, generate knowledge bases.

The repertory grid technique and knowledge acquisition from a minimal set of examples are even more focused on knowledge units. These two methods are discussed in Chapter 2.

Escaping the Strait-Jacket

But some researchers feel that the techniques embodied in many CAKE tools are too structured and therefore restrict the natural flow of ideas that can occur during knowledge acquisition. One product of this view is "sloppy modelling", which allows the user to take the initiative sometimes, and to give it back to the tool, when appropriate (Wrobel, 1988, p 461).

> This means that the user is not required to develop a complete and well-structured model beforehand in order to then transfer it into the machine. Instead, the modeling activity itself becomes part of the system-supported knowledge-acquisition process.

According to Wrobel (1988, p 461), sloppy modelling "differs from those approaches [e.g. that of AQUINAS] in its emphasis on a cooperative mixed-initiative modeling process".

The sloppy modeling paradigm has objectives identical to those of other knowledge-acquisition approaches. Indeed, a well designed (or really effective) CAKE tool ought to coax the knowledge out of the expert while interacting with him or her.

Conclusions

Building a knowledge-based system is constrained by the development cycle, the large amount of knowledge that must typically be elicited, and the slow speed inherent in the acquisition process itself. These three factors can probably be addressed by fashioning methodologies that reposition acquisition relative to other tasks, by trying to minimise the amount of knowledge needed to construct useful systems, and by streamlining the acquisition process itself.

This may involve subdividing the task into manageable units, using an iterative development cycle, and applying resources in parallel. In addition,

problems have to be selected which can provide significant payback from small amounts of knowledge. These systems should be developed with tools that can recognise when enough knowledge has been obtained, thus minimising the "sociopathy" problem. Where the methods used generate plenty of textual material, it is useful to find efficient ways of processing the text. Another approach is to use other methods that generate less text, and more knowledge knowledge units that are directly usable in knowledge bases. Two such methods are examined in Chapter 2, "Learning Without Case Records: a mapping of the repertory grid technique onto knowledge acquisition from examples."

Tools for knowledge acquisition facilitate these approaches by modelling techniques for eliciting, analysing, and organising knowledge. These techniques include different styles of interviews, knowledge acquisition from text, and learning from examples. Few of these tools have actually reached the market, so a person needing a knowledge-acquisition tool today may still have to build one. What the existing tools do is to demonstrate a range of approaches to the problem of knowledge acquisition in various domains. A more useful objective than choosing among tools might therefore be simply choosing among approaches, methods, and techniques.

With so many approaches to choose from, there appears to be a need for information to enable intelligent choice among them for any given problem domain. Because different techniques may be suited to different circumstances, builders of knowledge-acquisition tools have begun to include in them front ends that assess the suitability of the tool for the intended domain.

However, when there is a choice between two methods for a given domain, little evidence exists to favour the choice of one approach over another. It is evident that there is a need for the development of a sound theoretical basis for matching methods to situations. Such a theory would have to stand up to empirical testing; and the empirical data can help shape and

refine the theory. There is little doubt that such a theory will develop over time, as individual pieces of research fill in parts of the overall jigsaw.

One of the current limitations of most of these approaches is that they can be used only to build systems for analysis tasks, while there are other types of tasks for which the elicitation problem is no less significant. The knowledge required for design and tutoring systems, for example, tends to be more complex. But complex knowledge can often be subdivided into a set of related simple decisions.

Chapter 2. Learning Without Case Records: a mapping of the repertory grid technique onto knowledge acquisition from examples

Abstract

In contrast with the text-intensive methods discussed in Chapter 1, building a knowledge-based system can sometimes be expedited by applying some machine learning process to a set of historical cases. In some problem domains, however, such cases may not be available. In addition, the classes, attribute descriptors, and attribute values that comprise the partial domain model in terms of which cases are expressed may also not be available explicitly. In these circumstances, the repertory grid technique offers a single process for both building a partial domain model and generating a set of training examples. Alternatively, a minimal set of examples can be elicited directly. This chapter[2] explores the relationship between knowledge acquisition from a minimal set of examples and the repertory grid technique, and discusses their shared need for machine learning. Fragments of business-strategy knowledge are used to illustrate the discussion.

Introduction

The repertory grid technique and KAMSE both elicit information that is rich in knowledge units rather than text. Knowledge units elicited by either method can be used to build knowledge bases. The knowledge units elicited by one method can also be mapped to those elicited by the other method. Moreover, these knowledge units can be organised as input to different kinds of learning

---

[2] A paper based on this chapter has been published as Nicholson (1992a).

processes. These processes are discussed, with special emphasis on machine induction, the usual learning method used with the repertory grid technique.

This chapter explores KAMSE and the repertory grid technique, showing the similarities between the two approaches, the equivalence of the information elicited, and the shared need to distill this information into a representation to support inference.

Eliciting Examples

Consider the five examples shown in Figure 3 (the names of the organisations are disguised) of cases in which a consultant in business strategy advised his clients on the best way to implement their product or market development strategy. These cases are expressed in terms of three attributes: cost of entry (sometimes called startup cost), payback period (or the importance of early return), and risk of failure. Three classes are present in the training set: internal development, joint venture, and acquisition.

Set	Co.	Entry Cost	Payback Period	Risk	Strategy Implementation
1	Ace	low	short	low	Internal development
	BU	low	short	high	Joint venture
	Cha	high	long	high	Joint venture
	Day	high	short	low	Internal development
	EZ	low	long	high	Acquisition

Figure 3. A training set (set 1) for machine learning.

The set of examples is therefore expressed in terms of a partial domain model, as shown in Figure 4 on page 28. This model consists of familiar entities (Clancey, 1986), but also relations which are usually assumed rather than stated explicitly. Attributes describe an example, which belongs to a class.

The examples help to establish a mapping between patterns of attribute values on the one hand, and classes on the other.

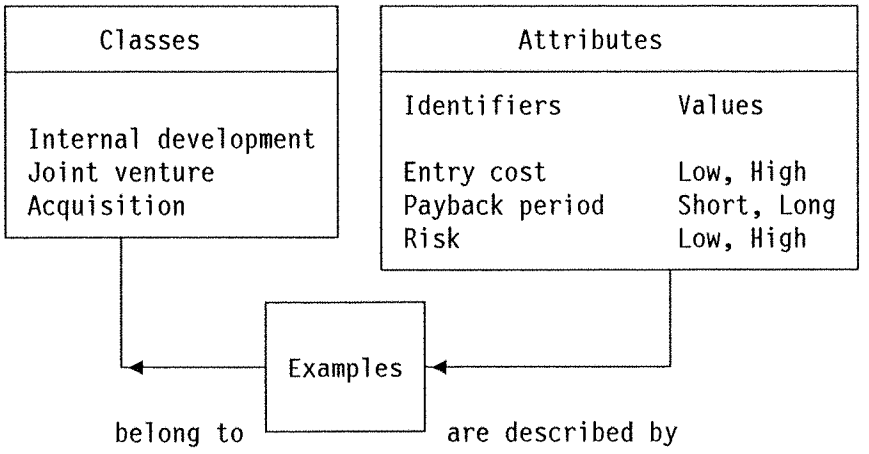

Figure 4. The partial domain model implicit in the examples

It may be argued that the induction algorithms do not require the partial domain model to be explicit; they require only that all cases be expressed in terms of the same model. Where historical records of cases are available with this information, it is simply a matter of distilling them into a knowledge representation, as discussed in "Deriving Heuristic Knowledge" on page 40.

In situations where such records are not available, one possible approach is to set up a mechanism for recording historical case data and then wait for enough data to be collected. Roskar (1988) set up a database to collect data about medical cases. According to him, this approach is "especially suitable for a domain in which the relative power of different pieces of diagnostic information has not yet been identified". Although this approach has the disadvantage of taking a long time, it has the advantage of capturing information about the expected frequency of each class.

However, if time is of the essence, or collecting data in this way is inconvenient or infeasible, retrospective case description, critical incident, and

forward scenario simulation (all concisely described in Neale, 1988) are interviewing techniques that can be used to elicit data in the form of examples. Lundell (1988) lists two of the kinds of examples that can be elicited: prototypes and exemplars. Exemplars, which can be elicited by forward scenario simulation, are random examples. A large number of examples is not necessarily required. For each class, a typical example (a prototype) and a few atypical ones (exemplars) can provide a high degree of coverage. This view underlies knowledge acquisition from a minimal set of examples (KAMSE), which is described further on page 72. In the domain being discussed, an exemplar might be elicited by the question:

> Suppose a company was facing a high cost of entry into its intended development area, and payback was expected to take a long time, but the risk of failure was low, what would you advise?

Prototypes, which can be elicited by the retrospective case description technique, are examples representative of a class — Lundell refers to them as "the central tendency" of cases seen in the past. A prototype might be elicited by the question:

> Give an example of a situation in which you would advise internal development.

But these methods all presuppose the existence of a partial domain model. Gruber (1988, p 583) recognises the necessity for a partial domain model, and asserts that

> for any application of machine learning to knowledge acquisition, ... somebody has to set up the learning problem for the induction algorithm: designing a representation for examples and generalizations,

defining all of the terms in those languages, encoding a set of training examples in the representation, and providing background knowledge ... that guides the induction algorithm to choose the right generalizations from the potentially infinite set of possibilities. This can require a significant knowledge engineering effort.

Morik (1987, p 93) also distinguishes between learning from cases and the domain theory in terms of which the cases are expressed. She sees "the construction of a domain theory as the first phase of the knowledge-acquisition process". Where this domain theory (or partial domain model) has not been made explicit, it must be developed before any meaningful gathering of examples can take place.

Of course, the attributes, their values, and the classes can be elicited as by-products of the recall and description of cases, as the following example illustrates. A knowledge engineer (KE) talks with an experienced business strategy consultant (Expert) about the same decisions discussed above.

KE: "Can you remember an actual specific case you worked on involving this decision? How did you decide which method of implementation to recommend?"

Expert: "Yes. At Ace the cost of entering the proposed industry was going to be low, and the payback period was quite short. So we advised them to pursue their product development strategy by internal development."

The expert has thus recounted a case in terms of two attributes and a class. At this early stage, the model consists of a single class and two attributes with two possible values each. The attribute "payback period", for example, can so far have a value of either "short" or "some other value". This information may be viewed as the beginnings of a domain model. But the question arises as to what other values each of the two attributes can assume,

because it is quite likely that the two attributes will not be sufficient to describe all cases of interest.

The knowledge engineer tries to expand the model by eliciting an exemplar.

KE: "Can you recall another case in which the cost of entry was low, and the payback period was short, but you made a different recommendation?"

Expert: "In BU's case, the cost of entry was also low and the payback period was reasonably short, but we recommended a joint venture because the risk of failure was quite high."

This case introduces a third attribute and a second class to add to the partial domain model.

The process discussed above is an unstructured one in which lists of classes, attributes, attribute values, and cases are being built up. In a more structured approach, the expert would be asked to list all the possible classes s/he could think of. When this list is complete, the expert would be asked to state the attributes or factors considered in making the decision and the possible values of each attribute. With this partial domain model in place, the expert can then be asked to recount cases that exemplify each class.

Any knowledge acquisition tool that uses this approach can probably enhance its efficacy (see page 82) by checking for conflicts (two cases with identical attribute values but different class) and for missing cases (classes for which there are no cases).

Some people find it necessary to produce a case for every possible combination of attribute values. Except where there are many classes, this kind of case-based approach is tedious. According to McClanahan & Luce (1988, p 112),

> If there are many values for each factor [attribute], the system can represent very fine distinctions. On the other hand, the number of examples increases dramatically with the number of alternative values per factor, as with the number of factors .... Even four factors, with four alternatives each, will require 4x4x4x4 or 256 examples.

In practice, these numbers are likely to be reducible if don't-care attribute values are catered for.

While cases *per se* can be used as the basis for inference (see, e.g., Bain, 1986) in a knowledge-based system, they are usually distilled into a representation (rules, a decision tree, or an artificial neural network) capable of supporting inferences about new cases. Machine induction, which is a way of distilling examples into rules, is discussed in "Deriving Heuristic Knowledge" on page 40. When a sufficiently comprehensive set of cases has been recorded, it can be processed by an induction algorithm or by any other process of learning from examples (e.g., a neural network or a genetic algorithm). The resulting knowledge can then be deployed in a performance system for validation, refinement, and consultation.

Another method of acquiring the same knowledge is the repertory grid technique described in the next section.

The Repertory Grid Technique

Although the repertory grid technique does not appear to be widely used by knowledge engineers working without CAKE tools (see, e.g. Nicholson, 1988), it has become a popular technique in the tools (see, e.g., Boose, 1985; Boose & Bradshaw, 1987; Shaw & Gaines, 1987; Garg-Janardan & Salvendy, 1988). Seven of the 26 tools compared in Boose (1989) use it.

The repertory grid technique, which was developed by Kelly (1955; see also Shaw & Thomas, 1978; Bradshaw & Boose, 1990) can be used for eliciting

the constructs that an expert uses in making decisions. Kelly (1955) used the technique to access the inner worlds of his patients. These inner worlds are modelled by multidimensional spaces in which each orthogonal axis represents a personal construct of the patient. Concepts, things, and people important to the patient are all located in that space, which is made explicit by the interviewing strategy and the grid used to analyse and structure the information being uncovered.

This same set of techniques has been employed in knowledge acquisition for knowledge-based systems (see, e.g., Hart, 1986). In this context, Hart describes the grid as

> a representation of the expert's view of a particular problem. A grid is composed of ... elements [and] constructs ... bipolar characteristics which each element has to some degree.

One early program to assist in repertory-grid analysis was Mildred Shaw's (1979) PEGASUS. She did not, however, use the elicited knowledge to build knowledge-based systems. Instead she used the grids to study agreement and understanding among individuals on a subject about which they shared knowledge. Later she did become interested in using grid techniques to acquire knowledge for knowledge-based systems (see, e.g., Shaw, 1981). As Boose (1988) has explained, "distinctions captured in grids can be converted to other representations such as production rules, fuzzy sets, or networks of frames".

Another tool embodying the repertory grid technique is AQUINAS (Boose & Bradshaw, 1987), which attempts to enable one or more domain experts to bypass the knowledge engineer, and use the tool to elicit their own knowledge. Using the repertory grid that it elicits, AQUINAS can generate knowledge bases in different formats suitable for use in some expert-system shells (e.g., S.1, KEE, EMYCIN).

An element E_i can be identical to a class from knowledge acquisition from examples. Alternatively, it can be equivalent to an example, allowing the possibility of multiple elements per class.

A construct C_j can be equivalent to an attribute identifier in knowledge acquisition from examples. Where constructs are grouped in named clusters, the constructs are equivalent to attribute values.

A single construct is equivalent to a single attribute using the visible pole as the descriptor, and yes and no as the values. For instance:

```
Construct                  Attribute
Green / Not green  ===>    Green
                           - yes
                           - no
```

A related set of constructs can often be equivalent to a single multivalued attribute. For example:

```
Constructs                 Attribute
Green / Not green  ===>    Colour
Amber / Not amber          - green
Red / Not red              - amber
                           - red
```

A user of the repertory grid technique starts by listing all the elements as in knowledge acquisition from examples. The user then employs a style of questioning, which Kelly called "the repertory test", and which is often referred to nowadays as triadic elicitation. The repertory test involves repeatedly going through the following steps (which can end when all elements are distinguished, by their ratings, from all other elements):

- Select three elements (perhaps, but not necessarily, at random).
- Ask which two are similar and different from the third.
- Ask for the construct C_j that makes two elements similar while different from the third.
- Ask for the opposite of C_j that characterises the dissimilar element.

- Ask for all other elements to be rated as fitting C_j, its opposite, or neither.

While following this procedure, a knowledge engineer gradually builds up a list of constructs and a set of ratings within a grid. But this grid can also be viewed as a list of prototypes, one for each element.

When Kelly used the technique he allowed only for entirely bipolar constructs. Every element E_i was rated against every construct C_j to assign in effect a value of 0 or 1 to the rating R_{ij}. More recently, people have used rating scales with an odd number of points (e.g., 1 to 5, 1 to 3, or even -2 to +2) with the central value indicating neither pole or either pole (or don't care). The rating scale is sometimes interpreted as a probability distribution. For instance, on the 5-point rating scale that AQUINAS uses, Boose (1989) appears to interpret ratings of 2 and 4 respectively as 40/60 and 60/40 distributions.

Under the repertory grid technique, the business-strategy expert might be asked: "Which of the following three situations is different from the other two — internal development is advised, joint venture is advised, and acquisition is advised?"

The expert selects internal development, and is then asked:

> What makes indications for joint venture similar to those for acquisition and different from those for internal development?

The expert says: "low risk of failure". This information is used to create an attribute with two possible values: "yes" and "no". Also created are three cases expressed in terms of the only attribute elicited so far. But all classes are not yet distinguishable from all others. Indeed, two of these cases conflict; so the same questioning is repeated with the same triad (since there are only three elements in this grid).

Herein lies one weakness of the repertory grid technique. Where the number of elements is small enough, triads cannot be found. Where there are only three elements, some tools will simply not embark on the repertory test (Garg-Janardan & Salvendy, 1988). Others use dyads to try and augment the list of elements while eliciting constructs (Boose, 1985).

One reason for the small number of elements in the domain discussed above is that classes are being used as elements. When this is done, only knowledge about prototypes is elicited. It is therefore likely that only three examples will be elicited, instead of the five shown in the previous section. A safer approach is to use examples rather than classes as elements. Later the elements can be clustered into class groupings. Applied in the domain being discussed, this approach increases the number of elements from three to at least five — and perhaps more, if the expert can think of other cases. Indeed, using cases as elements helps handle awkward disjunctive relationships, because each variation from the typical can be introduced as a separate element.

The repertory test asks for similarities and differences only to elicit distinctions between the elements in the grid. Once enough of the problem space has been captured to be able to distinguish every element from every other, it is unnecessary to elicit any further constructs. Indeed, there is no need to present any triad in which all three elements are already distinguished from each other.

For this reason, the repertory test can be optimised if constant attention is paid to these conditions. They will indicate which triads need not be presented and when to end the repertory test. But restricting the composition of triads could limit the knowledge acquired. This is because further distinctions between already distinguished elements could lead to ratings that distinguish between other elements not in the triad.

The repertory grid technique, which has been the basis for several knowledge-acquisition tools — e.g., AQUINAS, ETS, FMS-Aid, KITTEN, KRITON, KSSO (Gaines, 1987), and PLANET (Gaines & Shaw, 1986) — does not naturally accommodate disjunctive relationships between constructs. Yet such relationships occur in some domains. For example, in one of the word domains discussed on page 134, a compound verb may be formed by an AfV (affix + verb) or a PV (preposition + verb) combination (Selkirk, 1983), but not both at once. The kind of disjunctive relationship among constructs can be handled by having multiple elements of the same class and by catering for clusters of elements in the grid. So, in the example just given, we might be able to say that xV is a compound verb, and further that x = Af is one type of compound verb while x = P is another.

It is clear that the list of elements is crucial; and the questions asked to elicit these should be designed carefully to obtain both elements that are typical of a class, and those that are atypical. In addition to asking, as AQUINAS might, "list all the strategy implementations you can think of, one to a line", further questions are necessary. AQUINAS' question elicits the response "joint venture, internal development, and acquisition" — classes as elements. It would be useful to pose a further set of questions for each class so as to elicit the name of a typical case, and the names of other cases that differ from the typical one and from each other.

A repertory grid tool may also have facilities for displaying or printing scatter tables, and presenting various relationships that exist in a grid. Such facilities help to communicate the contents of a grid, and their implications. Several of these facilities are demonstrated in Boose (1989), which uses them, rather than statistical tables and graphs, to present the results of a comparative evaluation of 26 knowledge-acquisition tools. Boose's comparison conveys much more information about AQUINAS itself than any of the other tools

being compared, because AQUINAS is used as the vehicle for presenting the data.

So many scatter tables are typically possible, that there is a danger of presenting too much information, unless they are used selectively. AQUINAS appears to have overwhelmed Boose with data, and he seems to feel compelled to give his readers a glimpse of it without fully explaining or discussing what he presents. These data include the grid itself, scatter tables of one construct against another (a plane in the multi-dimensional space), and an implication graph. These different presentations do not in themselves identify inconsistencies in a grid, but may provide the information for the meticulous analyst to find inconsistencies. Indeed, his early presentation (p 5) of KNACK, OPAL, and SALT as using heuristic construction methods and being suited to synthesis applications seems inconsistent with his later presentation of both KNACK and OPAL under diagnosis tasks in a scatter table.

There are limitations in AQUINAS' ability to present hierarchies of constructs in scatter tables. For instance, the application tasks that Boose uses may be viewed as the hierarchy shown in Figure 5. However, the AQUINAS scatter table involving application tasks shows a flat scale with eight points.

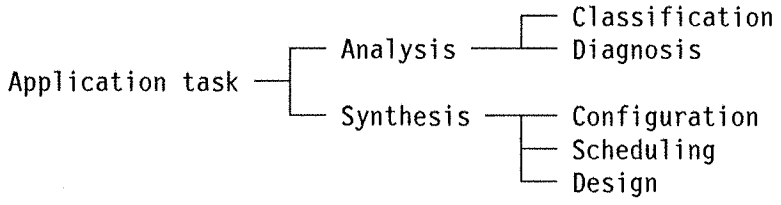

Figure 5. Boose's application tasks

AQUINAS thus gives no indication of the relationship between diagnosis and analysis, or between design and synthesis. However, these shortcomings can probably be overcome by users highly skilled in interpreting the tool's output.

But some writers question the worth of the repertory grid technique. According to Anjewierden (1987, p 31), "the weakness of AQUINAS is the use of [a] particular knowledge elicitation technique ('rating grids'), which is thought to be insufficient for general KBS development".

Analysis of the grid

When the analysis of the grid is left to be done after the repertory test (rather than during it), a batch job is in effect created for which the user must wait. In addition, when two constructs are found to be parallel, it is difficult to decide which one to delete at that late stage. It would be much more straightforward to look for a parallel construct immediately after rating. If a parallel construct is found, the similarity can simply be pointed out, and the user allowed to select one of the following actions:

- Introduce a new (or existing) element that would be rated differently on the two constructs.
- Rerate on one of the constructs.
- Leave the two constructs as they are.
- Discard one of the two constructs; one will be discarded later by machine induction anyway.

Ongoing grid analysis to determine when to end the repertory test does not always work well. One problem is that random triads may have a low probability of containing the two elements that are still indistinguishable (what Gammack, 1987, calls a "confused pair"). As discussed above, the triads can be restricted so that they contain these interesting elements. Or the tool could gauge when no further progress is being made by the repertory test. At that point, the tool could enter dyadic elicitation and show the confused pairs.

It is also during analysis of the grid that clusters of elements can be proposed and confirmed. Constructs can be treated in a similar way.

Boose (1988) has explained that "distinctions captured in grids can be converted to other representations such as production rules, fuzzy sets, or networks of frames". The next section discusses this conversion, which is in many ways identical to the transformation from cases to a knowledge representation.

The completed repertory grid is used to generate cases, which are then fed through a learning process (usually machine induction).

Deriving Heuristic Knowledge

Whether knowledge acquisition from examples or the repertory grid technique has been used to build up the partial domain model and gather a set of examples, these examples need to be reduced to a decision model which can be used for inference on new cases. This reduction is done by passing the examples through a machine-learning process that detects regularities in the data.

In general, machine learning starts out with a set of examples (the training set) and seeks to distill them into knowledge that can be used to diagnose cases outside the training set.

Machine learning, according to Mitchell, Carbonell & Michalski (1986),

> involves the study and development of computational models of learning processes. A major goal of research in this field is to build computers capable of improving their performance with practice and of acquiring knowledge on their own.

Research in machine learning has led to different approaches, including incremental learning, explanation-based learning, inductive or similarity-based

learning, neural networks, and genetic algorithms. These approaches are seen as essential by some writers (e.g., Wilkins, 1987), who have argued that the human-mediated methods described in Chapter 1 are not capable of generating the kinds of large knowledge bases likely to be required in the future. According to Wilkins, "this requires that computers learn independently".

Incremental Learning

Crawford (1989) argues that when machine induction (see page 46) is used, new examples are accommodated by the "brute force method" of adding them to the existing training set, and rerunning the induction algorithm. Matwin & Oppacher (1988) also argue that "relearning from scratch could be cumbersome" where a large set of training examples is involved. This viewpoint concludes that, while learning systems should "be able to learn from scratch", they should also be able to learn incrementally.

The solution provided by Crawford (1989) is to extend the CART induction algorithm to include "incremental learning., in which newly acquired examples can be used to update an existing tree". The training set still has to be retained, because a new example may cause a part of the existing tree to be "pruned away and the cases ... repartitioned".

Matwin's solution involves a tool called LEW, which acquires knowledge by parsing "cues" (question-solution pairs), and relating them to previously stored generalisations. Negative cues are used to specialise the knowledge, while positive ones are used to generalise. LEW, which uses concept clustering to "combine aspects of learning from examples and learning by analogy" (Matwin, Oppacher & Constant, 1989; Constant, Matwin, & Oppacher, 1988), can also be used for planning problems. The tool breaks such problems (e.g., the towers of Hanoi) into a set of simple problems each of which can be concentrated on, in isolation from the remainder.

Berwick's (1985) system for the acquisition (from example sentences) of knowledge about how to parse (syntactic knowledge) used an incremental learning approach. Incremental learning is also used in the knowledge-acquisition component of VIE LANG (a German-language dialogue system). According to Buchberger, Zsolnai, & Trost (1989), the system "incrementally augments its knowledge based on a sound basic repertoire".

Explanation-Based Learning

Explanation-based learning (EBL, see Mitchell, Keller, & Kedar-Cabelli, 1986) is a method of refining an existing knowledge base, rather than a way of building one from scratch. An EBL system starts with a knowledge base, also referred to as a complete domain theory. A single example is then presented to the system, which uses a theorem prover to build a deductive proof tree for the example. This proof tree shows why the "example is a member of a concept class" (Whitehall, 1990). The EBL system then generalises the proof tree (e.g., by changing constants into variables).

Genetic Algorithms

Another approach to creating knowledge bases from examples is by using a set of learning programs called genetic algorithms. Genetic algorithms, which use processes analogous to those in Darwinian evolution, were developed by Holland (1975, see also Smith, 1980).

The user first describes the form of the examples to be fed into the process. This description specifies the target expression (or class field) and where it is found in the examples. The attributes, and their position in the examples, are also described. A set of examples is then fed into the process, which uses them to generate (randomly, according to Forsyth, 1984b) a tentative set of rules. These rules are all expressed as strings of equal length (see, e.g., Yoneda, Minagawa, & Kakazu, 1992).

The next step is the essence of genetic algorithms: the accuracy of each rule is determined by applying it to the examples. Inaccurate rules are discarded, and accurate ones survive but are changed by genetic operators. These genetic operators change rules in three ways: crossover, mutation, and inversion. Crossover operates on two rules (character strings) at a time to produce "descendants [from] a kind of mating"; mutation is "a random transcription error", and inversion is "an internal crossover with reordering" (Forsyth, 1984b). In one iteration, or generation, all the rules are evaluated and either discarded or modified. The process is repeated for a number of generations specified by the user, or until convergence is achieved.

Typically, genetic algorithms require a large number of generations, and hence computer time, to reach convergence.

Artificial Neural Networks

Rather than generating rules or decision trees, neural networks simulate the physiology of arrays of nerve cells and thereby learning from examples. Forsyth (1984b) has asserted that the early researchers in the field had the dream of

> building a richly interconnected system of simulated neurons [which] could start off knowing nothing, be subjected to a training program ... and end up doing whatever its inventor wanted.

Rosenblatt's perceptron is only the best known of these early models of nerve cells. Rosenblatt (1958, pp 387-405) describes the perceptron as:

> a hypothetical nervous system, or machine, ... designed to illustrate some of the fundamental properties of intelligent systems in general, without becoming too deeply enmeshed in the special, and frequently unknown,

conditions which hold for particular biological organisms .... [These systems] can learn to associate specific responses to specific stimuli.

According to Bischel & Seitz (1989), "most applications of neural networks are classification problems". When classifying, a neural network takes as its input a number of binary-valued features of the item to be classified. The binary pattern is propagated through the network, and arrives transformed at the output ports, which indicate the class of the item. The transformation that the input undergoes as it passes through the network is brought about by the attenuation characteristics developed during training.

Rosenblatt's contemporaries (Farley & Clark, 1954) also built a model that could store knowledge in the links between its nodes, and thus learn to distinguish patterns of input. In the model of Farley and Clark, a neuron is represented as a simple organism with certain properties discussed briefly below.

In the network, any neuron can be connected to any other neuron, but in general is not connected to every other neuron. Farley & Clark define a connectivity ratio, K, as the probability that any given node will be connected to any other. With $K = 0.4$, for example, any node may in the extremes be connected either to all others or to no others, but on the average will be connected at random to only 0.4 of the other nodes.

Normally, the weights connection start out with random values; but Rada (1984) used a scheme based on the perceptron for refining knowledge. The weights do not start at random values; instead they are specified (or estimated) by domain experts). Refinement occurs as cases, whether for training or validation, are fed through. During learning the neural net is given feedback as to the correctness of its classifications. Where it finds that it has classified an object incorrectly, it makes adjustments to its internal characteristics so as to improve its performance.

According to Forsyth (1984b), Minsky & Papert (1969) proved that perceptrons "could be taught to recognise patterns, but only a limited class of patterns". After Minsky & Papert (1969) discredited the neural network approach as having serious limitations, the ideas were abandoned for several years until recently, when advances in computer technology have produced processors capable of implementing larger and more complex neural networks. But Minsky & Papert (1988) have reaffirmed their reservations about the entire connectionist approach.

There are now several neural-network models, including single-layer and multi-layer perceptrons (see, e.g., Lippmann, 1987). Several neural network software packages are on the market and applications are being developed for production use. Applications such as speech recognition, robot vision, diagnosis of lower back pain, mortgage underwriting decisions, stock price predictions, bond rating, signature verification, handwriting recognition for input to computers, and the grading of plywood have been reported.

But there are still problems with neural networks. According to Bischel & Seitz (1989), "an unfortunate property of most neural networks is the large number of training patterns necessary for the teaching". This does not necessarily require that a large training set be available. A small training set can be used repeatedly until the network stops learning anything new. But, as Bischel & Seitz (1989) point out, "this implies that the training phase is very computation-intensive".

However, this may not be very important. Buchanan *et al* (1983, p 158) have pointed out that intensive use of computation resources is often an acceptable price to pay for reducing the amount of work that knowledge engineers have to do.

Machine Induction

Unlike EBL, similarity-based learning does not require the existence of a knowledge base, rather it generates one by finding regularities in a set of examples. Machine induction, which originated with the concept-learning system of Hunt, Marin, & Stone (1966), is a form of similarity-based learning which involves analysing a set of examples to discover relationships between classes and patterns of attribute values. Like the direct process of interviewing, machine induction is also widely used is used in CAKE tools. Sometimes it is used to distill the knowledge obtained from an interview method into a compact, efficient, representation to support inference (see Chapter 2).

Michalski & Chilausky (1980) used machine induction to generate a knowledge base for diagnosing soybean diseases. According to Buchanan *et al* (1983), the knowledge base's diagnosis did not coincide with that of the expert in 100% of cases. However, the knowledge base was much more accurate than one made up of rules elicited directly from the expert.

But, as these and other writers (e.g., Gruber, 1988) have demonstrated, before relationships can be induced, the classes and the attributes (often called a partial domain model) have to be elicited. These attributes and classes must, in general, first be elicited from the domain expert or from other sources. As Buchanan *et al* (1983) put it, "finding meaningful, causal associations in a large data base requires considerable basic knowledge of the domain". Gruber (1988, p 583) argues that creating a partial domain model "can require a significant knowledge engineering effort when the task is more complicated than simple classification".

A few tools, e.g., KRITON (Diederich, Ruhmann & May, 1987), therefore include both the repertory grid technique for eliciting the required data, and an induction algorithm to distill the data into a compact, efficient, set of rules.

In addition to the carefully selected examples from which the rules are induced, a separate set of examples is generally required to validate the induced rules. Michalski and Chilausky (1980) used some three hundred examples for induction and a similar number of other examples to validate their induced knowledge base. Thus, gathering the examples, and expressing them in terms of attribute values and classes, can be an onerous prerequisite to induction.

Some writers (e.g., McClanahan & Luce, 1988) argue that every possible combination of attribute values should be present in the training set. Typically, this involves using a large number of examples. Others (e.g., Quinlan, 1986) argue that the induction process is more efficient if a small number of examples is used. Even where numerous examples are available, a window of a few (fifty or less) carefully chosen ones is actually used for induction. The remaining examples are used for testing the coverage of the induced knowledge.

Politakis (1985) has referred to inductive methods as "black boxes" which create rules that domain experts and humans in general sometimes have difficulty making sense of. This is especially true if continuous-valued attributes are involved, because induction typically subdivides them into ranges (Fayyad & Irani, 1992), which are often quite meaningless to domain experts. But Politakis points out that in some domains, such as medical diagnosis, domain experts will not have confidence in a system unless they understand and agree with the rules that it uses.

Generalising about the applicability of method to problem, Politakis (1985) argues that approaches such as machine induction are better suited to small problems, and not good enough for large or complex ones, which need "better methods ... to reduce the dimensionality". Quinlan (1991) has also pointed out that, while the attribute/value approach may be adequate for simple systems, many situations exist, e.g., engineering design, that involve hierarchies

of objects. A new approach embodied in a tool called FOIL generates PROLOG statements by detecting regularities in the structured examples.

The ultimate objective of machine induction is to break the training set into several smaller sets, each containing cases of a single class. The distilled knowledge is simply a trace of the sequence of decisions that brought about the transformation from a single multi-class set to several single-class sets.

Consider again the set of cases presented in Figure 3 on page 27. A visual inspection of the data reveals that there are three different ways of subdividing this set. One way is to subdivide it into two sets based arbitrarily on the risk of failure. (This division is not entirely arbitrary, because inspection reveals that low risk selects a single-class set, i.e., internal development.) As Figure 6 shows, Ace and Day fall into set 1.1 while BU, Cha and EZ fall into set 1.2.

Set	Co.	Entry Cost	Payback Period	Risk	Strategy Implementation
1.1	Ace	low	short	LOW	Internal development
	Day	high	short	LOW	Internal development
1.2	BU	low	short	HIGH	Joint venture
	Cha	high	long	HIGH	Joint venture
	EZ	low	long	HIGH	Acquisition

Figure 6. Sets 1.1 and 1.2

Because set 1.1 contains just one class, it needs no further subdivision and can be left alone. But set 1.2 contains more than one class, so further subdivision is necessary. Visual inspection of the three cases in set 1.2 reveals that this set can be subdivided based on either entry cost or payback period. Using payback period results in the three sets shown in Figure 7 on page 49.

Set	Co.	Entry Cost	Payback Period	Risk	Strategy Implementation
1.1	Ace Day	low high	short short	LOW LOW	Internal development Internal development
1.2.1	BU	low	SHORT	HIGH	Joint venture
1.2.2	Cha EZ	high low	LONG LONG	HIGH HIGH	Joint venture Acquisition

Figure 7. Sets 1.1, 1.2.1, and 1.2.3

It is evident that there are now two single-class sets and one two-class set (1.2.2). Obviously, set 1.2.2 is further subdivided by a test on entry cost.

The distilled knowledge can be expressed as a decision tree. But this can be converted (Quinlan, 1987) into a set of rules, as follows:

a) IF risk of failure is low
THEN internal development is recommended.

b) IF risk of failure is high
AND payback period is short
THEN joint venture is recommended.

c) IF risk of failure is high
AND payback period is long
AND cost of entry is high
THEN joint venture is recommended.

d) IF risk of failure is high
AND payback period is long
AND cost of entry is low
THEN acquisition is recommended.

Of course, this bit of manual induction has been done in a somewhat arbitrary manner. The rules arrived at are not the only ones possible. Indeed, this consultant's view of the problem space is not the only one possible (Appendix A, "Entry strategy selection: a broader view" on page 212 provides

a somewhat fuller discussion of the problem). In addition, the simplicity of the problem allows an intuitive approach to be successful. But when the size of the training set, the number of attributes, and the number of classes is increased, more systematic methods are needed. Such methods are embodied in algorithms for machine induction (Quinlan, 1986; Michalski & Larson, 1983; Niblett, 1987; Clark & Niblett, 1987 & 1989; Cestnik, Kononenko & Bratko, 1987).

These algorithms all require as input a set of possible classes, a set of attributes and their possible values, and a set of training cases.

The preceding discussion of manual induction illustrates the process that machine induction follows. One difference is that machine induction does not arbitrarily select attributes for splitting the training set. One approach is to select the test that minimises entropy (Quinlan, 1986). Other approaches are also used (see, e.g., Mingers, 1989; Buntine & Niblett, 1992).

It will be recalled that although all the cases were expressed in terms of the same attributes, not all the attributes were required in determining some classes. For example, the class "internal development" was determined by a single attribute, viz., risk of failure. On the other hand the other two classes could be distinguished from each other only by considering the other attributes as well. So, the inductive process generalises from specific cases and eliminates redundant or unnecessary tests. Clearly, some process of this kind is necessary in producing an efficient knowledge base and eliminating what Wilkins (1987) called "sociopathic" knowledge, discussed earlier on page 8.

The algorithms developed for finding regularities in examples include Quinlan's (1986) "iterative dichotomiser" (ID), which has gone through several refinements. ID3 is the best known of these refinements. Michalski has been associated with a series of algorithms called AQ (see, e.g., Michalski & Larsen, 1983). AQ11, one refinement in this series, was used by Michalski & Chilausky

(1980) to demonstrate the power of induction to distill effective and accurate knowledge from a set of training cases.

The different algorithms have their weaknesses and strengths, (see, e.g., Roskar, 1988; Hart, 1986) and attempts continue to find better methods (see, e.g., Fayyad & Irani, 1992). Clark & Niblett (1987) have tried to combine the best features of the AQ series and the ID one and produce an algorithm that has advantages over the two on which it is based.

Don't-Cares

If elements are allowed to be given don't-care (or not-applicable) ratings in the repertory test, that creates the problem that the induction algorithm will have to cater correctly for the don't-care attribute value. Similarly, if prototypes are used in knowledge acquisition from examples, it is quite likely that some cases will have don't-care attribute values. One approach is to expand each don't-care by generating several cases (one for each possible value of the attribute). This approach, however, can run into a combinatorial explosion.

Another approach avoids this explosion by generating a random value in place of each don't-care, but this can result in an erroneous knowledge base. Quinlan (1986) originally argued for replacing each don't-care with the most common (and, presumably, most likely) value of the attribute in question. Even that is not guaranteed to be entirely satisfactory.

A more effective approach lies in the way the grid is analysed. The distance between a don't-care and any other rating should be taken as zero. This encourages the knowledge source to distinguish adequately between what could be identical ratings. Even so, some don't-cares may still reach the induction process. Here, a satisfactory way to process don't-cares is to treat them as being equivalent, during learning, to every possible value (see e.g., Gams & Lavrac, 1987; Quinlan, 1989).

Inductive Tools

Learning from examples has been vaunted by many writers (e.g., Michalski & Chilausky, 1980; Quinlan, 1986; Michie, 1991) as a key solution to the problem of knowledge acquisition. Typical of these, Norris (1986) argues that

> the usual approach based on informal interviews ... is time-consuming, error-prone and often results in a knowledge base that is significantly incomplete.

Michalski & Chilausky (1980, p 79), argue that

> current computer induction techniques ... offer a viable knowledge acquisition method if the problem is sufficiently simple and well defined.

But not everyone is convinced: according to Roskar (1988, p 81),

> several limitations of the existing software for machine induction from examples have to be solved before this method can become efficient enough.

Roskar argues that logic-based induction is more suitable for representing "general and specific knowledge, and is thus more suitable" for medical diagnostic domains than similarity-based learning.

Tools discussed in learned journals — e.g, RL4 (Clearwater & Provost, 1990) and ITRULE (Goodman & Latin, 1991) — tend to be induction engines without the user interfaces that would make them worthy of being called knowledge-acquisition tools. But several commercially available tools (e.g,, 1st-Class, and AutoIntelligence) for expert systems development incorporate induction algorithms in knowledge acquisition from examples. These tools allow users to input both a partial domain model and examples, which the tool

then uses to induce rules. Such tools are often targeted at domain experts for their direct use without the mediation of knowledge engineers.

Learning's Promise

Machine-learning systems tend to be involved only during the construction of a knowledge-based system. The learning tends to be abandoned as soon as the system is judged good enough to be put into operation. Some writers (e.g., Bain, 1986; Kolodner, 1983) think that this is unfortunate. According to Bain:

> people tend to improve their abilities to reason about situations by amassing experiences in reasoning. The more situations which a person knows about, the more able he is to account for feature differences between new input and old knowledge .... The inability to save accounts of previous experience for future application and modification represents a serious shortcoming of most, if not all, rule-based systems.

Two aims of learning from experience involve refining

- Knowledge in the knowledge base
- Probabilities of different consequents.

The first aim requires a *post-hoc* analysis of each consultation to determine whether the consequent proved accurate, and, if not, what the correct consequent was, and how its antecedents differ from those that generated the wrong consequent. This is a kind of failure-driven learning, but it is by no means automatic.

The second aim does not necessarily require external confirmation, but assumes that the knowledge in the system is accurate. An episodic memory can be built up for future use in re-inducing the knowledge base (see, e.g., Kolodner,

1983); or this learning can be done incrementally. If the episodic memory shows that 90%, say, of all consultations result in a certain consequent, it may be appropriate to reorder the consultation so as to rule out this commonest, or most probable, consequent first.

Gaines (1989) has proposed an architecture for knowledge support systems, which eliminates the false dichotomy between acquisition and performance.

Conclusions

This chapter has explored two routes to the same objective. One route, knowledge acquisition from examples, is direct and can be either structured, unstructured, or a mixture of both. The other route, the repertory grid technique, is somewhat indirect, but efficient in that a single procedure gathers the partial domain model and a sufficiency of cases. Figure 8 on page 55 depicts the stages in knowledge acquisition from examples at the left, the repertory grid stages at the right, and the shared stages down the centre. With both approaches, machine induction or some other learning process can be used to distill the knowledge into an efficient knowledge base.

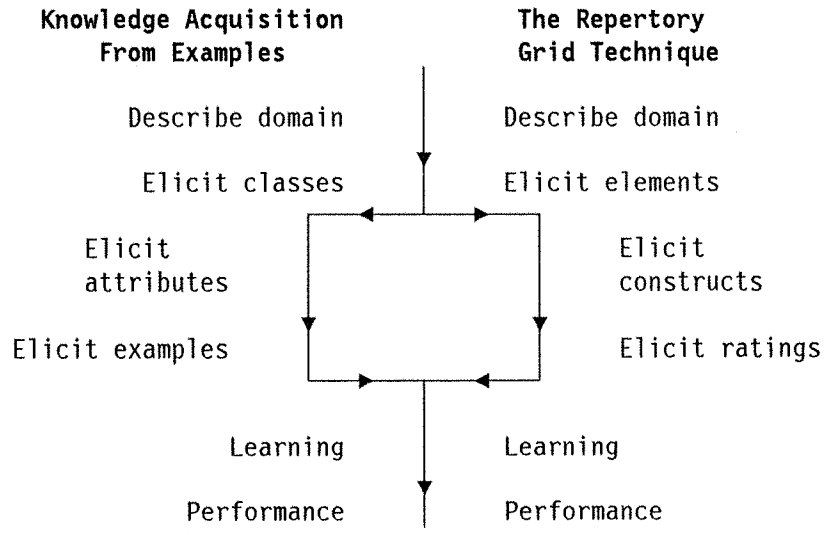

Figure 8. Two routes to a single objective

It has also been shown that the elements used in the repertory grid technique should be examples rather than classes. Otherwise, the knowledge acquired would not include important disjunctive relationships. The elicitation of elements should therefore be modified to focus on examples.

Some writers (e.g., O'Leary & Watkins, 1990) argue that different methods acquire different types of knowledge, and

> individual task comparisons of different forms of knowledge acquisition may understate or misstate the problems of interest to developers of expert systems.

However, if two methods are shown to produce the same kind of knowledge, then it is indeed useful to compare their relative efficacies and efficiencies. The preceding discussion has shown that the repertory grid technique and knowledge acquisition from examples do acquire the same kind of knowledge in different ways.

If, as in knowledge acquisition from examples, the domain expert has to think of suitable distinguishing attributes without much prompting, it can be fairly difficult. So any method that seeks (as the repertory grid technique does) to make these attributes easier to access and articulate is likely to speed up this part of the knowledge acquisition process. But there is little evidence to support or refute this conjecture.

Moreover, if the problem space (or domain model) is viewed as multi-dimensional, the attributes provide the dimensions of such a space. Each domain object (or class, or element) is located in the space at a point that can be expressed as Cartesian coordinates. The coordinates of all the classes can be organised as a matrix (as in Figure 7 on page 49). Whereas this matrix is built up one column at a time by the repertory grid technique, it is built up one row at a time by knowledge acquisition from examples. But it is by no means clear which way would be expected to be more efficient or effective.

Compilation of the expert's knowledge (Anderson, 1982) into procedural memory makes it futile to ask directly for rules; but examples can be successfully elicited. It may well be that experts find it easier to compare cases rather than to elaborate a partial domain model and describe individual cases. But there is little evidence to support this conjecture.

It is evident that light needs to be shed on a number of questions. Without answers to some of the questions raised above, it is difficult to plan knowledge acquisition activities confidently. It is also difficult to select knowledge acquisition methods intelligently.

Chapter 3. Some Implications of Cognitive Psychology for Knowledge Acquisition

Abstract

Explanations of why one knowledge acquisition method may be more efficient or effective than another are likely to originate from consideration of not just the knowledge engineer's techniques but the expert's mind as well. In particular, it is enlightening to consider theories about how knowledge is stored in the mind and how it is retrieved. This chapter discusses these notions and some hypotheses implied by them. The hypotheses relate to two knowledge-acquisition methods: the repertory grid technique and knowledge acquisition from a minimal set of examples.

Introduction

A domain expert's performance is a product of what s/he knows. This knowledge is developed, stored, and retrieved in certain ways (Anderson, 1982; Kolodner, 1983). Any attempt to elicit a domain expert's knowledge is therefore likely to trigger some amount of activity inside his or her mind. The nature of this activity is influenced by several factors including the knowledge-acquisition method used. Indeed, the pace with which knowledge is elicited and the nature of the knowledge obtained appear to depend on the extent to which the knowledge-acquisition method encourages appropriate mental operations to take place.

An important part of any comparative analysis is to understand the two methods being compared. The following two perspectives seem relevant to this understanding:

- External processes, stages, and outcomes, e.g., how the methods can be described, what they acquire, and what is done with the data obtained

- Internal (cognitive) processes, i.e., how the knowledge sought is represented in the expert's mind, how it can be retrieved, and what happens when particular techniques are used in trying to elicit it.

The two methods to be compared are the repertory grid technique and knowledge acquisition from a minimal set of examples (KAMSE). The external features of the two methods have already been compared in Chapter 2, "Learning Without Case Records: a mapping of the repertory grid technique onto knowledge acquisition from examples." This chapter is about the internal processes.

There are several levels of abstraction at which mental representation and retrieval can be modelled and understood. One of the lower levels is the neurophysiological; one of the higher ones is the cognitive. Although a complete mapping between these two levels is yet to emerge, cognitive psychology is concerned with modelling the mind and mental processes, without detailed consideration of the brain and central nervous system (Johnson-Laird, 1983; Anderson, 1976). This high-level view focuses on cognitive structures (e.g., memory) and processes (e.g., attention, learning, and recall).

But what is taking place in the mind cannot actually be observed. What can be observed is behaviour and perhaps stimuli that affect this behaviour. So, in some ways, the mind is a black box; and it is possible to propose different models that produce identical input-output behaviour. Such models are judged, not on how well they represent what is taking place inside a person's head, but on their predictive power; that is, how well their input-output behaviour corresponds to that of the system they seek to model and explain.

This chapter discusses the structure of human memory, and how information is stored in and retrieved from that memory. These notions provide

a framework for reasoning about what might be happening in the domain expert's mind while his or her knowledge is being elicited by either the repertory grid technique or KAMSE. Assuming the validity of these conjectures, some hypotheses are stated about the two techniques.

Divisions in Memory

Although questioned by a few cognitive psychologists (see, e.g., Morris, 1988, p 91), it is generally agreed that there are at least three kinds of memory: long-term, short-term, and perceptual.

According to Gilmartin, Newell & Simon (1975), information from the outside world enters the system through sensory organs (e.g., eyes and ears) and into sensory-related buffers. There are "two buffers in series for each sensory modality (a sensory store and an imagery store)". When a person receives a sensory input (whether visual, auditory, or other) from the external world, this input is held very briefly in perceptual store, then transferred into short-term memory.

Short-term memory is a controversial concept. Baddeley (1976) refers to studies of short-term memory as "concerned with short-term forgetting". Some writers (e.g., Baddeley, 1976; Posner, 1973; Anderson, 1983) use the expression "short-term" or "working" memory to mean a limited-capacity input store. Posner (1973, p 16) argues that short-term memory

> provides a system within which incoming information can be related to previously stored information ... providing a means of reorganizing and updating long-term memory.

Others (e.g., Mandler, 1986) argue that there is no limit to the capacity of short-term memory. Consciousness is the limited-capacity mechanism, while short-term memory is those contents of long-term memory that are activated

but not currently in consciousness. These writers also refer to short-term memory as "active memory" (see, e.g., Shwartz & Kosslyn, 1982) and "immediate memory" (see, e.g., Case, 1980).

The contents of short-term memory can be quickly replaced by subsequent input, but can also be retained by recycling (e.g., repeating a name just heard, over and over). Information can also be brought into short-term memory from long-term memory (as when we recollect, reflect, or imagine).

In contrast, long-term memory is a very large-capacity, very low-loss, store of information. Mandler (1986) sees long-term memory as the sum total of a person's life experience. He also argues that long-term memory is the unactivated portion of these memories. Some psychologists further subdivide long-term memory into different areas. For example, Tulving (1985) has proposed a "tripartite system: episodic, semantic, and procedural memories". Although some writers (e.g., Snodgrass, 1989) approve of Tulving's model, others (e.g., Baddeley, 1976) argue that the episodic-semantic distinction is too sharp. According to Claxton (1980, p 230), Tulving "makes the error of assuming that, because people can make judgements about knowledge, these judgements directly reflect basic principles of storage". Mandler argues that episodic memory is simply a subsystem of semantic memory, not a different kind of memory.

In addition to the changes in long-term memory brought about by information from short-term memory, access to some information in long-term memory may be lost through forgetting. The more or less permanent information stored in long-term memory includes:

- Events
- Schemata (or prototypes)
- Cultural values, attitudes and beliefs

- Knowledge underlying cognitive skills.

But these items are thought to be stored at different levels in long-term memory. Figure 9 shows the kind of information stored in short-term memory and at Tulving's three levels in long-term memory. Each kind of information is described briefly below.

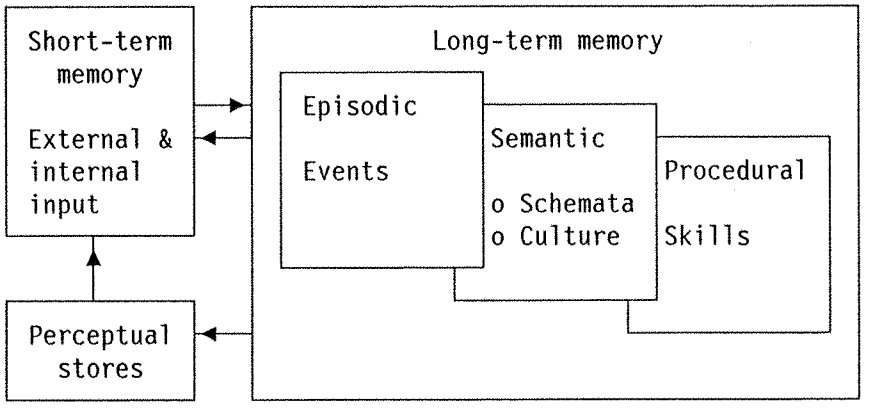

Figure 9. Memories, and types of information thought to be stored in them.

Episodic memory is described by Cohen (1989) as "memory for personal experiences ... [consisting] of subjective specific facts". According to Morris (1988, p 105), "episodic entries retain information about their place and time of occurrence", which is really another way of stating Aristotle's theory of association: recalling one event can trigger the retrieval of other events associated by contiguity, whether in space or time (see, e.g., Dalton, 1988). Mandler (1986) contends that some episodic entries may be conceptually, rather than temporally, organised. This might account for the fact that people sometimes recall an incident, and then have difficulty remembering where or when it happened. Kolodner (1983) argues that the richness of episodic memory is one of the main factors distinguishing an expert from a novice.

Schemata are knowledge structures that represent concepts or objects in generalised, idealised, or stereotyped form. For example, a person's schema for an elastic band might include the following information: made of rubber, can stretch, shaped like a flexible loop. This schema is generated from the person's experience of encountering elastic bands. It is a distillation of the essential characteristics of every elastic band the person has ever seen. A person's schemata help him or her to understand the world (Bartlett, 1932; Rumelhart, 1975; Noble, 1989).

So, for example, the next elastic band a person sees will be recognised as such, because it matches the person's schema for elastic bands. Conversely, items not matching this schema will be recognised as not being elastic bands. But Bartlett also points out that schemata are "active developing patterns"; so, although they shape a person's interpretation of things seen and events experienced, schemata are in turn generated and refined by a person's experiences.

Although Figure 9 on page 61 shows episodic memory as separate from semantic memory, the latter appears to have some influence on what is stored in and retrieved from the former. According to Bartlett (1932), people do not remember episodes as a whole, but rather they have memories of the essence of events based on their schemata. And when they recall past events, it is not so much a retrieval as a reconstruction. This reconstruction involves retrieving episodic memory of peculiarities about the specific event, and combining it with a schema for the type of event being recalled (see Bower, Black & Turner, 1979). Baddeley (1990; see also Rumelhart & Norman, 1985), notes that frames (Minsky, 1975), scripts (Schank, 1975, 1982), and schemata, although not identical, are very similar concepts.

Cultural knowledge is deeply rooted and generally not available to conscious retrieval, but it governs the way people think and act (see, e.g,

Connerton, 1989; Hofstede, 1980). People's minds can store these pieces of information unconsciously, not just at an early age, but throughout adult life as well. This ability is reflected in the phenomenon of corporate culture (see, e.g., Marshall & McLean, 1985), which is in many ways similar to national culture — but rather than being characteristic of members of a society, typifies employees in a company or members of a club.

Skill is the knowledge underlying many forms of expertise. This knowledge is improved by experience or practice and manifests itself in increased competence at performing a particular task. But although increased skill is manifested in improved performance, it is also often accompanied by a difficulty in explaining exactly how this performance is achieved. Johnson-Laird (1983, p 465) is one of several psychologists to point this out:

> You can never be completely conscious of how you exercise any mental skill. Even in the most deliberate of tasks, such as the deduction of a conclusion, you are not aware of how you carried out each step of the process.

That is not to say that a person may not try to explain; but, if Johnson-Laird is right, such "*post-hoc* rationalisations" (discussed on page 67) are not guaranteed to be reliable.

Retrieval of Information

Information in long-term memory can also be classified according to the means by which it can be retrieved. In particular, information can be at the conscious or unconscious level. Anderson (1982), for example, has argued (but he is by no means the only one; Posner, 1973; and Bainbridge, 1986, advance similar arguments) that the knowledge related to a cognitive skill can also be distributed between the conscious and the unconscious within long-term

memory. Strictly, conscious and unconscious describe the way information can be retrieved rather than anything about the information itself.

As shown in Figure 10, Posner (1973, p 43), distinguishes between "effortless retrieval [which] occurs when the input contacts its address in memory without any conscious search" and

> effortful retrieval [which] occurs when the subject is forced to search the items retrieved into active memory, or when he does not have sufficient content to locate the items in long-term memory unambiguously.

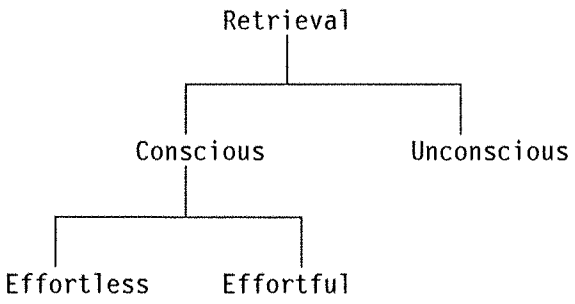

Figure 10. Modes of retrieval from long-term memory

But even "effortful" search can retrieve only information that is subject to conscious retrieval. As mentioned on page 10, Anderson refers to the consciously retrievable knowledge as "declarative" and to the unconscious knowledge as "procedural". According to Anderson (1982), the distribution of knowledge underlying a skill shifts from the conscious to the unconscious as a result of practising the skill. It is not that the declarative knowledge is lost; the declarative knowledge remains subject to conscious retrieval, while the productions underlying expert performance are developed beyond conscious reach.

Anderson's ACT* (adaptive control of thought) theory implies, for example, that when a business administration student first learns about "cash

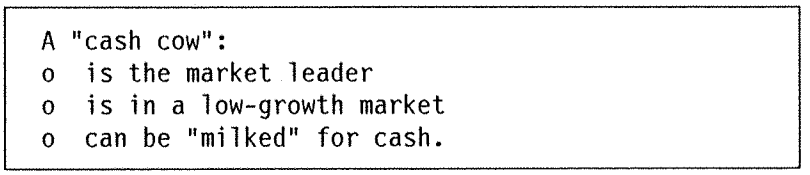

Figure 11. An example of declarative knowledge

cows" (see, e.g., Kotler, 1984), the knowledge is probably stored as the set of facts shown in Figure 11 on page 65. This is not Anderson's example, but it is consistent with ACT* theory.

This is an example of declarative knowledge and is probably (as discussed later) stored as a schema in semantic memory. Once this knowledge is deployed in the performance of some act (e.g., analysis of a marketing strategy case), it is used in the form of productions. ACT* theory implies, in this instance, a set of productions of the form shown in Figure 12 on page 66.

```
IF the goal is to classify a strategic business unit (SBU)
THEN a subgoal is to see what the SBU's competitive
     position is like.

IF the goal is to classify an SBU
AND the SBU's competitive position is like that of a "cash cow"
THEN a subgoal is to see what the SBU's market growth is like.

IF the goal is to classify an SBU
AND the SBU's competitive position is like that of a "cash cow"
AND the SBU's market growth is like that of a "cash cow"
THEN the SBU is a "cash cow",
     POP the goal.

IF the subgoal is to see what the SBU's competitive
     position is like
AND the SBU is the leader in its market
THEN the SBU's competitive position is like a "cash cow's"
     POP the subgoal.

IF the subgoal is to see what the SBU's market growth is like
AND the SBU's market growth is low
THEN the SBU's market growth is like a cash cow's,
      POP the subgoal.
```

Figure 12. Tentative productions constructed from declarative knowledge

Anderson (1983, p 30) asserts that in each production of this kind, the condition portion

> specifies some pattern that should be active in working memory, and the action specifies some cognitive or external operation that will be performed if the pattern is matched.

Repeated use, according to Anderson (1982), causes these tentative productions to be unified, two at a time. So, eventually, after using the knowledge a large number of times, a business strategist might end up with a more compact production, as shown in Figure 13 on page 67.

```
IF the goal is to classify an SBU
AND the SBU is a market leader
AND the SBU is in a low-growth market
AND the SBU can be "milked" for cash
THEN the SBU is a "cash cow",
     POP the goal.
```

Figure 13. Example of a compiled production

As this unification of productions (knowledge composition) takes place, the productions are stored beyond the person's conscious access. So, if asked why s/he says that a particular SBU is a "cash cow", a person might have difficulty answering, or might rationalise by constructing an answer from declarative knowledge. According to Anderson, compilation accounts for the speedup phenomenon that accompanies repeated use of a skill. It should be noted that this procedural knowledge, although shown as compiled productions, is essentially a mapping between a set of attribute values and a class. The productions provide a convenient and intelligible notation; but the essence of ACT* theory remains intact even if this knowledge is represented in the model as a neural network.

One argument against Anderson's procedural/declarative divide is that it is too closely based on computer models, with declarative knowledge being analogous to data, and procedural knowledge being analogous to programs. But Cohen (1989) is among those who affirm that ACT* theory explains most of the phenomena observed in expertise formation. However, she points out that (Cohen, 1989, p 177) it does not explain "the effects of emotional and attitudinal factors" (Morris, Tweedy & Gruneberg, 1985), or "the blurring of conceptual boundaries" (Murphy & Wright, 1984). Anderson (1989) has recently refined and updated ACT* theory. The new theory, embodied in a model called PUPS, includes "analogy-based generalization, a discrimination

mechanism, and principles of making causal inferences" among its induction mechanisms.

It can also be argued that object identification (see, e.g., Marr, 1980; Ellis & Young, 1988) is based on schemata. The first time a child sees an elastic band and learns what it is, s/he might develop a schema for elastic bands (see, e.g., Flavell, 1985, pp 48-54). This schema might include the size, shape, texture, flexibility, material, elasticity, and perhaps even the colour of such an object. When asked what elastic bands are like, s/he might say: "I've only ever seen one, but I think they are stretchable, they are about this size, brown, and made of rubber". The next one the child sees might be very similar to the first and would therefore reinforce the schema. Perhaps eventually the child will encounter an elastic band very different in size from any s/he has experienced before. The child might well be able to recognise the strange elastic band for what it is; but the match with the schema will be imperfect. This experience will cause the schema to be refined, perhaps to indicate that the size can be within some range. There is, however, little justification for supposing that the development of schema precludes that of productions, or *vice versa.*

The focus on conscious retrievability implies a model of long-term memory as shown in Figure 14 on page 69. When viewed alongside this model, Tulving's (1985) semantic memory appears to be a half-way house spanning the consciously and unconsciously retrievable contents of long-term memory.

Long-term memory	
Conscious o Episodes o Declarative models	Unconscious o Automatic productions

Figure 14. Retrievability of long-term memory.

One of the explanations that Johnson-Laird offers for unconscious retrieval of knowledge is that several strands of information are being used in parallel. So, according to Johnson-Laird (1983, p 468),

> any attempt to use introspection in order to become conscious of something that is normally unconscious is unlikely to succeed. Not only is the information inaccessible, but also an essentially parallel process has to be grasped by the serial deliberations of [our introspection mechanisms].

The discussion so far has focused on the possible structure of memory, and on the kinds of information stored in short-term and long-term memory, and split between conscious and unconscious retrieval. According to Posner (1973), there are three codes (formats) in which information is represented in these memories. First, there is an iconic code which represents images perceived, whether visually, auditorily, or by other sensory means. Second, there is a symbolic code that represents mainly words. And thirdly, there are motor codes that represent skills in doing physical things (like balancing the end of a broom in the palm of one's hand, riding a bicycle, or serving a tennis ball).

It is also interesting to reflect on the similarity between knowledge stored as motor codes and knowledge stored as compiled productions. Productions behave very much like motor codes: in an intellectual skill,

involving observation, discourse, and decision, a person may not be able to say how s/he actually makes a decision — in the same way as it is almost impossible to describe the procedure for riding a bicycle. Motor skills are similar to intellectual skills in that a first attempt to ride a bicycle is likely to end in failure. By continued practice (and what Schroder, Frank, Kohnert, Mobus, & Rauterberg, 1990, refer to as "failure-driven learning"), a person can reach a point where s/he has learnt how to succeed rather than fail. After that point, practice increases proficiency and speeds up performance (in what Schroder *et al*, 1990, call "success-driven training"); and the underlying knowledge is retrieved only by performance of the skilled act.

Implications for Knowledge Acquisition

As noted above, ACT* theory asserts that knowledge compilation in experts prevents their performance knowledge from being consciously retrieved. The compiled automatic productions are retrievable only by performance of the skilled act. But this performance knowledge is developed as a function of declarative knowledge (and experience in applying it), as a person progresses from being a novice to being an expert. Even after knowledge compilation, the declarative knowledge is not forgotten, but remains in long-term memory, and can be retrieved by appropriate cues. However, this declarative knowledge is not what the expert uses in performing the skilled act.

Fortunately, declarative knowledge can be transformed into productions that model expert performance. When eliciting knowledge from experts, it is therefore worth encouraging them to retrieve and articulate the pieces of declarative knowledge from which accurate productions can be generated. The minimum knowledge needed to model a simple classification decision (see, e.g., van Melle, 1981) consists of the following kinds of knowledge units:

- Classes (the possible outcomes of a classification decision in the domain)

- Attributes (factors to be considered in making the expert decision between possible classes), each in the form of
  - An attribute descriptor
  - Attribute values (the possible values that an attribute can assume).
- Information to link patterns of attributes to particular classes (for instance, rules, decision trees, or an artificial neural network, which can all be generated by detecting regularities in either of the following):
  - Examples or cases (instances of the domain classes expressed in terms of attribute values)
  - Ratings (estimates of the degree to which particular attribute values describe particular classes).

These are among the units processed by knowledge-acquisition tools such as AQUINAS (Boose & Bradshaw, 1987), FMS-Aid (Garg-Janardan & Salvendy, 1988), and KITTEN (Shaw & Gaines, 1987); and expert-system shells like 1st-Class. Their successful use in these tools demonstrates their adequacy for at least some kinds of analytic knowledge-based systems. The classes and attributes comprise a partial domain model, while the linking information, which is an important part of an expert's compiled knowledge, enables a representation to provide the basis for inference.

While these units are sufficient to model a simple decision, many problems involve making several such interdependent decisions. For example, MYCIN makes two: diagnosis and treatment. Some tasks (e.g., developing a plan) require effortful solution by the expert. Even these tasks are likely to be subdividable into several simple decisions, the results of which can be used to tailor a skeletal plan into a specific plan.

As noted in Chapter 1, "The Knowledge Acquisition Problem," several methods have been used to elicit knowledge units from knowledge sources (for a

list of methods, see, e.g., Neale, 1988; Foley & Lehto, 1989). The rest of this chapter, however, concentrates on two methods: the repertory grid technique and KAMSE. The two methods can be subdivided into five parallel stages as shown in Figure 15. Although the stages are shown as having distinct boundaries, they often overlap. For example, a list produced at stage 1 might need to be augmented when subsequently elicited attributes and examples act as cues to elicit further classes. Stages 4 and 6 are computer-intensive, while the other stages demand mental activity from the expert.

Stage	Repertory Grid	KAMSE
1	Listing all elements that exemplify the domain classes	Identifying the classes that cases in the domain can belong to
2	Identifying constructs that distinguish elements from each other	Identifying the attributes (e.g., supply voltage) considered in deciding the class of a case, and listing the values (e.g., 3V, 5V, 24V) that each attribute can have
3	Rating all elements on each construct elicited	Describing, without repetition, examples of all classes in terms of attribute descriptors and values
4	Using machine induction to find regularities, and distill the grid into a knowledge base	Using machine induction to find regularities, and distill the examples into a knowledge base
5	Classifying a set of exemplars and using these to evaluate the knowledge base produced in the previous step.	Classifying a set of exemplars, and using these to evaluate the knowledge base produced in the previous step.

Figure 15. Stages in the repertory grid technique and KAMSE

An analysis of only the external features of the two methods (see Chapter 2, "Learning Without Case Records: a mapping of the repertory grid

technique onto knowledge acquisition from examples") does not make clear whether the repertory grid technique or KAMSE would be more efficient or efficacious, although both methods are capable of producing identical results on a given domain. But, from the standpoint of the expert's mental activity, several hypotheses can be stated. Five of these are discussed and presented in the sections that follow. In these discussions, the expert's mental activity is often described in terms of a knowledge domain involving identification of objects.

Eliciting Domain Elements

Gammack & Anderson (1990) stress the importance of context in determining both the information elicited and the meaning to be given to it. In two different contexts, the same expert might give different information. For instance, Gammack & Anderson (1990, p 20) point out that "the similarity of Pepsi-Cola to Coca-Cola depends on ... the third item in the comparison". But if two knowledge-acquisition methods contain a stage in which the interaction with the knowledge source is the same, it is reasonable to expect that on the same domain and in the same context, the shared stage of the two methods will elicit identical knowledge and demand equal amounts of effort from the knowledge source. Stage 1 of the two methods fits this description: the expert identifies and lists all classes (in KAMSE) or elements (in the repertory grid technique).

If the objects, or facsimiles, are present (e.g., specimens in object identification, or historical records in some other domains), recognition (i.e., knowing that you have seen an object before) and then reminding (i.e., recognising the object activates the word that describes it) may occur. Such retrievals are likely to be quick and effortless.

If the objects are not present, but logically related to each other (e.g., things you would expect to find in an office, or in a desk drawer, or on the menu in an Indian restaurant), their normal location in the world probably acts as a cue for retrieval from episodic memory, and activation might spread to other items seen in the same location. Some of the items might be retrieved effortlessly; some might require effortful search, perhaps involving visualisation of the item's milieu (as when a person tries to recall all the windows in a familiar room). Activation may also spread to items related temporally (e.g., all the ailments diagnosed on a particular day).

Hypothesis 1: The elicitation of elements in the repertory grid technique is very similar to the elicitation of classes in knowledge acquisition from examples. When both prototypes and exemplars are used as elements, the repertory grid will require the greater effort. In any event, eliciting classes is not likely to ever require more effort than eliciting elements. If the elements elicited are all prototypes, then the two methods are identical at this stage; and no difference is expected, between methods, in the mental effort required of the expert at this stage of the process.

Eliciting Attributes and Constructs

In the repertory grid technique, constructs (bipolar attributes) play an important part. Both poles of a construct are vital because they delimit the construct's "range of convenience" (Ford, Petry, Adams-Webber & Chang, 1991). And a construct is valid only within its range of convenience. As discussed in Chapter 2, "Learning Without Case Records: a mapping of the repertory grid technique onto knowledge acquisition from examples," across the two methods, there is an analogy, or even equivalence, between attributes and constructs. But the attribute values in terms of which examples are described under KAMSE have a single pole.

Eliciting Attributes

During stage 2 of KAMSE, the expert has to list a set of distinguishing attributes (declarative knowledge) without conscious access to the automatic productions that contain references to them. But s/he can see the list of classes that s/he developed in stage 1. It is by considering the items in this list that the expert is able to identify distinguishing attributes. The limited capacity of working memory may not accommodate all the items along with any other information. Subsets must therefore be focused on at any given instant. But, because the list is there, schemata of all the items are probably activated in long-term memory.

If the expert can look at only the list of classes, several strategies for finding the distinguishing attributes might be used (focus on one, two, or some other number of classes at a time to see how they can be described, then using these attributes on some of the other classes to see if they are useful as discriminators). The expert might observe: "These things are made of different materials and have different shapes". So s/he might decide that shape and material are two of the important attributes.

If the expert can look at both the list of classes developed in stage 1, and the items themselves, s/he can compare the actual items rather than merely their schemata. This gives the expert more options than not having the items present; and some people employ sorting (physically separating out the items currently in consciousness) as a strategy to help them become aware of distinguishing attributes.

Whether the items are present or not, the expert is free to choose how to consider them, what groupings to make, in trying to become aware of the distinguishing attributes. In forming these groups, the expert is trying to interrelate different pieces of declarative knowledge. Although the attributes that a person articulates might not be sufficient to make all the required

distinctions, they do provide a language for starting to express the examples. During stage 3, there may be occasions when the expert has to return to the search for distinguishing attributes, but at such times, the search is aimed at resolving confused pairs; so two items at a time are compared there. But the attribute definition process depends on comparing the objects to find how they differ, and on defining these differences as new attributes. Whether it takes place all at once or as a consequence of describing examples, it is likely to require comparable amounts of mental effort.

Having identified an attribute, the expert then has to develop a list of values that are adequate for describing all the items. For instance, when a person thinks of the attribute "material", there are a vast number of materials in the world: cloth, paper, brass, steel, different kinds of plastic, rubber, and countless others. But not all of these are important in distinguishing among the items in the domain. To articulate the values that are important, a person actually considers the items in the list, thereby being reminded that some of the items are made of metal, some of plastic, and some of rubber, and that these three are sufficient values of the attribute "material" to distinguish among the items.

Eliciting Constructs

In the repertory grid technique, the situation is a bit different when the expert tries to identify constructs for distinguishing among the set of domain elements.

If the elements are present, the expert may select the triad physically, and look at its members for differences. Whether the elements are present or not, s/he may compare the schemata of the items in consciousness, trying to find constructs on which they differ. S/he focuses consciousness on the three schemata, trying to find differences between them.

The expert is not left to make her own choices about which items to consider, and what groupings to put the elements into. Rather, s/he is constrained to consider the elements in sets of three, and neither the composition nor the size of the sets is of her choosing. Kelly (1955) prescribed a set size of three elements for what appears to be as much a mathematical as a psychological reason: three elements form the minimum set from which it is possible to find two that are similar and one that is different. But it may or may not be true that, being left to their own devices, subjects would actually select items in threes to consider.

Kelly found that using the repertory grid technique enabled him to elicit the inner worlds of his patients. However, it is not clear whether he could have found an easier method, or whether the technique was easier on him but harder on his patients. It is not clear whether he could have found a direct style of questioning that would have requested the information he sought, in such a way that the patients would have been able to provide it without much effort. Of course, Kelly did not want them to employ defence mechanisms to cover up how they actually felt, by giving answers that were not painful for them.

It is also not known whether domain experts have the same kinds of inhibitions about stating what they think their knowledge consists of. Although it is clear that the repertory grid technique is useful and effective, it is not clear whether it is optimal or whether it restricts the domain expert. The repertory grid technique is potentially inefficient in that it has no safeguards against the repeated elicitation of parallel constructs. A few parallel constructs may be useful in helping the expert clarify her thinking, but in general they are redundant and therefore largely wasted. But it seems reasonable to assume that the technique is more efficient than asking directly for attributes.

Hypothesis 2: A set of constructs (or attributes) adequate for distinguishing among all the elements (or classes) in a domain can be elicited

more easily with prompting, as provided in the repertory grid technique, than without, as in KAMSE.

Traversing an Inference Matrix

If the classes in a domain (for example, classification of a strategic business unit) form the set E = {question mark, star, cash cow, dog} and the attributes form the set A = {growth less than 10%, relative market share less than 1, a lot of cash required, highly profitable}, there is a kind of connection matrix between the two sets. This matrix, which is shown in Figure 16, is what is referred to as the inference matrix.

	Growth < 10%	Relative mkt share < 1	A lot of Cash required	High Profits
Question mark	No	Yes	Yes	No
Star	No	No	Yes	No
Cash cow	Yes	No	No	Yes
Dog	Yes	Yes	No	No

Figure 16. An inference matrix for classification of a strategic business unit

In KAMSE, the range of applicability of the knowledge elicited is enhanced if the minimal set of examples used includes both prototypes (schemata) and exemplars (atypical examples). This is also true of elements in the repertory grid; so the inference matrix is the same, whichever of the two methods is being used. The repertory grid technique elicits the inference matrix one column at a time, whereas KAMSE elicits the same matrix one row at a time (see Chapter 2, "Learning Without Case Records: a mapping of the repertory grid technique onto knowledge acquisition from examples").

It is also interesting to consider the problem space as opposed to the mental model. The repertory grid technique is based on the notion that a person's micro-world can be represented in a multidimensional space, with each

dimension accommodating a construct. This space can map onto a two-dimensional inference matrix.

However, Tulving's (1962) theory of subjective organisation asserts that people organise information to make its use easier, but this organisation bears little relationship to the way the information is normally stored in the person's long-term memory. Gammack (1987) argues that the representation is simply a metaphor to help people discuss and partially appreciate what is going on. Whatever the mental representation, it is clear that classification knowledge can be mapped onto this multidimensional space.

Eliciting Examples

An example, which might be reconstructed from semantic and episodic memory, is described in terms of attribute values and a class. In describing an example, the expert considers the domain object and focuses on the values that it has for the various attributes. Having made the observations (or while making them), s/he articulates the relevant values.

On the first object for which the expert describes an example, s/he will simply consider the object and state the values that it has on the attributes. However, when s/he focuses on a subsequent object, the attribute values that s/he states may be identical to those for a previously stated example of some other object. For instance, if s/he identified "material" and "shape" as the attributes, as discussed in "Eliciting Attributes and Constructs" on page 74, and s/he used a one-penny coin as the first example, then the attributes will be sufficient to describe the penny (which has the values metal and round). The adequacy of the two attributes is tested only when a similar object is focused on. For instance, if the expert takes the one-cent coin as the second case, then s/he will be stating "metal" and "round" again. And those attribute values will

be identical to the ones for the penny in the previous example, creating a confused pair.

The only satisfactory way of resolving this conflict is by finding a new attribute on which the two objects differ. For example, "engraved figure" with values of "woman's head", "man's head", and "other". This process of new examples taking the existing attribute set to its limits and exposing its inadequacy is one that continues until examples of all the domain objects have been described. So, there is first a process of considering the values that the object has on existing attributes. There is sometimes the additional process of comparing a confused pair of objects to identify an attribute on which they differ, and then defining that attribute. Of course, it is possible for a sufficient set of attributes to be identified during the initial definition of the attributes in stage 2, so that subsequent expression of examples in stage 3 is smooth, without any return to stage 2.

Eliciting Ratings

After a construct has been identified, the expert is asked to assign a rating to each item to reflect the degree to which the item possesses the feature. Because the construct is personal, whatever a domain expert states as a constructs, it ought to be relatively simple to rate all the elements on it. That is as long as the constructs are defined to be generally applicable to elements outside of the triad.

It might be expected that people inexperienced in using the repertory grid technique will sometimes articulate constructs that are not generally applicable to elements outside of the triad. For example, if the triad consisted of "penny", "paper clip", and "elastic band", an inexperienced user of the repertory grid technique might introduce a construct of "made of metal" as one pole and "made of rubber" as the opposite pole. When faced with rating other

elements, this person might have difficulty rating the plastic paper clip, because it is made of neither metal nor rubber. A scale with a central point rescues such persons, and allows them to rate the element as neither pole. Those with more experience at using the technique might have decided to use the construct "made of metal" / "not made of metal". They would probably have found it easier to rate all elements. Indeed they would have been able to assign extreme ratings to all the elements.

Hypothesis 3: There is little reason to expect an expert to find it easier to express cases in terms of attribute values than to identify similarities and differences between examples of classes.

Efficacy of Method

Efficacy differs from efficiency (see page 82) in that, whereas the latter takes account of effort for results, the former is concerned merely with results. So a method is efficacious if it produces the desired results. Efficacy can therefore be defined as the diagnostic accuracy of the knowledge acquired. So, one method could be considered more efficacious if, regardless of how much effort it required (within reason), using it resulted in more accurate knowledge.

A word about accuracy is appropriate here. Accuracy is a measure of the ability of the knowledge base to arrive at decisions with which the knowledge source (domain expert) agrees. If the expert's decision is noted in n cases, and these cases are then presented to the knowledge-based system, the KBS will in general arrive at the same decision as the expert in m cases (where $m <= n$). The accuracy of the knowledge base is computed as m/n. This measure has also been referred to in the literature as diagnostic, classification, and predictive accuracy.

Adequate accuracy is normally a necessary (although not always sufficient) quality that a knowledge base must have if it is to become

operational. But it is difficult to find any reason why a knowledge base developed from the repertory grid technique should be either less or more accurate than one developed using KAMSE.

Another possible measure of efficacy is the number of knowledge units acquired. Knowledge units are the entities that can be observed as increasing in number when a successful attempt is made to elicit an expert's knowledge. They are the entities acquired that are useful for building a knowledge base. Among these are the kinds of units listed on page 70.

But it is difficult to say whether a larger number of knowledge units (Burton *et al*, 1990) means that a method has more efficacy. In some ways, it depends on one's point of view. Wilkins (1987) expresses the view that a method that acquires all the required knowledge as a small number of units is more efficacious than one that acquires all the knowledge as a large number of units.

Hypothesis 4: Overall, for the entire knowledge acquisition cycle (stages 1 to 5 in Figure 15 on page 72), there is no efficacy difference between the two methods; the knowledge base generated by induction will be equally accurate between methods.

Efficiency of Method

Efficiency can be defined as effort per unit of knowledge acquired. According to Dhaliwal & Benbasat (1990, p 149), it is

> associated with the resources expended in the development of a knowledge base. The effort, cost and time of the expert(s) and the knowledge engineer(s) are the major determinants ...

Burton, Shadbolt, Rugg, & Hedgecock (1990) measure efficiency "in terms of effort for gain". The effort is measured in elapsed minutes, while the

gain is in the number of clauses acquired. However, their misgivings about the appropriateness of number of clauses as a measure of gain led them to perform a further experiment in which the experts evaluated the acquired knowledge to provide a more meaningful estimate of the gain.

One method would be said to be more efficient if using it resulted in acquiring a given amount of knowledge with less effort. We have already predicted, in the previous sections, that the two methods require the same amount of effort at stages 1 and 3, but that at stage 2 the repertory grid technique requires less effort than KAMSE. If this is true, and the differential effort is large enough, it will be reflected as a difference in the overall process (stages 1 to 5), unless there is a compensating difference at the evaluation stage. The author was of the opinion that the differences in effort would cancel each other out for the overall process.

Hypothesis 5: Overall, for the entire knowledge acquisition cycle (stages 1 to 5 in Figure 15 on page 72), there is no efficiency difference between the two methods.

Conclusions

Anderson's ACT* theory, because it accounts for most of the observed phenomena relating to experience and performance, is useful for proposing plausible accounts of what goes on in the domain expert's mind. Firstly, becoming an expert appears to be an iterative process of enriching episodic memory, refining schemata and composing productions. Secondly, expert performance is largely driven by these productions. But episodes, schemata, and productions are not retrievable to the same extent, thus imposing limits on knowledge acquisition for knowledge-based systems. Any attempt to capture a person's expertise must therefore contend with the retrievability constraints of these knowledge structures.

Although cognitive psychology does not allow many conclusions to be drawn with certainty, it does provide some insight into the domain expert's mental activity during knowledge elicitation. It is useful to consider this activity in trying to explain why one knowledge acquisition method might perform with more efficacy or efficiency than another. Effective methods need to facilitate those mental processes that result in the expert articulating the knowledge units required to build an accurate knowledge base. These units can be structured and processed to produce a useful knowledge base.

The repertory grid technique and KAMSE acquire knowledge, not only of a very similar kind, but also in stages that are directly parallel and analogous. At some stages, the activity taking place in the domain expert's mind appears to be similar between methods. Although the style of the two methods is different at the attributes/constructs and examples/ratings stages, the expert's mental activity often appears to be quite similar. Even where the expert's mental activity appears to differ between methods, there is little reason for expecting one method to demand of the expert a greater effort than the other method. It is difficult to see why one method should acquire more accurate knowledge than the other.

Several hypotheses have therefore been stated about the effect of method on knowledge acquisition. For some of these hypotheses, an attempt has been made to describe the mental processes that cognitive theory and introspection would suggest are taking place. If the empirical data support these hypotheses, these expectations will continue to appear justified. But if the data contradict these hypotheses, it might be possible to make statements about what mental processes appear not to be occurring. In addition, the empirical data is likely to uncover reasons to use one knowledge acquisition method rather than the other or to be indifferent between them. The data could also be useful to those interested in developing new methods.

Chapter 4. Evaluation Measures

Abstract

Any knowledge can be described in terms of its characteristics (e.g., its accuracy). Comparing two knowledge bases involves contrasting their descriptions. For example, one knowledge base may be said to be more accurate, or larger than another. These characteristics can be influenced by factors present while the knowledge base is being conceived or constructed. This chapter discusses these characteristics, how they can be measured, and the factors that can affect them.

Introduction

People who develop knowledge bases tend to evaluate their performance as well; that is a familiar problem (see, e.g., Weiss & Kulikowski, 1984; Ginsberg, 1988; Saito & Nakano, 1988). Typically they do what Cohen & Howe (1988) refer to as a "comparison study". But there are two kinds of reasons that people have for evaluating knowledge-based systems: certification and research. A growing number of researchers evaluate groups of related knowledge bases, in search of the influence of knowledge-acquisition method and other factors (see, e.g., Michalski & Chilausky, 1980; Burton & Shadbolt, 1988; Adelman, 1989). To facilitate such evaluations and comparisons, some writers have called for standard problems and measures (see, e.g., Cohen & Howe, 1988 and 1989; Adelman, 1989; Hayes-Roth, 1989). The machine-learning community already uses such problems, e.g, Fisher's (1936) flower learning set or the faulty-calculator data of Breiman, Friedman, Olshen, & Stone (1984), to evaluate the efficiency and efficacy of different algorithms.

This chapter discusses key characteristics of knowledge bases, and examines the measures used by some of those who have, for whatever reason, evaluated knowledge bases. The chapter argues that there is a need to compare

knowledge bases with each other (e.g., knowledge of a single domain elicited with two methods, knowledge of two domains elicited with the same method, or knowledge of one domain elicited from different sources). These comparisons will help to develop a shared understanding of the issues involved in evaluation of knowledge-based systems. They will also help provide the foundations upon which hypotheses may be tested and theories may evolve.

The chapter also examines the factors, present in system-building activities, that might influence the characteristics of knowledge bases.

Certification for Operation

Evaluation (sometimes called verification or validation, e.g., by Buntine & Stirling, 1988; or quality analysis, e.g., by Collins, Ghosh, & Scofield, 1988) is often a process of testing, faulting and refinement that attempts to transform a knowledge-based system into usable software that performs at a level of competence comparable to that of a human specialist in the domain.

Liebowitz (1986) distinguishes between validation ("whether the correct problem was solved") and evaluation ("the software's accuracy and usefulness"). Vinze (1992, p 312) defines verification as "a method of evaluating the effectiveness of" a knowledge-based system. Lydiard (1992, p 102), who distinguishes between verification ("are we building the product right") and validation ("are we building the right product"), sees evaluation as "a feature of both verification and validation".

Evaluation is generally a prerequisite for placing a system into routine operation. Indeed, Gaschnig, Klahr, Pople, Shortliffe & Terry (1983) refer to evaluation as "certification for operation". Two kinds of testing are necessary: the first is to assess the system's expertness or competence; the second is for usability.

Firstly, the builder of a knowledge-based system evaluates it because s/he is interested in seeing what it does right and what it does incorrectly. It is here that the program's performance is measured and compared with that of some standard. S/he is also interested in making corrections to the knowledge base so as to increase the instances in which the system acts correctly and to reduce those in which it is wrong (see, e.g., Ginsberg, 1988). The overall aim is usually to have the system demonstrating a level of competence that both domain expert and user alike will find at best impressive and at worst adequate.

But mere "expertness" may not be enough to satisfy potential users of a knowledge-based system. So evaluation often also aims at finding out whether the users are comfortable with the system's style of interaction and speed of response, and whether the advice given by the system is useful to them in the functions that they have to perform (see also Waterman, 1985, p 199).

Evaluation for Research

A growing number of researchers are also evaluating knowledge bases to investigate, for example, the effect of knowledge-acquisition method on the development effort required (see, e.g., Michalski & Chilausky, 1980). Increasingly, researchers (e.g., Adelman, 1989; Cohen & Howe, 1988 and 1989; and Hayes-Roth, 1989) have also been recognising the need to compare knowledge-based systems as a part of the process of developing theory and testing hypotheses. The simplest comparison would be between two knowledge bases developed separately for the same domain. As discussed in "Efficacy of Method" on page 81, the accuracy and size of the two knowledge bases might be important in such a comparison.

Not long ago, Hayes-Roth (1989) argued that knowledge engineering was ripe for transformation from a largely practical discipline into one where practice could begin to be viewed within the framework of coherent theory.

But, according to Hayes-Roth (1989, p 101), "the field of knowledge engineering ... lacks meaningful measures of progress". He argued that such measures would enable the

> design of knowledge system experiments that would address typical categories of knowledge and key knowledge engineering costs and performance parameters.

It is not clear whether a measure of progress is different from a measure of knowledge-base quality. Nor is it clear whether he is asking for a vision of the future, and progress reports on the current state of the journey towards that vision. Perhaps both are interconnected, as in the progress from analysis to synthesis systems, or improvements in the ratio of knowledge-base size or accuracy to knowledge-acquisition effort.

Evaluation: What is Involved

Cohen & Howe describe the actions usually performed in evaluating a single knowledge-based system. According to Cohen & Howe (1988, p 40),

> in the basic form of a comparison study, we select one or more measures of a program's performance; then, both the program and a standard solve a set of problems; and, finally, the solutions are compared on the measures.

This section discusses the standards used for comparison, the characteristics (e.g., performance) that are evaluated, and the measures that have been used.

Standards of Comparison

When a knowledge-based system is built, its performance is usually compared with that of an appropriate standard. But writers (e.g., Stevens, 1984; Hayes-Roth, 1989; Forsythe & Buchanan, 1989) disagree about which standard should be used. The standards recommended include groups with known levels of expertise, an independent expert, the knowledge source, and normative theories.

Evaluation Panels

Cohen & Howe (1988, p 40) argue that the standard against which a knowledge-based system is compared can be a group consisting mainly of domain experts, but with a few novices included as "an interesting control condition to ensure that successful performance requires expertise". This view is shared, at least partly, by Bratko & Kononenko (1989), who compared the performance of some of their medical diagnostic systems to that of specialist as well as non-specialist medical doctors. These approaches make it possible to locate the performance of the system along a spectrum of expertise ranging from novice, through semi-expert, to expert. As discussed in Chapter 3, "Some Implications of Cognitive Psychology for Knowledge Acquisition," expert performance is generally distinguished from that of novices by accuracy, the number of steps in the reasoning process, and speed of decision (Anderson, 1982).

If the group contains multiple experts, there may be differences of opinion among them about what is correct, which can both enrich and complicate the evaluation. It is often useful to seek consensus where a panel of experts is used. Kors, Settig, & van Bemmel (1990) have reported satisfactory resilts from using the Delphi method. However achieved, this consensus becomes the standard against which the knowledge-based system's performance is judged (see e.g., Collins, Ghosh, & Scofield, 1988).

Evaluation by an Independent Expert

On the other hand, if a "gold standard" is used (see, e.g., Burton & Shadbolt, 1988, p 11), then the performance of both the knowledge source and the expert system may be compared with that of the standard. In such circumstances, it is possible for both the artifact and the source to perform less than perfectly. Adelman (1989) also used a gold standard, so did Berwick (1985). However, there is often little justification for taking as gospel the views of a single expert.

Where a gold standard is used, its utility in comparing two knowledge bases with each other is that it allows each to be measured in turn against a common standard. This might be appropriate where knowledge for a single domain is acquired, from different sources or by different methods, to build two knowledge bases.

But the standard clearly cannot be universal. It is, in general, not transferable between systems. For example, the standard against which a rheumatology diagnosis system is compared is quite different from one for a mortgage underwriting advisor. Less obvious is the fact that the knowledge units in the knowledge base may not correspond to the ones obtained from the gold standard. This is especially likely, according to Cohen & Howe (1988, p 40), "when test problems have so many acceptable solutions that a program and a standard cannot be expected to generate the same ones".

Evaluation by the Knowledge Source

Of course, the performance of the system can simply be compared to that of the source from which the knowledge was acquired. This approach was used by Yih (1988), who compared "the performance of the extracted rules ... with the performance of the 'expert schedulers', from whom the rules were extracted".

When the performance of a knowledge base is compared with that of its knowledge source, what is being measured is how well the acquisition process has created a model of the source's expertise. This is so as long as the test

cases fall within the limits of the domain, as a knowledge source (particularly a human one) is not necessarily bounded in its domain of expertise in the same way as a knowledge-based system is.

Evaluation against Normative Theories

In some problems, it is possible to use what Cohen & Howe (1988) describe as "objective, recognized standards [e.g.,] normative theories". For instance, Yih (1988) compares

> the performance of the optimal policy ... with that of the resulting rules from trace-driven knowledge acquisition .... The results show that the rules extracted from trace-driven knowledge acquisition ... yield near optimal performance.

Clearly, not every domain is susceptible to this kind of analysis, but wherever problems can be selected that have model answers, these answers will provide a scoring key against which to assess the system's performance. The question arises, however, as to the reason for using a knowledge-based approach for a problem that has a clear algorithmic solution.

Test Cases or Expert Assessment

Some builders of knowledge bases have found it convenient to compile a set of test cases along with their solutions from the expert and to store these in a case database. Such a database can be used for testing a knowledge base via a batch process (that is, where the knowledge-based system is equipped to deal with input in that form; see, e.g., Roskar, 1988). This is an approach commonly used for inductively generated knowledge bases, but it also allows evaluation to proceed rapidly even where induction is not used. It also has the advantage that it can easily be an iterative process of evaluation and refinement until the system performs as well as the knowledge sources.

To be usable in this way, test cases must be expressed in terms of the same domain model as the knowledge base to be evaluated. Or the latter must be a subset of the former. For instance, if the knowledge base includes a test of colour, the test cases must also be described in terms of colour.

If a knowledge base exists without compatible test cases, exemplars can be generated for solution by the expert (Lundell, 1988). They can then be used to evaluate the knowledge base. Refinements to the knowledge base may also demand changes to the test cases.

It is possible, as Cohen & Howe (1988) have pointed out, that each test case could have several correct solutions. To cater for such situations, the representation used for each case should be flexible enough to accommodate several classes. If this flexibility is not available, as Cohen & Howe (1988, p 40) put it, "one cannot compare the program's performance with the expert's but must instead rely on the expert's direct assessment of the program". This is how Muggleton (1986) evaluated his chess end-game expert system, i.e., by having a human expert assess the program's play.

But Cohen & Howe have pointed out the tendency of experts to be "overly generous to the program. Moreover, direct assessment does not tell us whether the program is performing better than the expert." Only an independent expert can do this.

In systems for synthesis, evaluation by expert assessment is often the most convenient method. Indeed, it is sometimes necessary (see, e.g., Birmingham, 1988) to go beyond the expert's assessment and to build the object designed by the knowledge-based system to see whether it works. This is costly and can usually be done only for a small sample of cases. According to Birmingham (1988), the knowledge captured by CGEN "was sufficient to build an interesting set of designs, one of which was constructed and shown to work".

Nicholson (1988) also contains the following account given by a knowledge engineer regarding the validation strategy for an expert design system built by his organisation:

> When [the knowledge-based system] comes out with a configuration, the first level of validation was: 'Is it a system that would be acceptable to the experts?' The second level of validation was: 'Can we build it; and does it work when we build it?' So your only real validation is experience ... that's only a way of proving it's wrong, but never a way of proving it's right.

Evaluation is not always easy. Davis & Lenat (1982, p 123) assert that

> judging the performance of a system like AM ... is a very hard task, since AM has no 'goal'. Even using current mathematical standards, should AM be judged on what it produced, or the quality of the path which led to those results, or the difference between what it started with and what it finally derived?

But AM is a special case: it is a program that discovers interesting mathematical relationships by a kind of introspection.

Characteristics of Knowledge Bases

Knowledge bases have several characteristics; some appear to be only describable while others are also measurable. Among the characteristics that people have found interesting are performance, usability, cost, size, and intelligibility. These are discussed in the rest of this section.

Performance

The term "performance" has already been used in discussing standards. But, like several other writers, this author has done so without explaining or defining

it. When you evaluate the performance[3] of a knowledge base, you are likely to be interested in being able to say that the conclusion of the system coincides with that of the standard in some percentage of cases.

Builders of knowledge-based systems often invest considerable effort in satisfying themselves about their systems' performance and in demonstrating to prospective users that the systems are likely to perform well. It is often said, for instance, that DENDRAL performs better than domain experts (see, e.g., Weizenbaum, 1976, p 230), although it is more usual to find that knowledge-based programs perform somewhat less impressively.

Different writers have used their own expressions in referring to performance:

- "Performance" (e.g., Davis & Lenat, 1982)
- "Number of times correct" (Stevens, 1984)
- "Effectiveness of rules" (Buchanan *et al*, 1983)
- "% correct diagnosis" (Michalski & Chilausky, 1980)
- "True positives" (Politakis, 1985; Weiss & Kulikowski, 1984; and Ginsberg, 1988)
- "Hit rate" (Cohen & Howe, 1988 and 1989)
- "Appropriateness of choices" (Hayes-Roth, 1989)
- "Diagnostic accuracy" (Bratko & Kononenko, 1989).

As mentioned in "Efficacy of Method" on page 81, the computation of this quantity appears to be widely agreed. When N cases with known solutions are presented to the knowledge-based system, it produces M (less than or equal

---

[3] Lydiard (1992, p 102) equates knowledge-base performance with efficiency of execution.

to N) correct solutions. The diagnostic accuracy (or whatever you wish to call it) is said to be M/N, often expressed as a percentage.

There are difficulties with this measure in domains where several correct solutions to a given case may exist. The system's solution may be correct but different from that preferred by the standard. Michalski & Chilausky (1980) used a supplementary measure ("% preferred diagnosis") to indicate the number of cases in which their expert system was not merely correct, but had produced the solution preferred by the standard.

Moreover, such a bald statistic as diagnostic accuracy is likely to suffer from the same weakness as any measure of central tendency: it says nothing about the number of cases used to evaluate the system. Stevens (1984) expresses a concern for the statistical significance of apparent differences in performance:

> if a modification to a system means that the number of times it is correct increases from say 60% to 65% how can we know whether this increase is significant? ... How can we exclude the possibility that an improvement is the result of chance?

Perhaps the statistical F-test based on an analysis of variance for the data is one way, in properly designed experiments, to alleviate such concerns (see, e.g., Chapanis, 1959).

Equally important, diagnostic accuracy does not indicate how representative these test cases are of the universe of cases that the system will be required to handle over its lifetime (Gaschnig *et al*, 1983). This concern is shared by Lehner (1989, p 658), who recommends "the use of a representative set of randomly selected test problems".

Some researchers have also supplemented the diagnostic accuracy measure with a measure of the number of times that the system fails to produce

a correct solution. This has been called "% false positives" (see, e.g., Politakis, 1985; Weiss & Kulikowski, 1984) and "% not diagnosed" (see Michalski & Chilausky, 1980). The less restricting latter measure is not strictly equivalent to the former. They are subtly different because "% not diagnosed" allows for the counting of those cases in which a knowledge-based system, unblinkered by the closed-world assumption (see, e.g., Steels & Campbell, 1985), fails to offer a solution.

The percentage of false positives is used in combination with % false negatives to describe the performance of systems that make binary-valued decisions, e.g., whether a thermostat is normal or faulty. In such systems, the measures are related as follows:

diagnostic accuracy + % false positives + % false negatives +
% not diagnosed = 100%.

In situations where assessment by the expert is necessary, diagnostic accuracy and the other performance measures may be difficult to interpret.

In contrast with the situation where two knowledge bases for the same domain are compared, it is also useful to be able to compare two knowledge bases for different domains. One dimension on which they can be compared is how well they achieve their intended objectives. Buchanan & Feigenbaum (1982) refer to this as "power ... to perform its task well". Good measures, even qualitative ones, of how well an expert system is achieving its objectives would facilitate such comparisons. However, the measures discussed above appear adequate. If two knowledge bases have diagnostic accuracy of 80%, it is safe to conclude that they have both modelled their source's expertise imperfectly to the same extent, assuming the knowledge source is used as the standard in both instances, and the test cases are obtained by the same method.

Performance is probably the most important attribute of a knowledge base, because if the diagnostic accuracy is not acceptable, then other features of

the knowledge base are hardly likely to persuade anyone to put the system into routine operation.

Usability

Rasmussen (1980) has argued that software users appear to have internal models (stored in long-term memory, and accessed subconsciously) of familiar situations. These models respond to perceptual inputs and operate like fast analogue computers. But in strange situations, outside the applicability of their stored models, users reason at a conscious level. This conscious reasoning uses cognitive processes that are slow and have low capacity. It is therefore important that interactive computer systems harmonise appropriately with these two distinct human information processing modes.

So some builders of knowledge-based systems also subject their systems to some form of testing by users, mainly to assess how easy the systems are to use, how appropriate their user interfaces are, and overall how useful thee systems are. The importance of this characteristic is clear to Vinze (1992), who treats usability and effectiveness of a knowledge-based system as synonymous, measuring them by using a questionnaire that focuses on users' satisfaction with the system.

Measuring usability may also involve a pilot operation or the release of a *beta* version to selected users (see Nicholson, 1988). Problems found here may demand restructuring of the knowledge base to improve response time, rewording of the system's questions to eliminate ambiguity, or even a change of the delivery platform. Where large knowledge bases are integrated into database systems, this is especially vital (see, e.g., Mumford & MacDonald, 1989, pp 116-123).

One of the dimensions proposed by Davis & Lenat (1982) for evaluating their AM system is "the character of the user-system interactions". But while some aspects of usability (e.g., the quality of the questions asked) may be

characteristics of the knowledge base itself, others may be characteristics of the user interface or the inference engine.

Cost

The cost of developing a system is a part of the history of that system. It may not be readily observable after the system has been built, but it can be especially useful if known before the system is developed. Both Gaschnig *et al* (1983, p 247) and Hayes-Roth (1989, p 109) acknowledge the importance of this characteristic, for which Hayes-Roth proposes "cost per knowledge unit" as a measure.

After the system has been built, it carries with it a further intrinsic cost, which, according to Gaschnig *et al* (1983, p 247), includes the expense of "providing for its maintenance". Hayes-Roth (1989, p 109) proposes "costs and effects of knowledge incrementation", and "ease of knowledge deletion" as measures of this characteristic. Gaschnig *et al* (1983, p 247) argue that the cost of "running it, interpreting the results" is important.

But Buchanan *et al* (1983, p 158) attempt to put costs in perspective by distinguishing between computing costs and human costs:

> the cost of computer resources is falling rapidly with recent progress in hardware technology, while cost of human labor (especially of knowledge engineers) is increasing. Thus the benefit of reducing human resource costs far outweighs the cost in computer resources.

Cost is, of course, a relative concept: it might be sensible to incur great costs if even greater returns are expected to result. Connell (1987) argues that, whereas a proof-of-concept system may be appropriately judged by whether it works, an operational system can be subject to more stringent measures, such as return on investment.

Size of Knowledge Base

Michalski & Chilausky (1980) describe the size of their soybean disease diagnosis system in terms of the number of findings or attributes, the number of conclusions or classes, and the number of rules. Both Politakis (1985) and Ginsberg (1988) also describe the size of their rheumatology diagnosis system in these terms. These measures (especially number of rules) are to be used with caution because it is well known that the inclination of the knowledge engineer and the restrictions of the formalism can both affect the number of rules in a knowledge base.

It is not necessarily true, for example, that a knowledge base with 1000 rules is smaller than one with 2000 rules. However, the three quantities of Michalski & Chilausky taken together give a good indication of relative sizes, if the tool being used is also stated.

Another source of difficulty is that the size of knowledge bases represented as frames or semantic networks is difficult to compare meaningfully with that of those represented mainly as rules. Indeed, a system cam perform the same task with either a rule-intensive or frame-intensive representation. So some people reckon size by the number of bytes of storage occupied by the knowledge base. Even that has its pitfalls, as verbose people may use long character strings to express classes and attributes.

Intelligibility

Some writers insist that rules must make sense to the human expert. Politakis (1985), for example, argues that this is vital in medical diagnosis systems, to help the experts feel confident about the soundness of the systems' underlying knowledge (see also Spackman, 1991). Other writers (e.g., Hart, 1986) point out that certain induction algorithms sometimes yield rules which, although diagnostically accurate, are unintelligible.

Szolovits (1986) has pointed out that although rules are more commonly used to represent knowledge in expert systems, Lisp functions are sometimes used. They achieve "efficiency and ease of initial construction at the expense of making its knowledge inaccessible in an explicit form". Hayes-Roth (1989) proposes "accessibility of knowledge via queries" and "understandability of results" as measures of intelligibility, while Sykes (1987) mentions "ability to explain its reasoning".

But intelligibility may sometimes be obtained at the expense of efficient operation, or even of tractability (Michie, 1982).

Even so, it may be enlightening to evaluate expert systems on the quality of their reasoning, not just the accuracy of their results.

Other Characteristics of Knowledge Bases

There are other dimensions beside those discussed above. Some knowledge-based systems in current use are valued not only because they can arrive at the same decision as an expert almost every time, but equally because they are able to do so in a fraction of the time that the domain expert traditionally took (see e.g., Nicholson, 1988). By using such a system, the expert can perform more efficiently.

Sykes (1987) mentions brittleness (Cohen & Howe, 1988, p 40 refer to the opposite pole of this as "robustness"), and depth of reasoning as important attributes of a knowledge base. He sees shallow reasoning being characteristic of knowledge bases that model experts' heuristics, while deep reasoning is characteristic of systems that model "the underlying physical mechanism" of the domain. (The trend towards using the latter kind of architecture is mentioned on page 6.)

Possible Influences

A number of factors appear to be capable of influencing the outcome of a knowledge acquisition project; five of these are listed by Adelman (1989): the knowledge-acquisition method, the knowledge representation formalism, the domain and the type of knowledge that characterises it, the knowledge source, and the knowledge engineer.

To be able to analyse sensibly what is going on, it is necessary to develop measures that can characterise each of these factors. With such measures, it will begin to be possible to discover causal links or correlations between the knowledge-base characteristics discussed in "Characteristics of Knowledge Bases" on page 93 and other variables that may affect them.

This section examines some of the factors that writers have, whether from experiment or experience, seen as influential.

Internal Influences

It is hardly fanciful to expect that some characteristics of a knowledge base will be strongly correlated with others. For example, intelligibility may be influenced by representation (see, e.g., Szolovits, 1986), and both intelligibility and size of a knowledge base may affect the cost of maintenance. Size may also affect speed, especially in backward-chaining systems (see, e.g., Davis & Lenat, 1982, pp 409-410). But while speed may degrade with increases in knowledge-base size, speed can sometimes be increased by restructuring a large knowledge base into a hierarchy of smaller ones.

Another assertion about the possible effect of knowledge-base characteristics on each other was made by Feigenbaum & Buchanan (1982). According to them,

> as with all such insights, it appears in retrospect to be common sense: that the power of an intelligent program to perform its task well depends

primarily on the quantity and quality of knowledge it has about the task.

But researchers are not likely to find out how well, how much, and what kind without effective measures of important characteristics of knowledge-based systems. And without such measures, researchers are unlikely to be able to transform this conjecture of Buchanan & Feigenbaum into more precise forms such as those depicted in Figure 17 and Figure 18 on page 103.

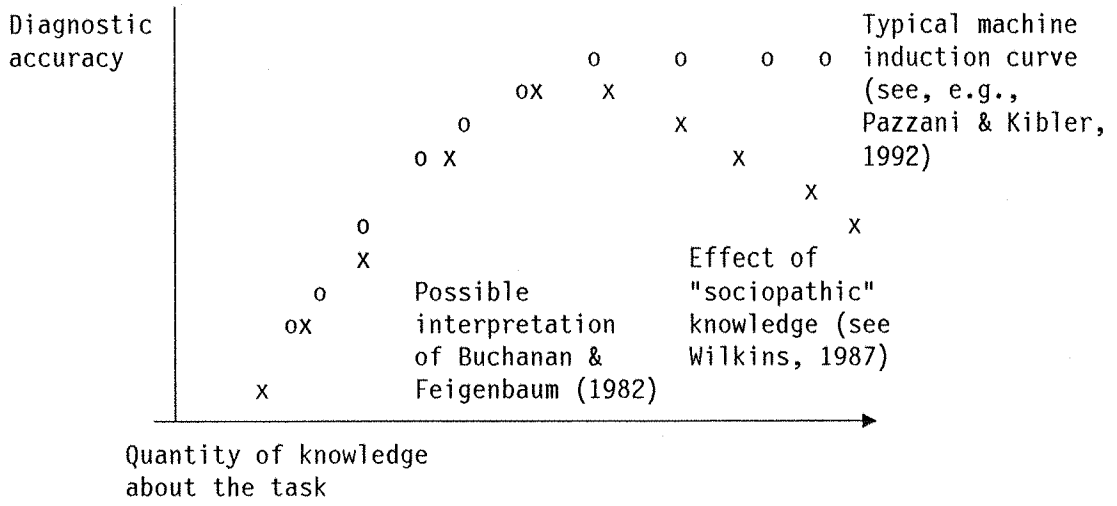

Figure 17. Possible relationship between knowledge quantity and KB performance

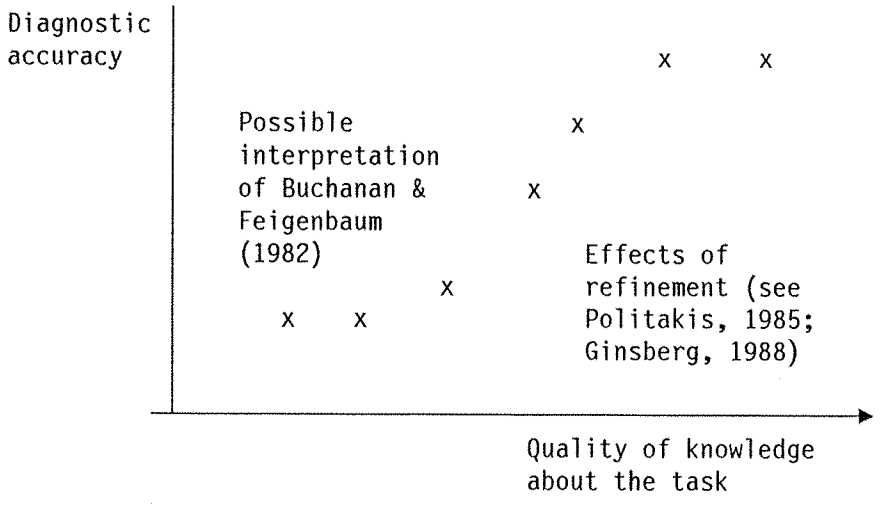

Figure 18. Possible relationship between knowledge quality and KB performance

Influence of Method

Quite apart from the effect that knowledge-base attributes may have on each other, it is clear that factors external to the knowledge base also influence knowledge-base characteristics. Arguments have clearly been forwarded in Chapter 3, "Some Implications of Cognitive Psychology for Knowledge Acquisition" about how the repertory grid technique and KAMSE might affect the knowledge they are used to acquire. More generally, empirical work has been done to demonstrate that these effects do exist. For instance, Michalski & Chilausky (1980) showed that machine induction created a knowledge base with higher diagnostic accuracy than "hand-crafted" rules.

But Michalski & Chilausky appear to have failed to determine how much better machine induction was, because they set the inference threshold for the inductively generated knowledge base at a higher level than that for the hand-crafted system. Perhaps a richer experiment would have been to determine the effect of threshold on diagnostic accuracy (Levi, 1989, also discusses this connection) for the two methods. Had that been done, Michalski & Chilausky would have been able to establish (subject admittedly to grave

doubts about statistical significance) the sort of relationship depicted in Figure 19 on page 104. But there really is only a single point on each of those two curves.

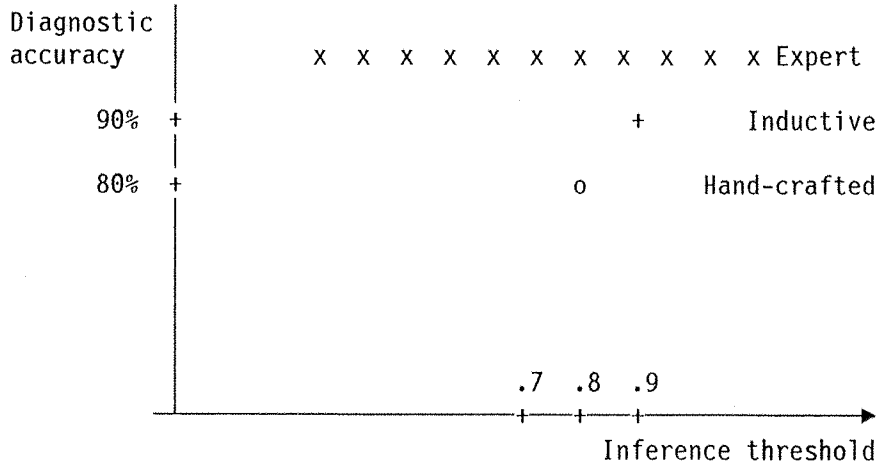

Figure 19. A possible context for the two observations of Michalski & Chilausky

With a different experiment design, Michalski & Chilausky's findings would have enabled the knowledge-engineering community to say more than Buchanan et al (1983) did in their assertion that

> AQ11 has been used to formulate factual knowledge in the form of rules for diagnosing plant diseases. These rules have proved more effective than those generated by an expert (although the expert could still analyse test cases more effectively than the program using its automatically formulated rules).

In addition, Michalski & Chilausky's failure to describe their "hand-crafted" method more precisely than "conference with the experts" deprives others of further valuable insights.

The other major finding of Michalski & Chilausky was that the effort required to create an accurate knowledge base is much less when machine

induction is used than when rules are hand-crafted. However, Michalski & Chilausky employ a crude measure of knowledge-acquisition effort: "approximately 20 hours were required to [develop] the descriptions for the above 15 diseases" (p 70). And other researchers are left to assume, for example, the number of people and thence the effort in person-hours.

Researchers at Nottingham University, notably Mike Burton and Nigel Shadbolt, have over several years done a series of experiments to discover relationships between the knowledge-acquisition method used and "the time and effort taken ..., the amount of information generated ..., and the amount of genuine, usable knowledge generated" (Burton & Shadbolt, 1988, p 11).

Knowledge-acquisition method is also linked with effort by Muggleton (1986, p 67), who asserts that whereas

> designers using dialogue acquisition methodologies talk of constructing prototype systems in terms of years, MUGOL-based applications have been consistently prototyped in around six person months.

This comparison of his, however, lacks precision as it compares elapsed time (years) with effort (person months). Yet it still indicates that both are possible measures of the cost of building a knowledge-based system.

But how can knowledge acquisition methods be characterised, placed along a continuum, in a plane or within a space? This is a difficult endeavour, as is evident from Chapter 1. On one dimension they can be said to be either interactive or inductive (Muggleton, 1986). But that is not enough. Other writers have used other dimensions: Politakis (1985) classifies knowledge-acquisition methods as direct and indirect. According to him,

> a direct approach has tried to find efficient techniques to extract knowledge from the domain expert, while other research efforts have sought ways of acquiring expert knowledge indirectly from sample cases.

Clearly, his direct / indirect construct is equivalent to Muggleton's interactive / inductive.

Influence of Domain

It would also be useful to have dimensions that characterise a domain adequately. Some writers (e.g., Partridge, 1986) have classified domains according to their degree of structure, with one extreme being highly structured and its opposite pole being ill structured. Some (e.g., Boose, 1989) have classed them as either analytic or for synthesis. And some (e.g., Muggleton, 1986) have categorised them as either evaluative or controlling. In addition, some kinds of expertise appear to be stored in different ways in people's minds. For example, knowledge of how to find one's way around the campus might be stored differently[4] from knowledge about correct spelling. Some writers (e.g., Burton & Shadbolt, 1988) have also mentioned as well that a domain might be highly procedural or it might be classificatory.

Politakis (1985, p 33) describes the kind of knowledge domain for which learning from examples is unlikely to work well:

> realistic and large-scale medical diagnostic applications where the dimensionality is large both in numbers of findings and conclusions and there is much uncertainty in diagnostic reasoning. In these applications,

---

[4] Neuropsychological evidence of selective impairment of some types of knowledge (e.g., face recognition; see Ellis & Young, 1988) supports this.

a set of cases rarely covers the range of possible ways of arriving at diagnostic conclusions which an expert physician has experienced.

Influence of People

A domain expert can perhaps be characterised according to how much expertise he or she has. Novice and expert would be represented by points on this scale. A domain expert is simply a person, and could therefore be classified like any other person, e.g, by psychometric testing. In a similar way, the knowledge engineer might be tested. Some writers (e.g., Forsythe & Buchanan, 1989) have also mentioned power differences between the domain expert and the knowledge engineer as being potentially significant. Perhaps some way could be found of characterising the power of each one. Some writers have also mentioned willingness, articulateness, and being busy as possible dimensions (see, e.g., Bell, 1985).

Adelman (1989) used knowledge engineer (he had six of them participating in his experiment) as an independent variable, but failed to propose any meaningful dimensions on which these people could be characterised or compared. The dimension he settled for was crude: he grouped them by the institution in which they were trained.

Other Influences

Anjewierden (1987) briefly evaluates five knowledge-acquisition tools (ROGET, KEATS, AQUINAS, ETS, and KREME) based on the "level of representation" (see Brachman, 1987) that they support.

Several writers (e.g., Rasmussen, 1980; Arblaster, 1983; Waterman, 1985) have indicated that the user interface is an important factor in the success of their systems. If they are correct, a CAKE tool's user interface may affect the knowledge that it can acquire. The user interface also affects how easy a tool is

to learn to use (Boose, 1989), and how much information the user has to enter (Klinker *et al*, 1987).

Muggleton (1986) uses the concept of inductive efficiency to measure the effort that an algorithm expends in transforming examples into formal knowledge.

Conclusions

People evaluate knowledge bases to certify them for operation or to discover interesting relationships through experimentation. Several characteristics of knowledge bases can be measured, but the most commonly used is diagnostic accuracy as an indication of performance. Few agreed quantitative measures exist for describing the other attributes of knowledge bases.

Several measurable factors can influence the characteristics of a knowledge base. With appropriate dimensions for each factor, and suitable measures of knowledge-base characteristics, it is possible to investigate relationships empirically, through well designed experiments. For not only must knowledge-based systems be evaluated, but also the methods and tools used to build them. It is evident that metrics are needed, although those adopted for conventional software development (see, e.g., Grady & Caswell, 1987) appear inappropriate for knowledge bases.

There is also a sense in which evaluation is a continuing process, and that even in routine operation a system's performance is under constant scrutiny. Perhaps evaluation over time, or some time-changing factor, would be appropriate (see, e.g., Cohen & Howe, 1989).

Chapter 5. The Controlled Experiment in Knowledge-Acquisition Research

Abstract

This chapter[5] is based on a review of the literature about controlled experiments in research on knowledge acquisition. The review was carried out to help the author make decisions about the design of his own experiment comparing two knowledge-acquisition methods. The chapter looks critically at six experiments reported in the literature, and proposes a framework within which such empirical work can be viewed. It concludes that some of the apparent difficulties can be resolved, and that controlled experiments can be a useful way of discovering the relationships at work in a knowledge-acquisition project.

Introduction

Case studies and benchmarks have been used widely in research on knowledge-based systems. For example, in a case study, Michalski & Chilausky (1980) investigated the effect of the acquisition method in a single domain on the effort needed to acquire the knowledge, and on the diagnostic accuracy of the resulting knowledge bases. In a benchmark, Quinlan (1986) used several case bases as input to different induction algorithms, and observed the effect of these variables on the diagnostic accuracy of the induced knowledge bases.

But there appears to be a growing awareness (Adelman, 1989) that controlled experiments can help advance understanding of how the knowledge source, representation, acquisition method, domain, and engineer affect the effort needed to build a knowledge base and the quality of its performance.

---

[5] A paper based on this chapter is being published as Nicholson (1992b).

Burton & Shadbolt (1988, p 11) argue strongly in favour of controlled experimentation:

> Although one can get useful practical information from case studies, there will always be many factors unique to any particular knowledge elicitation session. Hence the need for a formal experimental analysis.

Indeed, researchers such as Burton, Shadbolt, Hedgecock & Rugg (1987), Lundell (1988), Stevenson, Manktelow, & Howard (1988), Deffner & Ahrens (1989), Adelman (1989), and Agarwal & Tanniru (1990) have used methods from experimental psychology to explore research questions in knowledge acquisition.

The author's interest in the subject arose from his own need to compare two knowledge-acquisition methods in terms of the effort they demand from a domain expert, and the accuracy of their outcomes. The controlled experiment seemed the ideal way to do the investigation, so a search was made of previous uses of this approach in the field of knowledge acquisition. It is evident that not many researchers have used controlled experiments for this purpose. However, the few that appear in the literature do contain lessons from which the author's own design was able to benefit. These lessons, and their influence on the author's design, are discussed in this chapter.

Experiments

This section discusses six experiments reported in the knowledge-acquisition literature.

Congruence of Representation

Proposing hypotheses based on Anderson's (1982) theory of skill acquisition, Lundell (1988) argues that while novices store their expertise in declarative memory, or at the conscious level, experts do so in procedural memory, or at the tacit level. Lundell further argues that it ought to be easier to elicit rules from novices than from experts, and that it ought to be easier to obtain typical examples (or what he calls "prototypes") from experts than from novices.

In addition, Lundell conjectures that an artificial neural network (built using prototypes and exemplars obtained from an expert) ought to have greater diagnostic accuracy than a similar knowledge base derived from exemplars and prototypes that have been elicited from novices. Conversely, a set of rules elicited directly from a novice ought to have a higher diagnostic accuracy than a set elicited directly from an expert.

Lundell's "representational congruence" hypothesis asserts that if, for example, a rule elicitation method is used, then it will elicit primarily knowledge stored as rules in the mind of the expert. Lundell's representational and "elicitational congruence" hypotheses involve the following independent variables:

- Elicitation method
- Expert's level of expertise
- Knowledge representation in the knowledge base.

These variables are all controllable in an experiment. The dependent variable, which, Lundell argues, is a function of the variables listed above, is the diagnostic accuracy of the knowledge base built using the knowledge elicited from a subject. To test his hypotheses, Lundell had to vary the controllable variables in turn, and record the effects on diagnostic accuracy. Taking several observations for each setting of each controllable variable allowed him to

increase the reliability of his results. Of course, the subjects themselves are also variable (see, e.g., Chapanis, 1959, and Adelman, 1989).

Lundell's experiment is essentially a two-group design, in which each subject fills in four different types of questionnaire. It used a random presentation order in an attempt at eliminating sequence effects.

Two of Lundell's questionnaires were aimed at eliciting rules directly. One he called the "direct rule" questionnaire, and the other the "decomposed rule" questionnaire. These two complemented each other in his subsequent creation of rule bases.

The two other questionnaires were aimed at eliciting examples, from which knowledge could be derived by some kind of machine learning. One of these questionnaires elicited a set of typical examples or cases; this one he called the "prototype elicitation" questionnaire. The fourth questionnaire, which he called the "exemplar questionnaire", consisted of a randomly generated set of undiagnosed hypothetical cases for the subjects to diagnose.

Using these questionnaires for knowledge acquisition appears to impair the external validity of Lundell's experiment. The antecedents and the consequents are given, whereas in practice it seems more usual for these to have to be elicited from the knowledge source by various methods. The considerable amount of knowledge acquisition which clearly went into the preparation of these questionnaires deserves to be acknowledged openly. Moreover, questionnaires are rarely used to acquire knowledge for knowledge-based systems (see, e.g., Welbank, 1990).

Lundell used the completed questionnaires to build a number of expert systems, but little is said, in his dissertation, about this process. And without any assurance to the contrary, his readers are left wondering about the scope for introduction of errors at this stage. Still, perhaps this criticism is a bit unfair, because the graphical representation on his questionnaires seems capable

of being easily transformed into production rules. In the case of his connectionist networks, it appears obvious that the exemplar and prototype data were simply coded as examples and used to train the networks in the diagnostic task.

Lundell's subjects emerged from his training with a range of levels of expertise in the diagnostic task. Some had become good at it, and others had learned to a lesser extent. Lundell classified his newly trained subjects as either skilled or unskilled. He set his criterion at the median test score, so that half the subjects were "unskilled" and the others "skilled". It appears to be an arbitrary distinction with little basis in theory and little rationale, save that of balancing the sizes of the two groups.

After basing his initial arguments on the theory that experts' skills reside at a tacit level while novices' skills are represented consciously, Lundell appears to make little use of this representational differential that would be expected to exist between his skilled group and his unskilled one.

Perhaps an improvement would have been to use an adaptive questionnaire to gather the same type of data. Under this approach, subjects would interact with a computer program that asks questions based on answers already given. By doing this, he would have introduced some of the flexibility characteristic of real-world knowledge acquisition, while providing systematic and consistent recording of data.

By creating his own experts in a domain of his own making, Lundell may have sacrificed external validity, but at the same time he gained a ready-made set of test cases against which both the experts themselves and the elicited knowledge bases could be evaluated. He also limited the scope of the task to a size amenable to analysis and experimental control.

Thinking Aloud

Stevenson *et al* (1988) also did an experiment to test a hypothesis implied by Anderson's (1982) ACT* theory. Their hypothesis was that their own method of knowledge acquisition would be more effective than "traditional" methods. They argue that it is wrong to assume that analysis of thinking-aloud protocols accurately unearths the knowledge contained in an expert's automatic productions. What thinking aloud is more likely to do, they argue, is to slow down and even distort the expert's actions. They argue that it is more effective to let the expert perform his task undisturbed except for the scrutiny of a videotape camera and recorder. At some later time, the expert can explain his actions while watching the videotape. These explanations can be used to generate production rules. Stevenson *et al* call this method an "evaluation technique".

The experiment of Stevenson *et al* tested their hypothesis by varying the acquisition-method treatments to which subjects were assigned. They used a two-group repeated-measures design, although one group (the experts) was very small (two subjects) compared with the other group (eight subjects). All subjects received all treatments, but in the same order (there was no attempt to correct for sequence effects by counterbalancing). But time (more than a day) was allowed between treatments, perhaps to allow the attenuation of any carry-over effects.

Stevenson *et al* appear not to have taken the analysis of the data as far as Lundell did. They did not measure the diagnostic accuracy of derived knowledge bases. They did, however, employ a more qualitative approach than Lundell's essentially statistical one. They examined the differences between the kinds of constructs that the experts produced and those that the novices produced.

But although Stevenson *et al* assert that thinking aloud may be less effective than their evaluation technique, they fail to support this empirically. Or, more precisely, they appear not to have designed their experiment to test this.

Computer-Assisted Knowledge Elicitation

Deffner & Ahrens (1989) were not comparing knowledge acquisition methods; they were simply evaluating the single method embodied in a tool of theirs. This method involves having a domain expert enter rules in a formal language and, as a second stage, refine any ill-defined quantifiers used in the rules. According to Deffner and Ahrens, deferring the refinement solves the problem of experts "drying up" when they are interrupted and asked to be more precise about quantifiers.

Like Lundell, Deffner and Ahrens used an artificial domain and created experts in it by training their twenty-two subjects. The domain used was nutritional prediction in a simulation of a person to be fed from a menu. During training, the subjects were free to display their tendency to explore the domain. This tendency was observed by tracing each subject's interactions with the training software.

Although apparently not so by design, Deffner and Ahrens' experiment is a two-group one. The groups were discovered by *post-hoc* cluster analysis of some of the training interaction data. Both groups received the same treatment (elicitation method), but they also had what Deffner and Ahrens assume to be two different levels of expertise. One dependent variable is the accuracy of the generated knowledge base, and this is measured by testing the rules against the simulation. Other dependent variables are the number of rules elicited and the average number of attributes per rule.

Deffner & Ahrens do not say how many of their subjects fall into each group. Nor do they say how they treat the two subjects who do not "fall clearly into one of the two groups".

Deffner & Ahrens (1989, p 359) concede that their tool "may at first sight appear not to be very practical". They try to remedy this lack of external validity by suggesting where the use of the tool might fit in a series of knowledge-acquisition stages.

Elicitation Efficacy

Whereas Lundell (1988) and Stevenson *et al* (1988) were testing hypotheses, Burton *et al* (1987) wanted to determine the relative efficacies and efficiencies of different knowledge-elicitation techniques. Burton *et al* wanted to be able to predict which methods would be most appropriate for which circumstances, so that builders of knowledge-based systems would have some empirical basis for their choices.

Burton *et al* also stopped short of building knowledge bases, and therefore did not reach as far as measuring diagnostic accuracy. However, they did perform other kinds of evaluation on the elicited knowledge, which they coded as "pseudo-English production rules". In a subsequent experiment, these rules were each rated by the experts on a four-point scale ranging from true to false. Thus, Burton *et al* were able to compare (at least for some of their data) the overall quality of rules resulting from each elicitation technique.

In their experiment, Burton *et al* had as independent variables the elicitation method and the expert's personality type. They tried to keep the knowledge representation constant. Their dependent variables were the amount of knowledge elicited per unit time, and the quality of elicited rules.

They also made the distinction between procedural and declarative knowledge. Indeed, they assert that two of their methods (protocol analysis

and formal interview) are likely to elicit procedural knowledge while the others (card sort and laddered grid) are likely to elicit declarative knowledge. But they were forced to conclude that their results did not support this assertion.

Although, like Lundell, they used students as subjects, Burton *et al* did not create instant experts. Thus, their claim of expertise is more credible, especially in the light of Anderson's (1982) assertion that it takes a long period of practice to create an expert. On the other hand, Burton *et al* offer little proof of the subjects' expertise. Burton's subjects were not tested for skill level as Lundell's subjects were.

There is usually some danger of impairing external validity when university students are used as subjects in experiments (see, e.g., Jung, 1969). To get some idea of the effect of using students, Burton *et al* followed their 1987 experiment with another — this time using "real" experts. The earlier results were vindicated (see Burton, Shadbolt, Rugg & Hedgecock, 1990).

Knowledge Engineer as a Variable

Adelman (1989) did not build expert systems with the knowledge elicited from his 138 subjects. What he was trying to do was to determine the effect of two variables (knowledge engineer and elicitation method) on the "predictive accuracy" of the knowledge elicited.

He used two methods ("top-down" and "bottom-up") and six knowledge engineers in what he describes as a "2x6 factorial" design. However, he appears to have had some difficulty in specifying exactly how the knowledge engineers differed from each other. He finally decided to use the institution from which the knowledge engineer had received his or her training as the dimension on which to group them. With this grouping, he reduced his data to that of a 2x3 factorial design. Perhaps it would have been more meaningful to have used

either the psychometric profiles of the knowledge engineers to find clusters (as did Deffner & Ahrens, 1989), or some aspect of their experience.

One of Adelman's chief concerns was with the quality of a domain expert's expertise. Adelman argues that the expert is a factor in the quality of any elicited knowledge. But, as with his knowledge engineers, he appears not to have decided what attribute of the expert is the variable of concern. Yet there is a theoretical reason for focusing on skill level (see Anderson, 1982). In addition, both Deffner & Ahrens and Burton *et al* have found the expert's personality to be important. Thus, Adelman might have tried to vary these systematically. He did not.

Effects of Training

Agarwal & Tanniru (1990) used a completely randomised single-factor design to compare unstructured interviewing with "a specific kind of structured interview". They did this to test four hypotheses about the relative efficacy and efficiency of the two methods of knowledge acquisition.

They did well to find as subjects thirty "expert practitioners who were responsible for [a capital budgeting / resource allocation] decision". The subjects were split into three groups of ten; and each group was given one of the three treatments. But there is some doubt about the consistency with which the treatments were administered in the experiment.

The control group of experts had their knowledge elicited, via unstructured interviews, by what Agarwal & Tanniru call "experienced knowledge engineers". However, only some of these "knowledge engineers" had any experience at eliciting knowledge for expert systems. The others were systems analysts who were experienced at interviewing. Agarwal & Tanniru do not say how many of these interviewers were used, but do admit to having been unable to find enough experienced knowledge engineers.

Novice knowledge engineers, unlike experienced ones, were apparently abundant. Agarwal & Tanniru were therefore able to take care to establish that the novice knowledge engineers all started with comparable lack of experience of the domain and the knowledge-acquisition methods to be used. However, the novice knowledge engineers were given training in only one of the methods (structured interviewing), and left to administer the other method (unstructured interviewing) without the benefit of any training.

As a comparison of two methods, Agarwal & Tanniru's experiment appears therefore to have been biased toward one method. However, they did succeed in showing that knowledge engineers who receive training in a technique are likely to be more efficient and effective using it than those who try to apply a technique in which they have not been trained.

Conclusions

Lessons from the Past

Building a knowledge base can be viewed as the process depicted in Figure 20 on page 120. As noted in Chapter 4, "Evaluation Measures," there are several inputs into knowledge acquisition; various characteristics of these inputs interplay to produce some acquired knowledge in a representation formalism. The knowledge and representation are clearly interrelated, with the latter being the form in which the former can exist in a knowledge base. This knowledge base itself exhibits qualities, such as diagnostic or predictive accuracy.

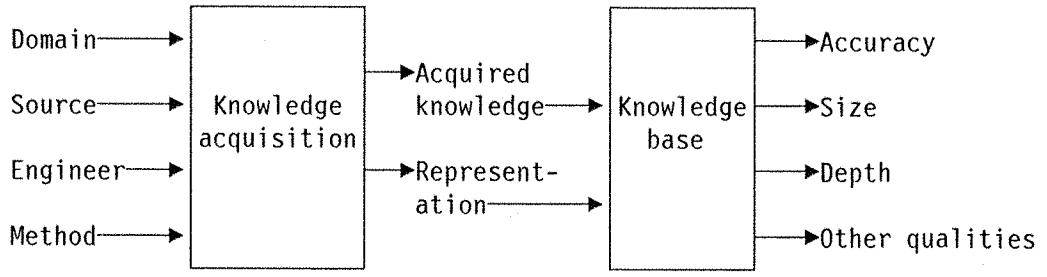

Figure 20. Factors and effects in building a knowledge base.

The qualities displayed by the resulting knowledge base are affected by all the variables that provide input to the knowledge base. A major reason that researchers do empirical investigations is to see how these potential independent variables to the left affect the dependent variables to the right.

The understanding gained from experimenting with these variables is likely to bring more predictability to building knowledge bases, and allow knowledge engineers and planners to make choices based on more solid foundations than are available at present. The researchers discussed in this chapter have experimented for various purposes: testing hypotheses, evaluating a method or a tool, and looking for correlations. The efforts of these researchers have highlighted various challenges.

For example, because of difficulties in maintaining consistency, and the constraints on time, it can be difficult to control the acquisition method. Some researchers have had to use a restricted of form the method (as did Burton *et al*, 1987) or use artificial ones (as did Lundell, 1988). In addition, there are several characteristics of both the knowledge source and the knowledge engineer that deserve attention as variables in their own right (e.g., level of expertise, and personality).

Being sure that subjects are experts (determining their level of expertise) appears also to be a common problem in experiment design. The means used

to solve this problem have not been entirely convincing. Some researchers have selected people whose expertise they appear to think unquestionable (e.g., academics who specialise in the topic). They have also selected subjects whose novice status they appear to think indisputable (e.g., first-year university undergraduates). Others have trained and tested their own subjects; but these researchers appear to have difficulty deciding on appropriate criteria for expert and novice.

There are also problems with using accuracy as a measure of expert-system performance or knowledge-base quality. Each researcher defines accuracy in a different way, according to what is convenient for the experiment design. Even with a consistent definition of accuracy, there is still likely to be a problem. If the test cases are taken randomly from a very large set of historical records, the frequency distribution of certain attributes and classes is likely to have certain characteristics. However, if the test cases are exemplars, or even cases that an expert thinks interesting, then the frequency distribution of attributes and classes is likely to be quite different. Thus, diagnostic accuracies cannot be meaningfully compared without an accompanying comparison of the source of the test cases.

Effects on a New Experiment

The need for consistent application of a knowledge-acquisition method to all subjects can be addressed by modelling the method in a tool, and eliminating the knowledge engineer altogether. However, the use of knowledge-acquisition tools to compare two methods can introduce a confounding variable: the user interface. Even if all the problems identified in this chapter are not solved in the author's design (described in Chapter 6, "Design of an Experiment to Compare two Knowledge-Acquisition Techniques" on page 123), being aware of them helped make the design sounder than it would otherwise have been.

The controlled experiment still appears to be a promising approach to investigating the relationships at work in knowledge acquisition. As present and future researchers respond to the challenges posed by their predecessors, the quality of research design and the value of the findings are likely to be enhanced.

Chapter 6. Design of an Experiment to Compare two Knowledge-Acquisition Techniques

Abstract

This chapter discusses the design of an experiment to test the hypotheses stated earlier. The variables to be manipulated and measured are extracted from the hypotheses. Decisions are made about the type of experiment and how the knowledge-acquisition methods will be administered. The chapter also discusses the type of subjects to be used, and estimates how many of them are required. Some possible domains are explored, and the most appropriate one is selected. The data to be collected and how it will be analysed are also discussed broadly.

Introduction

Chapter 3 states a number of hypotheses about the relative efficacies and efficiencies of the repertory grid technique and KAMSE. These hypotheses are an elaboration of the research question: whether experts find it easier to express their classification knowledge by describing examples or by making distinctions between examples. However, the hypotheses are only conjectures which cognitive psychology and other considerations suggest. These hypotheses have to be tested empirically. As Rugg, Corbridge, Major, Burton, & Shadbolt (1992) have argued, what is expected to happen with knowledge-acquisition methods often has limited basis in reality; and controlled experiments are an ideal way of testing how valid those expectations are.

But controlled experiments are not without their problems. As discussed in Chapter 5, there is not a long history of using controlled experiments in research on knowledge acquisition. The few that have been done suggest that the approach can be productive, if potential pitfalls are anticipated in the design.

Clearly there is a need to design an experiment to test the hypotheses about the repertory grid technique and KAMSE. The variables of interest need to be identified. The type of experiment must be decided. Subjects need to be targeted in numbers sufficient to provide the data with acceptable power. How the variables will be measured and the treatments administered must also be decided. This chapter therefore discusses all these matters, culminating in an agenda for a session of the experiment, and a list of actions that still need to be performed before the experiment can actually be done. This chapter is therefore a blueprint for the rest of the investigation.

Hypotheses

For convenience, the five hypotheses to be tested, which all compare the repertory grid technique with KAMSE, are restated here:

Hypothesis 1: The elicitation of elements in the repertory grid technique is very similar to the elicitation of classes in knowledge acquisition from examples. When both prototypes and exemplars are used as elements, the repertory grid will require the greater effort. In any event, the elicitation of classes will never require more effort than the elicitation of elements. If the elements elicited are all prototypes, then the two methods are identical at this stage; and no difference is expected, between methods, in the mental effort required of the expert at this stage of the process.

Hypothesis 2: A set of constructs (or attributes) adequate for distinguishing among all the elements (or classes) in a domain can be elicited more easily with prompting, as provided in the repertory grid technique, than without, as in KAMSE.

Hypothesis 3: There is little reason to expect an expert to find it easier to express examples in terms of attribute values than to identify similarities and differences between examples of classes.

Hypothesis 4: Overall, for the entire knowledge acquisition cycle (from element / class elicitation to evaluation of the generated knowledge base), there is no efficacy difference between the two methods; the knowledge base generated by induction will be equally accurate between methods.

Hypothesis 5: Overall, as for hypothesis 4 above, there is no efficiency difference between the two methods.

Variables

Figure 21 shows the independent variables, treatments, and dependent variables involved in testing each hypothesis. The stages mentioned are those in Figure 15 on page 72.

Hypothesis	Independent variables	Dependent variables
1	Method	Effort at stage 1
2	Method	Effort at stage 2
3	Method	Effort at stage 3
4	Method	Classification accuracy
5	Method	Total effort (stages 1 to 5)

Figure 21. Variables involved in the hypotheses

Figure 21 shows a single independent variable, which is knowledge-acquisition method, and there are two treatments: the repertory grid technique and KAMSE.

Another possible independent variable is the domain or problem, but that can be fixed by using the same domain for all the subjects. There is little point contrasting synthesis domains against analysis ones, because any results obtained for analysis systems will have some applicability to synthesis systems. This is so because the latter can usually be broken down into analysis components (see, e.g., Hayes-Roth *et al*, 1983).

The subjects themselves are also variable; and perhaps some sense of the variations can be obtained by testing them as Lundell (1988) did, perhaps to establish individual levels of expertise. Indeed, that is one of the things that can be noted about Lundell's subjects: that they were all novices in the sense that they had just learnt the task. Their knowledge was probably entirely at the conscious level because they had just been trained and tested. Perhaps that is something else that affected the external validity of Lundell's results. The experiment carried out by Stevenson *et al* (1988) appears more credible in the way experts and novices are selected. Indeed, Anderson (1982) himself appears to discount the notion of instant creation of experts. (For a fuller discussion of this, see Chapter 5, "The Controlled Experiment in Knowledge-Acquisition Research.")

The dependent variables are classification accuracy and the mental effort expended by subjects at each stage of the process. So the experiment will have to measure these variables. How to measure accuracy was discussed in "Performance" on page 93. Measuring mental effort involves a surrogate, perhaps the number of key strokes that the expert uses, perhaps the amount of time spent interacting with the expert, perhaps other measures.

The real effort, however, is not the physical energy expended in pressing the keys, but the mental resources used to think of the knowledge units. Posner (1973) is one of the writers who argue that mental effort can be measured by the length of time a mental operation occupies. Anderson (1983) also presents empirical evidence about the time taken by various mental operations. So comparative mental effort can be measured by timing operations. One practical implication of this is that mental operations that take smaller amounts of effort enable people using them to be more productive. Other researchers (e.g., Klein & Cooper, 1981; Posner & Klein, 1973; Welch, 1898) have tried to measure

mental effort in other ways. These other approaches are all variants of the secondary task method:

- Welch measured physical force as an indication of mental effort. The greater the effort being expended on a mental task, the smaller the physical force the subject is able to exert concurrently on a hand dynamometer.

- Posner & Klein generated an "auditory probe tone" at random intervals during a mental task. The subject must acknowledge this probe by pressing a designated key. The amount of time taken to react to the probe "serves as a measure of" mental effort.

- Klein & Cooper played a tape recording that called out a random digit every five seconds. The subject was required to call out the sum of the latest digit and the previous one. The error rate indicated the mental effort used on the primary task.

These approaches all distract the subject by occupying some of the limited capacity of his or her consciousness. This capacity would otherwise have been devoted to the primary task. Measuring time is a much less obtrusive approach.

Method

The hypotheses to be tested demand an empirical comparison between two styles of knowledge acquisition: the repertory grid technique and KAMSE.

Let us look at exactly what would be done in an experiment like this. At a high level, one treatment would consist of using KAMSE to elicit subjects' knowledge about the domain — i.e., by having a program that would elicit classes, attributes, and examples. The other treatment would consist of using the repertory grid technique; and this would also require a program. It would

also be necessary to train the subjects in the use of the two knowledge-acquisition tools first, to ensure that their degree of familiarity with the software is not a confounding variable. Those who fail to complete the training successfully would not be given the experimental treatment.

So, in each session, half the subjects will use KAMSE, and the other half will use the repertory grid technique. And they will sit there quietly interacting with the knowledge-acquisition tool. A two-group design (see, e.g., Matheson, Bruce & Beauchamp, 1978) could be used. One group would use the repertory grid technique while the other would use KAMSE. Alternatively, a one-group design could be used if each subject were given both treatments of the acquisition-method variable. This would make more use of the available subjects, reduce errors due to subject variability, and enable a within-subject analysis of the data (see, e.g., Keppel, 1982, p 68). Whether this can be done will also depend on the amount of time subjects have available and the amount of time it takes to go through knowledge acquisition with each method.

Since each subject is to use both methods on a single domain, it is well to bear in mind (see, e.g., Stevenson *et al*, 1988) that once they reconstruct units of the automatic knowledge at a conscious level (which is what either method of knowledge acquisition is likely to do), then subsequent efforts to elicit the same knowledge may be somewhat easier. A reasonable period of time will therefore be left between treatments so as to attenuate carry-over effects. In addition, the treatments will be randomly assigned to subjects, which will probably result in the treatment order being reversed for half the subjects. This will tend to ensure that both methods benefit comparably from any carry-over effects, and take care of sequence effects (which have been assumed to be linear).

Some researchers have not necessarily compared like with like. Some of them have observed a truncated part of knowledge acquisition from examples,

and compared that with some other entire process (e.g., Michalski & Chilausky, 1980). But, as the preceding chapters have made clear, the complete process of knowledge acquisition from examples involves more than just running examples through an inductive program, although that is a vital part of it.

For the subjects using the repertory grid technique, the tool determines when all the required distinctions have been made. Under both methods, the ID3 algorithm will be used to generate knowledge bases for which the classification accuracy would be assessed by running through a set of test cases. These test cases can be exemplars, randomly generated from the attribute values used in the knowledge base, and then classified by the subjects.

Subjects

One of the possible confounding variables is directly related to the subjects. And that is the level of expertise of the subject. If subjects are not going to be trained (if a domain can be found in which all the subjects are likely to be expert), then other hypotheses about the level of expertise could also be tested. However, this is just a second variable, and it would take the experiment out of the realm of single-factor designs and put it into the realm of a two-variable experiment, resulting in a need for more subjects or more treatments per subject, to give the data the desired power.

So either a single level of expertise can be assumed, or the subjects could be tested to verify their levels of expertise. An expert could be defined as someone attaining over a certain grade in the test — and the others could be discarded. But this would be wasteful. It should be possible to interest university students, university staff, and even people outside that environment to participate as subjects, and find a domain in which they are all expert.

Number of Subjects Needed

In deciding on the sample size required to give the data the desired power, it is necessary to look at the "critical effect size", and the population mean and standard deviation. According to Kraemer & Thiemann (1987), the critical effect size is "a measure of how strong the theory must minimally be to be 'important to society'". Traces built up over six months of developing and testing the SCENIC knowledge-acquisition tool (see Chapter 7, "Design of SCENIC: a CAKE Tool for Empirical Work") failed to show any consistent difference in the amount of time taken by one method or the other. These traces therefore provide little indication of what the treatment effect might be, if indeed there is any.

But it is possible to argue, arbitrarily perhaps, that if employing one method on a project would mean finishing the work earlier or using less resources, and if the findings of this experiment can give pointers that can be put to this kind of use, then the community would be interested in the findings. The time saved would be that of both knowledge engineers and domain experts; and it is well known that the latter tend to be busy people typically able to spend only measured amounts of time transferring their skills to a system. It appears reasonable to presume that, if one method were shown to take about one half less time than the other, developers of knowledge-based systems might be interested, other things being equal, in using that method rather than the other. From this viewpoint, an interesting difference might be arbitrarily set at one half of the mean.

Kraemer's Method

The critical effect size (Δ; see Kraemer & Thiemann, 1987, p 38) requires an estimate of the population mean (μ) and standard deviation (σ). According to Burton, Shadbolt, Rugg & Hedgecock (1990), we can expect to acquire between 0.8 and 1.6 clauses per minute using the techniques of interviewing, protocol

analysis, laddered grid, and card sorts. For this purpose, they define a clause as one conditional statement in a pseudo-english production rule. But Burton *et al* report only averages and no variances. It is therefore impossible to use their data to estimate the mean and standard deviation in this experiment.

The most useful estimates came from the trace built up while developing and testing SCENIC. Analysis of these data yields a somewhat crude mean of 9.3 minutes for stages 1 to 4 with standard deviation of 7.4 minutes. Even so, using these estimates would yield a δ (difference in the mean as a fraction of the standard deviation; see Kraemer & Thiemann, 1987, p 38), defined as

$$\delta = (\mu - \mu_0) / \sigma.$$
$$= (9.3 \times 0.5) / 7.4 = 0.63$$

Now, the critical effect size is

$$\Delta = (e^{2\delta} - 1) / (e^{2\delta} + 1)$$
$$= 0.56.$$

Cohen (1977, pp 53-56) has argued in favour of designing for 80% power. The power table (Kraemer & Thiemann, 1987, pp 105-112) for a 0.05 level of significance (α) with a power of 80% and a critical effect size of 0.56 requires a sample size of 21 for the two-tailed test (or about 11 subjects, each receiving both treatments). Little guidance is given in the statistical literature about estimating sample sizes for experiments with several dependent variables. However, because five dependent variables are to be used, the Bonferroni adjustment may have to be made. It is therefore appropriate to use the $\alpha = 0.01$ tables, which give 31 as the required sample size. For a repeated-measures experiment, this means about 16 subjects. Even more subjects will enable a smaller effect to be detected with no loss of power.

Keppel's Method

Keppel (1982, p 71) uses a different procedure to estimate sample size. Keppel's ϕ_A^2 (the square of "a quantity [used] in consulting power charts") can be calculated as

$$\phi_A^2 = s'((9.3 \times .5)^2 / 2) / 7.4^2$$

where s′ is the number of subjects. This yields

$$\phi_A^2 = 0.20\ s'.$$

So $\phi_A = 0.44 \sqrt{s'}$.

Different values of s′ give different denominator degrees of freedom ($df_{denom} = 2\ (s' - 1)$) and different values of ϕ_A. So several values of s′ have to be tried. Keppel's (1982, p 549) power curves for $\alpha = .05$ and one numerator degree of freedom indicate that a ϕ_A of between 20 and 25 subjects are required for 80% power. This translates to between ten and thirteen subjects in a repeated-measures design. The power curves for $\alpha = .01$ and one numerator degree of freedom indicate that between 30 and 35 subjects are needed, which means fifteen to eighteen subjects in a repeated-measures design. As discussed under Kraemer's method, it is more appropriate to use the latter estimate.

Qualifying the Estimates

Two close estimates of the number of subjects required for 80% power — 16 and 18 — have been obtained. Indeed, if an even larger number of subjects could be used without undue increase in cost, it would be sensible to go ahead and use them. However, whereas the estimates obtained above are based on a univariate experiment, there are five dependent variables in the experiment. To cater for the additional variates, the estimates have been subjected to the more stringent .01 α level. p. In addition, Stevens (1980) has noted that small effects are difficult to detect with small samples. The calculations above have been done for 80% power. Stevens argues that even 70% is adequate, but cautions that estimates of mean and standard deviation have to be available for all the

variates, to produce a rigorous estimate of sample size. So the estimates calculated here are only a rough guide. Moreover, if the standard deviation found in the empirical data is smaller than that used here, a correspondingly smaller difference in the mean will be detectable with acceptable power.

Based on the foregoing, it would appear that, although a sample size of about 18 would be adequate, thirty would be a safe number to use, bearing in mind that some subjects might fail to complete the procedure and others might generate unusable data. Even after such attrition, there would probably still be enough usable cases to achieve the power sought.

Knowledge Domain

As discussed in "Method" on page 127, the comparison cannot be made effectively unless one or more suitable domains are found. One of the problems in doing a controlled experiment in knowledge acquisition is that experts are typically scarce individuals. Where they are not scarce, they are busy. So their time is at a premium. As discussed in Chapter 5, "The Controlled Experiment in Knowledge-Acquisition Research," researchers who have done controlled experiments in knowledge acquisition have run into this problem and tried to solve it in various ways — usually not entirely convincing or satisfactory.

An important criterion for an experimental domain is therefore that it should be an area of knowledge in which experts are abundant. A more complete list of criteria can be enumerated as follows:

- The domain should be one in which a sufficient number of experts can be assembled without too much difficulty.
- The domain should involve analysis (classification or diagnosis) rather than synthesis. Results obtained for the former may be applicable to the latter, which are complex exploitations of the fundamental building block (Dechter & Michie, 1984).

- To ensure its suitability for the repertory grid technique, the domain should embody a fair number (say, seven or more) of elements.
- The domain should also have features that make it suitable for KAMSE. That is, it should be reasonably easy to find examples (or cases) that instantiate different classes.

Spelling

At the outset, the author had imagined that it would have been fairly easy to find a domain connected with spelling. It had been argued that skills connected with spelling and the use of words and language are widespread in the society. Domains of this kind include distinguishing a correctly spelled word from an erroneously spelled one.

Judged by some of the criteria, this appeared to be a suitable domain because it is evident that educated native speakers of English develop, throughout their years of schooling, expertise in making this distinction. Moreover, by the time they become undergraduate students, they have developed long experience and a well established skill. Although the number of classes appears to be small (correctly spelled, and incorrectly spelled), examples would create plenty of elements. But correct spelling is not necessarily a widely held skill, even among educated Britons. Even a restricted set (e.g., three-letter words) contains many words that people are not familiar with.

Words

Hall (1965) argues that writing is simply a way of representing speech and that the primary purpose of letters is to represent the sounds of a language. This idea generates the interesting question: what are the sounds of English, and how are they represented in spelling? So it is conceivable that there is abundant expertise on the relationship between writing and speech. For example, if a

word is pronounced in a particular way, how is it spelled? Or, if a word is spelled in a particular way, how is it pronounced?

This domain appears to have ample classes and attributes, but there is one major difficulty: representation of phonemes (minimum significant units of speech) in textual form is something that the average person knows very little about.

There are other possibilities in the field of word formation. For example, Selkirk (1983) has pointed out that there are rules governing the formation of compound words:

- A compound noun may consist of a noun, adjective, preposition, or verb on the left and a noun on the right.
- A compound adjective may consist of a noun, adjective, or preposition, followed by an adjective.
- A compound verb may consist of a preposition followed by a verb.

But there is little evidence to suggest that native speakers of a language engage in this business of forming compound words. For the most part, speakers use compound words with which they are familiar - words that they have read or heard previously. If this is so, the experts in this domain would again be people who have more than a passing acquaintance with linguistics.

Grammar

Another possible domain might be identifying parts of speech. This is a more promising domain than spelling, because it is possible to identify several parts of speech (noun, pronoun, etc.). And it appears reasonable that many people would be able to recognise several instances of words used in contexts that cause them to be classified as one part of speech rather than another.

It appears likely that, even in this domain, the expertise may not be as widespread in the population as might at first be imagined. It would therefore be necessary to screen the experimental subjects by having them do a test to determine their level of expertise. Although those who pass such a test may be classified as experts, those who fail it may not be classified as novices but rather as non-experts. This is because novices are people who are learning the skill and who have assimilated much of the declarative knowledge, but have not accumulated enough experience for compiled productions to have been formed.

In grammar there are only about eight parts of speech, which (even if only prototypes are used as elements) certainly makes triadic elicitation possible. The domain also appears suitable for KAMSE, because people can usually recognise examples of different uses of a word. A trial with two people confirmed the unsuitability of this domain.

Conversation

One domain that appeared promising at first is described by Taylor & Cameron (1987, p 45):

> The illocutionary act... may be the primitive unit of conversation. A description which takes this idea as its starting point is then committed to the following three questions: first, what speech acts exist in a language, second, what are the rules for producing and interpreting them, and third, what are the rules for sequencing them coherently?

Unfortunately, Taylor & Cameron also argue that there is little evidence to suggest that conversationalists have knowledge of these speech acts, and that conversation analysts disagree on both the nature of the speech acts and any rules for producing, interpreting, or sequencing them.

Other Domains

Another possibility was to use, as Lundell (1988) did, the diagnosis of faults in a hypothetical machine. But the authenticity of the expertise generated in this type of domain is not altogether convincing. The same problem arises if, as Deffner & Ahrens (1989) did, the experiment used the prediction of the effect of some action on a simulation.

The mention by Blythe *et al* (1990) of object recognition provides the most suitable domain, of those considered. Each subject could be given a bag of objects that people are familiar with (a pencil, a pen, a paper clip, a button, and so on). This is certainly suitable for the repertory grid technique, and will provide several examples for KAMSE.

Object identification is a domain in which experts are abundant. From early in our cognitive development, we encounter new objects and learn their identities (the words used to describe them). We encounter objects similar to ones we have seen before, but subtly different, for example, a leopard after a tiger, a cat after a dog, a motorcycle after a bicycle, a donkey after a horse, a pen after a pencil, boots after shoes.

If Anderson (1982) is right, the more a person exercises this knowledge and makes identifications involving these distinctions, the more this expertise becomes compiled, automatic, and also difficult to articulate, indeed, the more skilled a person becomes at identifying these objects. In western societies like Britain, most people (by the time they become adults) have already exercised the object identification skill often enough to have become good enough at it to be worthy of the description "expert". Indeed, it is difficult to find anyone unfamiliar with common objects like pens, pencils, rubber bands, coins, and erasers.

An experimental domain was created consisting of a domain package of a small number of objects that any university student or office worker or

academic living in Britain would be expected to be very familiar with. The domain objects were therefore ones that these kinds of people would be able to identify quite easily. Indeed, this object identification domain affords the flexibility of designing various effects into the domain by carefully selecting certain combinations of objects. Based on preliminary trials, the domain was designed so that the acquisition of the knowledge could be completed within an acceptable length of time.

Some Requirements for the Apparatus

It would be useful to have an interactive tool to support KAMSE. This tool would essentially ask for examples of all classes, and then induce rules by using, say, the ID3 algorithm on the data. Having done that, a further set of examples would be needed for testing the diagnostic accuracy of the resulting knowledge base. Exemplars could be used for this purpose.

On the other hand, if enough is known about the domain beforehand, a "gold standard" set of test cases could be elicited from a respected expert of some sort. This gold standard could be built before the experiment, and would contain sufficient examples to test the knowledge bases generated in the experiment. However, it could also happen that having come up with a set of test cases from an expert, the attributes that s/he used might not include some of the attributes elicited from subjects during the experiment. Also the gold standard could include attributes that fail to come from some of the subjects. Such discrepancies would make the gold standard useless.

But the efficacy of a knowledge acquisition method is a measure of how well the expertise of the knowledge source has been retrieved and modelled. It is not an absolute measure of how good the elicited knowledge is; that is not just a function of the method but of the expert as well. What is required is a measure of how well the source's knowledge has been captured. It is therefore

more meaningful to have each expert diagnose randomly generated examplars, which can then be used as test cases against the new knowledge base.

Having administered the treatments, a set of data would have been collected in terms of treatment and dependent variables. As for KAMSE, software is needed that will use the repertory grid technique and then transform the grid into a rule base. As for KAMSE, software is needed that will elicit examples, and use them as a set of test cases. Ideally, it should be possible to batch-process the test cases, and automatically record the results. It may also be necessary, as a preliminary to the experiments, to elicit the knowledge from some acknowledged expert, not for the gold standard, but to test the software.

It is therefore necessary to design and build those bits of software and an expert-system shell. This design is discussed in Chapter 7, "Design of SCENIC: a CAKE Tool for Empirical Work."

Data to be Collected

Performing the experiment discussed in the preceding sections of this chapter would generate a set of results for the repertory grid technique and a set for KAMSE. These data would enable the statement that, on the average, the effort involved in eliciting knowledge under the repertory grid technique was so much and under KAMSE so much. The significance of any main effects of method could be evaluated by using statistical F-tests. In addition, differences in the respective resulting diagnostic accuracies of knowledge bases generated under each method would be found. The significance of these differences could also be evaluated. Since several dependent variables are being affected by the independent variables, the data will have to be subject to a multivariate analysis of variance (MANOVA).

Summary of the Experiment

Five hypotheses have been stated (see Chapter 3, "Some Implications of Cognitive Psychology for Knowledge Acquisition" on page 57). Here is the essence of the questions raised by these hypotheses: Do experts find it easier to express classification knowledge by describing examples or by making distinctions between examples?

For the purposes of the experiment, ease is defined as having two components: efficiency, and efficacy. Efficiency can be measured in terms of effort (time) per unit of knowledge (class, attribute, attribute value, example, rating) acquired. Efficacy can be measured in terms of classification accuracy of the acquired knowledge.

A single-factor repeated-measures (within-subjects) design will be used. Each of about 30 volunteer subjects will be randomly assigned to one of the two knowledge-acquisition methods. Each subject will be given a diskette and the corresponding instruction sheet for the assigned method. Each subject will be trained to use the relevant portion of a knowledge-acquisition tool embodying the method assigned (for about 10 minutes). This training will consist of using the tool to build a small knowledge base with the aid of some written instructions.

When each subject completes the training exercise, s/he will be given a domain package and asked to use the tool until sufficient knowledge has been acquired to distinguish between all classes in the domain. (The domain package is useful as a visible reminder of the limits of a domain, which, after all, is an artificial subset of a real one.) The tool will create a knowledge base from this knowledge. Then each subject will classify some exemplars (about 32), which will be used as test cases to evaluate each knowledge base. Invite the subjects who completed their first session to return a week later for the other method. During knowledge elicitation, the tool will record:

- Method
- Time spent on each stage
- Number of knowledge units elicited at each stage.

During evaluation, the tool will record:

- Method
- Number of cases processed
- Number of cases in which the inferred class matched the expected class.

Store the data for subsequent statistical analysis.

Agenda for the Experiment

- Administration (5 minutes)
  - Get a list of subject's names; number them and randomly assign to method.
  - Hand out appropriate diskettes & instructions.
- Briefing (10 minutes)
  - Explain the task
  - Self-training
  - Hand out domain packages.
- Experiment (30-40 minutes estimated)
  - Handle problems individually, log them, note time wasted.
  - When someone finishes, collect his or her diskette.

Conclusions

It is possible to test the hypotheses by using a single-factor repeated-measures (within-subject) design involving eighteen or more subjects. Object identification is a viable domain, in which there is not likely to be a shortage of experts.

If a tool is available, the appropriate data can easily be collected to do a MANOVA and identify significant main effects of the knowledge-acquisition method.

The variability of administration of knowledge-acquisition methods can be minimised, and the appropriate generation and evaluation of knowledge bases can be handled efficiently by using a tool embodying the repertory grid technique and KAMSE. Chapter 7, "Design of SCENIC: a CAKE Tool for Empirical Work" on page 143 describes the design of such a tool.

Chapter 7. Design of SCENIC: a CAKE Tool for Empirical Work

Abstract

Using a knowledge-acquisition tool in a controlled experiment can reduce the potential for inconsistent administration of the method treatments. When more than one method is involved, differences in the user interface may introduce a confounding variable. A single tool with a consistent user interface throughout can minimise that effect. A tool also facilitates collection of data, analysis and processing of knowledge, and evaluation of knowledge bases.

This chapter describes the design of SCENIC, a tool embodying both the repertory grid technique and KAMSE. SCENIC was built specifically for the experiment designed in Chapter 6, "Design of an Experiment to Compare two Knowledge-Acquisition Techniques." SCENIC also includes features that make it applicable outside the narrow confines of the research laboratory. A second version, capable of handling more complex domains, is being designed to take it further in that direction.

Introduction

The experiment design described in the previous chapter relies heavily on the availability of a knowledge-acquisition tool embodying two methods: the repertory grid technique and KAMSE. The tool, called SCENIC after its author's first initial and surname, will assume that an analysis domain has been identified, and will acquire knowledge through all the subsequent stages, including generation of a knowledge base and the evaluation of it through batch consultation.

The stages followed by the two acquisition methods are shown in Figure 22 on page 144 (for convenience, repeated here from page 72).

Stage	Repertory Grid Technique	Minimal Set of Examples
1	Elicit elements	Elicit classes
2	Repertory test: elicit constructs	Elicit attribute descriptors and values
3	Repertory test: elicit ratings	Elicit examples
4	Induce rules	Induce rules
5	o Classify evaluation exemplars o Evaluate knowledge base	o Classify evaluation exemplars o Evaluate knowledge base

Figure 22. The knowledge acquisition stages of the two methods implemented in SCENIC

The first stage is the elicitation of classes (for KAMSE) or elements (for the repertory grid technique). Whereas this search for classes is seldom highlighted in the literature on machine learning, it is openly discussed in articles on the repertory grid technique (see, e.g., Boose & Bradshaw, 1987; Garg-Janardan & Salvendy, 1988). Both methods use the same program; so both the processing code and the user interface are identical. The only discernible difference between the two methods at this stage is the panel heading (see pages 154 and 159).

The second stage is different between methods. Having identified the elements, the user of the repertory grid technique must also identify the constructs that constitute the dimensions of the problem space. To elicit these constructs, the tool administers a repertory test — presenting triads of elements and asking for the odd one out to be selected. When this selection is made, the tool asks for the trait that makes the two similar elements different from the

other. Indeed, the essence of the repertory grid technique is the way that constructs are elicited. For each construct entered, the tool goes into a rating stage, which elicits a rating, on the particular construct, for all elements. It is worth pointing out that there is another way that tools do the repertory test. ETS (Boose, 1985) and AQUINAS (Boose & Bradshaw, 1987), for instance, elicit all the domain constructs before entering the rating stage. However, this approach is potentially inefficient, because it cannot assess when enough constructs have been elicited. SCENIC's interleaving of the rating and construct elicitation stages allows the grid to be analysed continually, to determine when constructs are parallel, rotated, potential cluster members, or no more are needed. This enables the tool to complete the knowledge-acquisition session quickly.

Under KAMSE, the parallel stage is the elicitation of attributes, which is not subject to the degree of analysis and monitoring that accompanies the elicitation of constructs. As with the identification of classes, the problem of finding attributes is seldom acknowledged in the machine learning literature. Perhaps this is because, in inductive knowledge acquisition, it is usually assumed that classes, attributes, and cases are all given. But, as discussed in Chapter 2, "Learning Without Case Records: a mapping of the repertory grid technique onto knowledge acquisition from examples," there are situations where this assumption is not correct, and a domain theory must be developed before any induction algorithm can be put to work.

In Chapter 2, it is argued that the repertory test has to be goal-oriented to prevent it from going on for an unnecessarily long time, repeatedly eliciting parallel constructs. So the repertory test has to guide the user toward completion (that point at which all the elements are distinguishable on their ratings from each other). Until that state is reached, the repertory test would go on presenting random triads. One challenge for any repertory-grid

knowledge acquisition tool is therefore to encourage the user not so much to find new ways in which the same elements differ, but rather to find new pairs of elements and the traits that they share. So SCENIC reduces the randomness as the number of confused pairs of elements gets smaller, thus ensuring that a triad is never presented unless it contains a confused pair.

When the tool has elicited distinctions between all the elements, and no confused pairs remain, it goes into the next stage: the induction of rules from a set of examples generated from the repertory grid.

With KAMSE, once the classes have been elicited, the tool then elicits descriptors and values of all the attributes (that the user thinks have to be considered in distinguishing the domain objects from each other).

After these attribute descriptors and values have been entered, the tool elicits a number of examples. A number of checks are also built into KAMSE. At least one example of each class must be entered, if the class is to be represented in the eventual knowledge base. The tool reminds the user of classes for which no cases have been entered. The tool also alerts the user when clashing cases are entered (i.e., two cases with identical attribute values, but different classes).

When the user is satisfied that enough examples have been entered to adequately represent all the classes, s/he causes the tool to progress to the next stage, inducing rules from the examples entered. A single program performs this induction for both knowledge-acquisition methods.

The next stage for both methods is to generate a set of exemplars from random attribute values. These exemplars have to be classified by the user. Having decided on classes for all thirty-two exemplars, the user then enables the tool to progress to the next stage — using the test cases for batch evaluation of the knowledge base. This process counts and logs the number of test cases

seen, and the number in which the tool's diagnosis agrees with that of the expert.

Throughout all these stages, for both methods, the tool collects and logs data required for later statistical analysis. The data logged for each of the five stages is:

- Number of keys pressed
- Start and end times
- Number of knowledge units (i.e., classes, attributes, values, and examples) acquired.

Structure of the Tool

To provide the functions described in the previous section, a single integrated system was designed and developed. As Figure 23 on page 148 shows, the tool includes four high-level functions: (RGT, which administers the repertory grid technique), KAMSE, options (e.g., setting tracing on and off), and a performance system (expert-system shell) through which the knowledge may be consulted or evaluated.

For uniformity, it is important to have both the RGT and KAMSE functions using similar user interfaces. Such uniformity helps eliminate the user interface as a possible confounding variable, when the tool is used in an experiment.

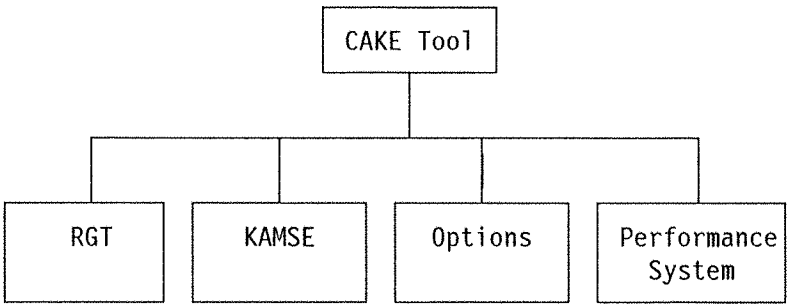

Figure 23. Functions of the SCENIC CAKE tool

The RGT Function

The RGT function interacts with the user, and elicits the elements, constructs and ratings that make up a repertory grid for the domain of interest. Primarily at stages 2 and 3, the function analyses the emerging grid for parallel constructs, confused pairs of elements, and clusters of constructs. It then generates an efficient knowledge base by first transforming the repertory grid into a set of examples, and then using the ID3 induction algorithm to distill these into a heuristic knowledge base. It then generates some exemplars for the user to classify. These exemplars are passed to the performance system for evaluating the knowledge base.

Inputs and Outputs

- A repertory grid
- Panels to user
- Knowledge base
- A file containing test cases expressed in terms of the domain model used in the knowledge base
- A trace file containing measures taken during the experiment.

Subfunctions of RGT

The RGT function performs subfunctions which fit together in a hierarchy, as shown in Figure 24 on page 149. Each of the subfunctions is described below.

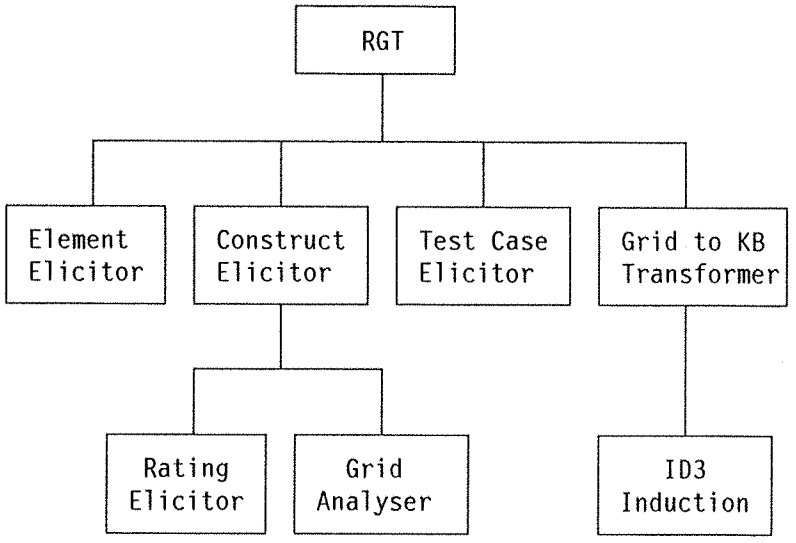

Figure 24. Subfunctions of the RGT function

Elicitation of Elements: This subfunction asks the user to type in the names of all the elements. These are stored in a list. When the user indicates that this list is complete, the elicitation of constructs will begin.

Elicitation of Constructs: This subfunction selects at random three elements at a time, and asks the standard questions about similarity and difference. Both poles of each construct will be elicited, and then the rating subfunction will be invoked. It is also possible for a single triad to be used in eliciting several orthogonal distinctions, if it is "milked" (Gammack, 1987). A tool could possibly gain efficiency by doing this. However, use of this device while the tool was being tested evoked unfavourable responses from users, who found the technique tedious.

Because of the bipolar nature of constructs, the repertory grid technique, in its simplest form, generates knowledge bases that ask questions with "yes" and "no" (and perhaps "don't know") being the only possible answers. Depending on the capabilities of the performance system's inference engine, some kind of certainty estimate may allow these sharp distinctions to be blurred. KAMSE, however, allows multi-valued attributes and is therefore capable of generating knowledge bases that ask multiple-choice questions. Constructs can be clustered so as to become attribute values, as explained on page 34.

The number of constructs generated by the repertory grid technique depends on the number of triads of elements presented to the user. Some systems therefore seek to limit this number. But some of these limits are crude: Garg-Janardan & Salvendy (1988) limit the number of triads to one half the number of elements in the grid. It may, however, be more effective to proceed with triadic elicitation until enough constructs have been gathered to distinguish every element from every other.

Rating of Elements: For each construct, this subfunction asks the user to enter ratings for all the elements stored in the grid. Proceeding in this sequence implies that whenever additional elements are subsequently elicited, they will be unrated in terms of previously elicited constructs. The user could be invited to rate the new elements on these constructs, but that embellishment is not needed in the experiment. The tool will assign the central rating value (don't know) to such elements.

Grid Analysis: This subfunction analyses the grid (see Shaw, 1981) to identify confused pairs of elements, parallel constructs, rotated constructs, and clusters of constructs. The user is notified of parallel and rotated constructs, and left to do with the information whatever is appropriate. The user is also informed of possible construct clusters, and invited to confirm and name them. The

presence of confused pairs causes the tool to continue eliciting constructs (and ratings). The grid is considered completed when it contains no confused pairs; when this state is reached, the tool progresses to creating a knowledge base.

Transform Rep Grid into a Knowledge Base: Elements and clusters of elements in the completed grid are used to create classes. Constructs are converted to attributes, as discussed on page 34. Then one example is created from each grid element. This involves converting ratings to attribute values: a rating higher than 3 converts to "yes"; one lower than 3 becomes "no"; and a rating of 3 becomes "don't care". The examples so created are processed by the ID3 induction algorithm (described in "Induction by the ID3 Algorithm" on page 160), which finds regularities in them; and creates a knowledge base compatible with the performance system.

Eliciting Test Cases: Random values of the attributes used in the knowledge base (as opposed to the redundant ones) are combined to form a set of exemplars which need to be classified. The classes are elicited from the user. When the exemplars have been classified, they are passed to the batch evaluation subfunction.

Batch Evaluation: This subfunction evaluates the knowledge base by consulting with the performance system on all the test cases available.

The KAMSE Function

The KAMSE function elicits the pieces of information necessary to generate a knowledge base by machine induction. The function interacts with its user to elicit classes, attribute descriptors and values, and examples, which can be prototypes, exemplars, or historical cases. For each example, the tool asks for the class and the value of each attribute. The examples are used by the ID3 induction function to create a knowledge base, which is evaluated by the batch evaluation subfunction.

Inputs and Outputs

- A knowledge file storing the partial domain model and examples to be used for induction.
- Panels to user
- Knowledge base
- A file containing test cases expressed in terms of the domain model used in the knowledge base
- A trace file containing measures taken during use of the tool.

Subfunctions of KAMSE

The KAMSE function is structured as a hierarchy of subfunctions, as shown in Figure 25. Each subfunction is described below.

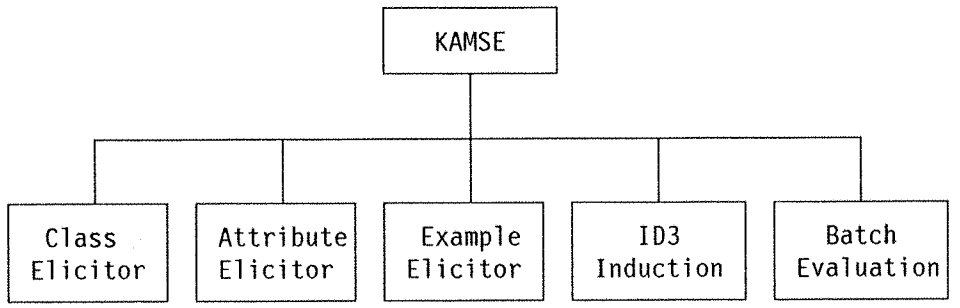

Figure 25. Subfunctions of the KAMSE function

Elicitation of Classes: This subfunction elicits classes, and is identical to the element elicitation in the KAMSE function. When the user is finished, the tool progresses to the elicitation of attributes.

Elicitation of Attributes: This subfunction will ask the user for the factors considered in deciding among the classes. For each attribute, the values relevant to the domain will also be asked for. When the user is finished, the tool progresses to eliciting examples.

Elicitation of Examples: At least one example of each class will be elicited. This subfunction will alert the user about conflicting examples, whenever one is entered. When the user indicates that all examples have been entered, the tool will progress to the ID3 induction subfunction. At any time, the user can go back to previous stages (e.g., elicitation of attributes) to modify the domain model.

ID3 Induction: This subfunction will read through the examples, and find regularities using the ID3 algorithm (as in Quinlan, 1979, p 171; Quinlan, 1986; Hart, 1986, p 114).

Batch Evaluation: This subfunction is identical to that described under the KAMSE function.

Data Elicited and Stored

- For each domain, a short name is elicited and stored.
- For each element, the following pieces of information are elicited and stored:
  - Name
  - Ratings (one for each attribute)
- For each construct/attribute, the following pieces of information are elicited and stored:
  - Descriptor
  - Opposite pole (only for constructs)
- For each case, a case identifier is generated, and the following pieces of information are elicited and stored:
  - Class identifier
  - A value for each attribute.

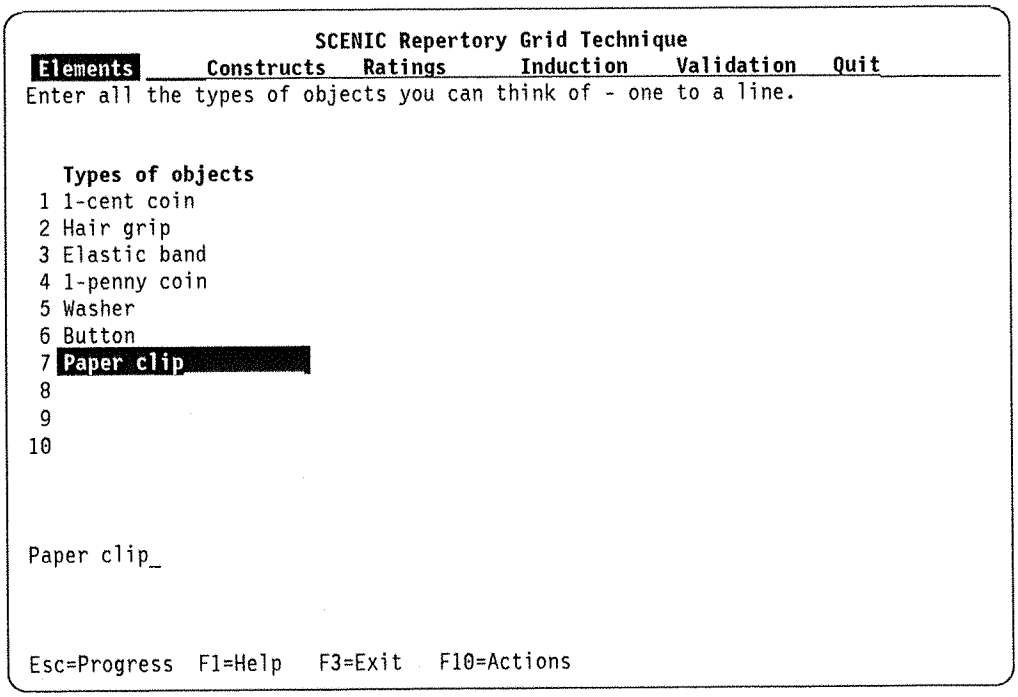

Figure 26. The panel used to elicit elements

Repertory Grid Panels

The panel shown in Figure 26 on page 154 is used to elicit elements under the repertory grid technique. The user can move the pseudo-cursor up and down the list, making changes to any of the elements. New elements are added at the end of the list. As with all the panels used in SCENIC, the menu bar at the top indicates the stage reached in the knowledge acquisition process. For each triad of elements, the construct that distinguishes between them is elicited through a set of three related panels, used at the construct elicitation stage. First the user is asked to select the element that differs from the other two. This is done on the panel shown in Figure 27 on page 155. The user positions the pseudo-cursor on the element to be selected, and then presses the Enter key. When the user has selected one of the three elements, the selected element is

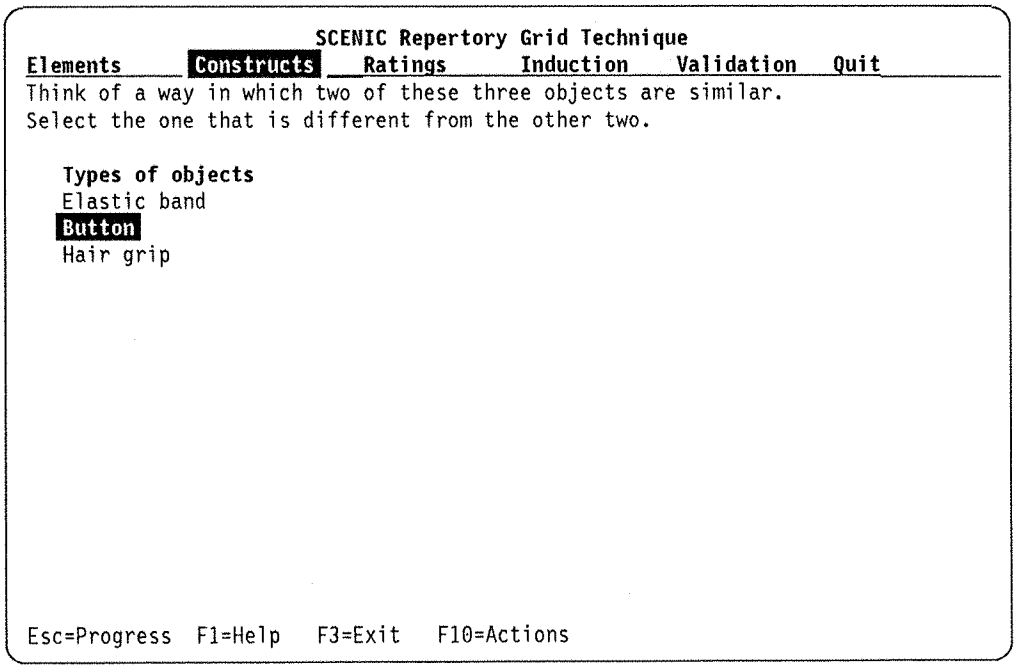

Figure 27. One of the panels used to elicit constructs

moved away. It appears by itself at the bottom of the triad, separated from the other two elements. The user is asked to input the construct on which the selection was made. All this takes place on the panel shown in Figure 28 on page 156. The questions asked by a repertory grid tool have to be carefully chosen if the desired kind of information is to be successfully elicited from the tool's user. For instance, the question "What makes A similar to B and different from C" elicits a construct that may be useful in classifying. However, a different question must be found to elicit distinctions between circumstances in which one would select A or B but not C. A more effective question for a diagnostic domain might be: "Under what circumstances would you select A or B but definitely not C?".

Moreover, the question tends to elicit a noun construct when what is required is an adjectival one. Even in a classificatory domain, the question still

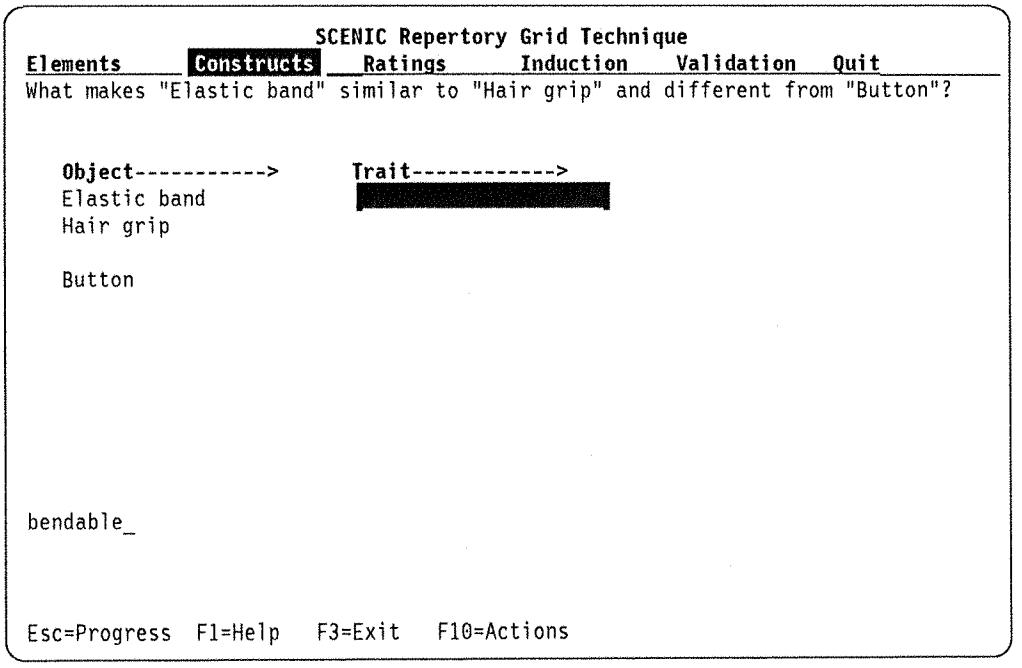

Figure 28. The construct-elicitation panel with a separated triad

tends to elicit a comparative construct such as "much heavier" when what is needed is a positive one such as "heavy". The most appropriate wording for the question can be determined by the tool based on a description of the goal of the domain.

After the main pole of the construct has been entered, the panel shown in Figure 29 on page 157 elicits the opposite pole. When both poles of the construct have been entered, the user can press the Enter key to confirm that they are correct. This causes the tool to progress to the rating stage. Here, the user is asked to rate all elements on the construct. Three of the elements arrive here pre-rated from the construct stage, but these ratings can be changed, if desired. The panel shown in Figure 30 on page 158 is used for rating. One question that arises is what kind of rating scale should be built into SCENIC. Kelly (1955) insisted on a two-point rating scale on which every element should

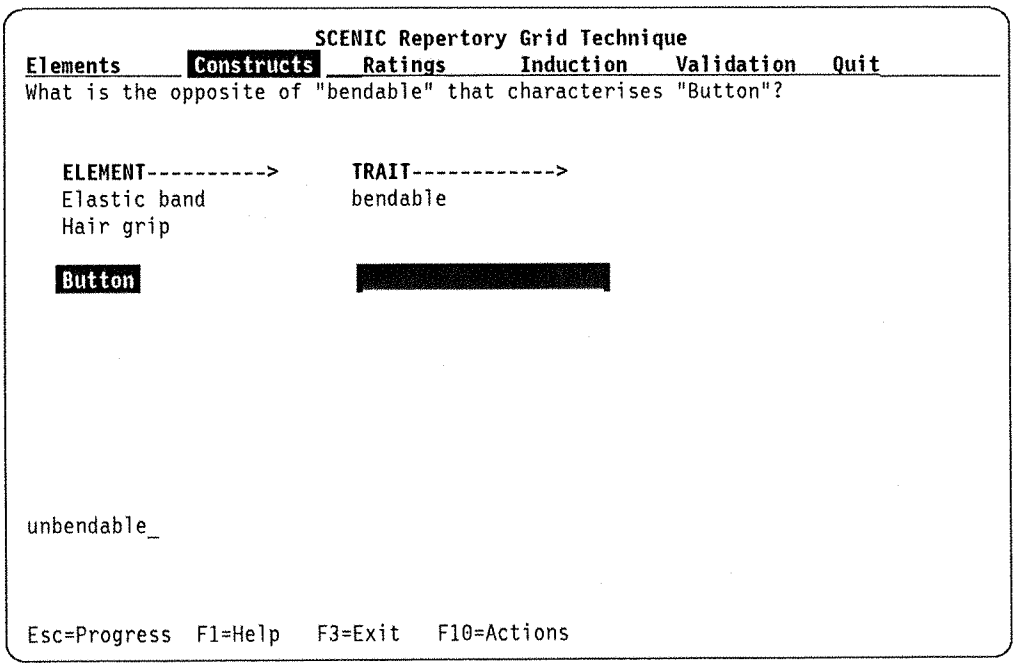

Figure 29. The panel that elicits the opposite pole of a construct

be rated. KITTEN uses a five-point scale for rating elements; and Shaw & Gaines (1987, p 256) justify this by arguing that "The use of a multi-point scale with an odd number of values allows for a central rating which does not force the user to choose either pole."

The commercial knowledge-acquisition tool NEXTRA uses the repertory grid technique. Traits (called "qualities") are bipolar with "neither being an option". This scheme is equivalent to a three-point rating scale on which 1 and 3 are the opposite poles of the construct, and 2 is neutral.

KAMSE Panels

Classes are elicited in exactly the same way as elements in the repertory grid technique. The similarity between the panel used for the purpose, which is shown in Figure 31 on page 159, and the corresponding one for the repertory

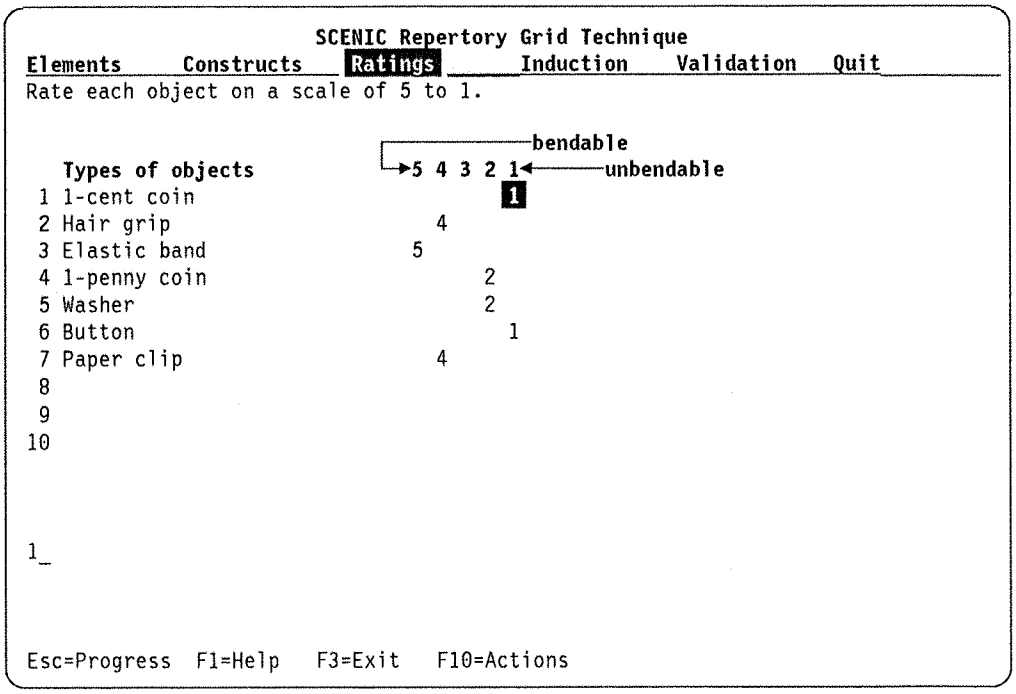

Figure 30. The rating panel

grid (in Figure 26 on page 154) is obvious. Attributes are elicited on the panel shown in Figure 32 on page 160. The classes previously elicited are listed at the left to remind the user of the scope of the domain. But this panel is used for input of attributes, each consisting of a descriptor and multiple values.

When a user lists attributes that distinguish between classes of objects, s/he may not know whether all the attributes and their possible values are essential to the analysis task. The machine induction process may reveal that (based on the set of examples used) some of the given attributes or values are redundant (or at least redundant for the training set provided). Alternatively, the given attributes may not be sufficient to distinguish between all classes.

At the example elicitation stage, the user enters at least one example of each class. Each example is described in terms of the attribute values

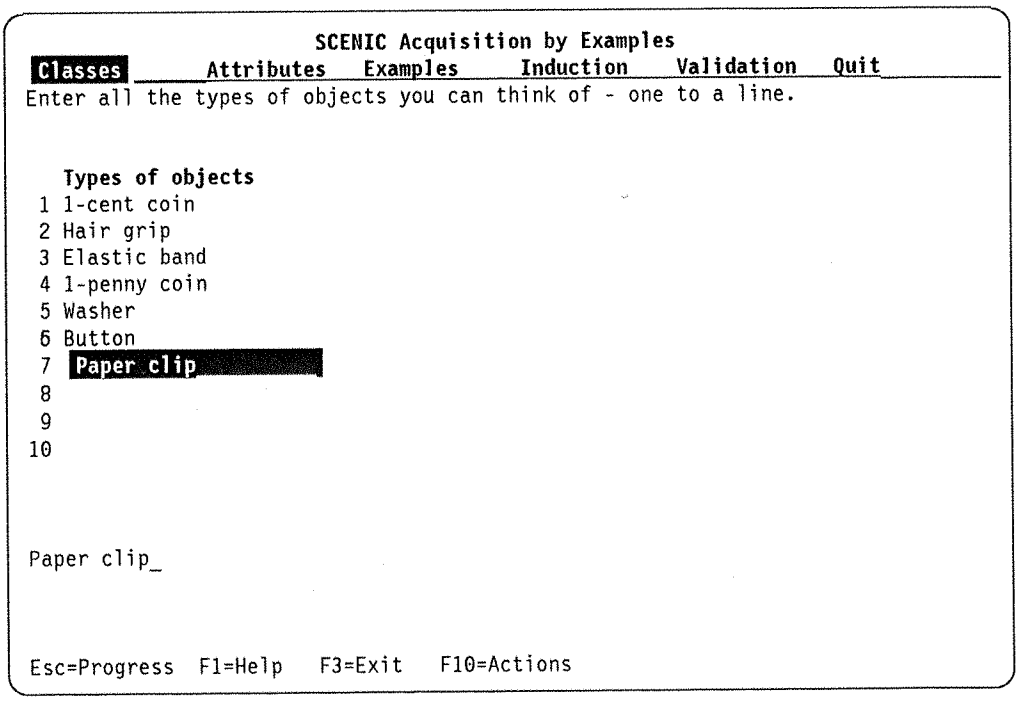

Figure 31. The panel used to elicit classes

previously entered. Neither the attribute values nor the class are keyed in; because they are already in the computer, the user simply selects them from the lists in which they are stored. The panel shown in Figure 33 on page 161, which facilitates all that, is used to elicit examples. Hart (1986, p 116) shows a set of seven cases in which two have identical attributes but different classes. One way of treating such conflicting cases in induction is simply to ignore them. But when this is done, something is lost, because what is required is an additional attribute to distinguish between the clashing cases. It could equally be that one case is erroneous and needs to be corrected. Indeed, there appears to be no reason why a tool eliciting examples could not check for such clashes before applying the induction algorithm. In addition, the tool will also inform the user of any classes for which no cases have been given.

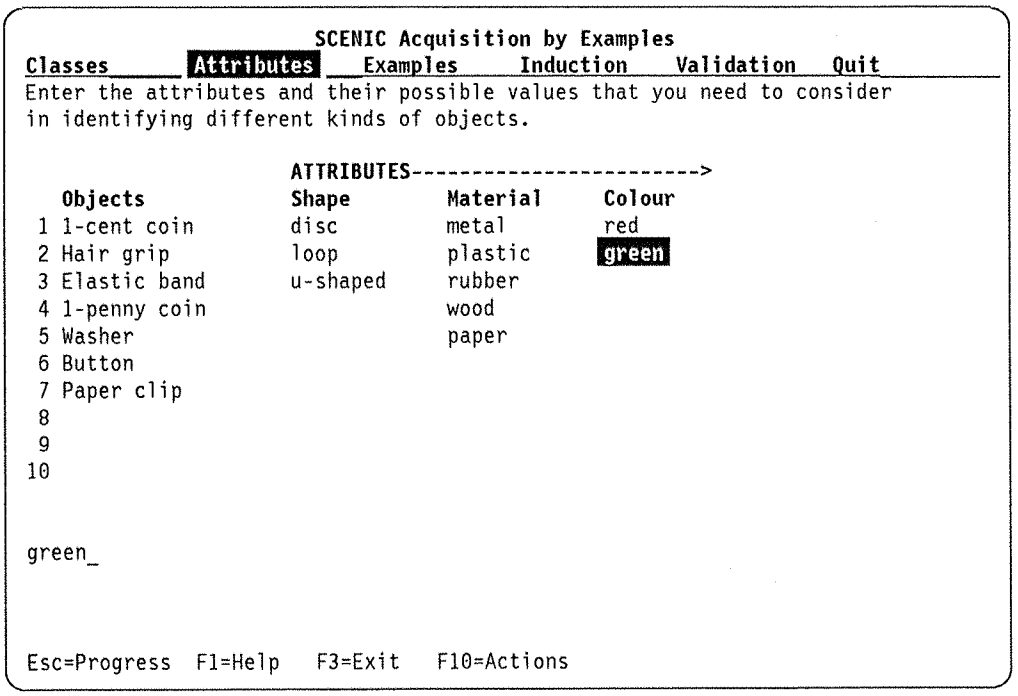

Figure 32. The panel used to elicit attributes

Induction by the ID3 Algorithm

Some writers see induction as an essential part of the analysis and organisation performed on knowledge gathered by the repertory grid technique. Rappaport & Gaines (1990), e.g., mention "topological induction" in discussing NEXTRA, while Boose (1985) refers simply to "inductive generalization" in discussing ETS. Boose goes even further, saying that the rating grid represents "training examples" (p 501).

Shaw & Gaines (1987, p 258) also mention induction in their discussion of the tool KITTEN:

> The resultant grids are analyzed by ENTAIL which induces the underlying knowledge structure as production rules that can be loaded directly into an expert system shell.

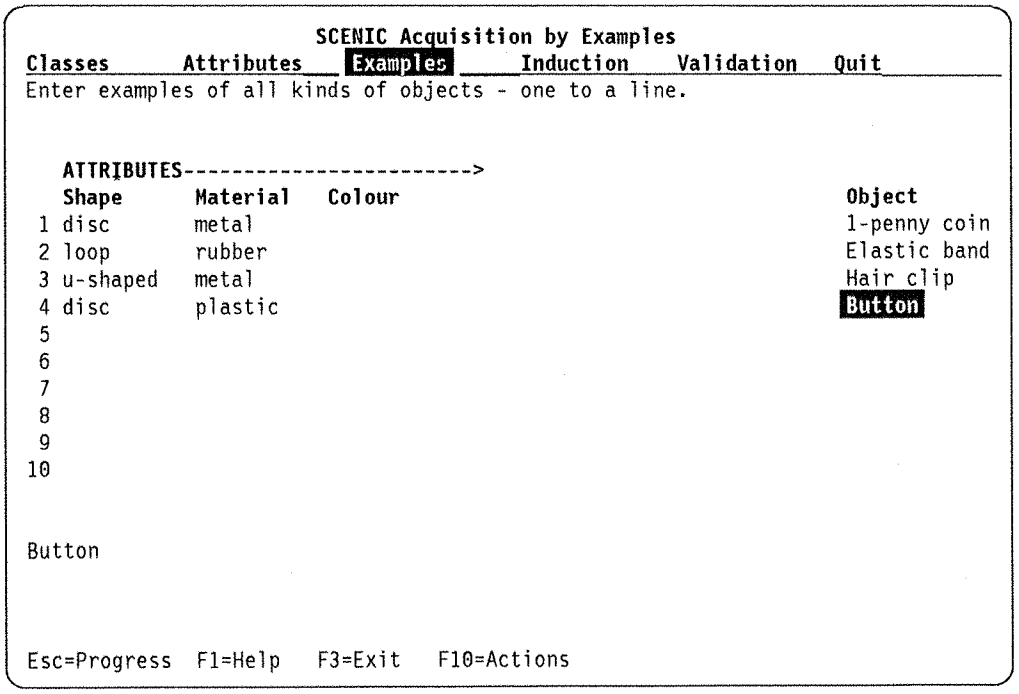

Figure 33. The panel used to elicit examples

Indeed, it is often useful to use an induction algorithm to transform a repertory grid into a rule base. This transformation can have two stages: first a decision tree is generated, then the tree is transformed into a set of simpler production rules. The induction process acts to organise the attribute tests into an efficient sequence and to prune redundant tests. Applying induction in this way makes sense only when we view each element and its ratings as a case. And it is hardly fanciful to argue that these cases could also be used to train a neural network (by repeated presentation if necessary).

Let us look at a specific situation as a preliminary step to writing a generalised algorithm. Assume a case base having five cases (C1 to C5), a closed set of three classes (D1, D2, and D3), and seven attributes (A1 to A7). Assume further that each attribute has the same three possible values (".", "Y",

and "N"). The attributes have the following specific meanings: "Y" = yes, "N" = no, and "." = don't care. In this situation, "." is actually a superclass of "N" and "Y". Such a case base would look like this:

Case Id	Attributes A1	A2	A3	A4	A5	A6	A7	Class
C1	Y	Y	N	.	Y	N	.	D1
C2	.	Y	.	Y	.	.	.	D1
C3	N	.	N	N	.	Y	Y	D2
C4	Y	N	.	.	Y	.	Y	D2
C5	.	N	Y	.	N	.	.	D3

Quinlan (1986) suggests that the program find the test with the highest "information gain". For the time being, it is assumed that attribute A2 has the highest information gain. Splitting this case base by the attribute values of A2 leads to the following sequence:

Set	Case Id	Attributes A1	A2	A3	A4	A5	A6	A7	Class	Test
1	C3	N	N	N	N	.	Y	Y	D2	A2="N"
1	C4	Y	N	.	.	Y	.	Y	D2	A2="N"
1	C5	.	N	Y	.	N	.	.	D3	A2="N"
2	C1	Y	Y	N	.	Y	N	.	D1	A2="Y"
2	C2	.	Y	.	Y	.	.	.	D1	A2="Y"
2	C3	N	Y	N	N	.	Y	Y	D2	A2="Y"

The case base has thus been divided into two sets of cases according to the value of attribute A2. It will be noted that case C3 exists in both sets. This is because of the way the don't-care value, which C3 has for attribute A2, is treated. Because set 1 consists of cases of different classes (D2 and D3), we need to find a further test to split it. This choice would again be based on maximising information gain.

To find the best test, the information gain or entropy is calculated for every attribute not previously used (i.e., every attribute except A2). The method used for this calculation is that given in Quinlan (1986, pp 89-90). Assuming that using attribute A5 as the next test will result in the highest information gain in set 1, splitting set 1 by value of A5 results in a case base subdivided as follows:

	Case	Attributes								
Set	Id	A1	A2	A3	A4	A5	A6	A7	Class	Test
---	---	--	--	--	--	--	--	--	-----	----
1.1	C3	N	N	N	N	Y	Y	Y	D2	A2="N" &
	C4	Y	N	.	.	Y	.	Y	D2	A5="Y"
1.2	C3	N	N	N	N	N	Y	Y	D2	A2="N" &
	C5	.	N	Y	.	N	.	.	D3	A5="N"
2	C1	Y	Y	N	.	Y	N	.	D1	A2="Y"
	C2	.	Y	.	Y	.	.	.	D1	
	C3	N	Y	N	N	.	Y	Y	D2	

Set 1.1 is a single-class set and therefore needs no further splitting. Let us focus on set 1.2 and assume that attribute A3 has the highest information gain. Splitting set 1.2 on the attribute A3 results in the following case base:

Set	Case Id	Attributes A1	A2	A3	A4	A5	A6	A7	Class	Test
1.1	C3	N	N	N	N	Y	Y	Y	D2	A2="N" &
	C4	Y	N	.	.	Y	.	Y	D2	A5="Y"
1.2.1	C3	N	N	N	N	N	Y	Y	D2	A2="N" & A5="N" & A3="N"
1.2.2	C5	.	N	Y	.	N	.	.	D3	A2="N" & A5="N" & A3="Y"
2	C1	Y	Y	N	.	Y	N	.	D1	A2="Y"
	C2	.	Y	.	Y	.	.	.	D1	
	C3	N	Y	N	N	.	Y	Y	D2	

Each time a split is done, the program stores information about the composition of each set created, including the sequence of tests used. This splitting process is continued until each set contains examples of a single class. At that point, induction is complete. The series of tests that were used to create a single-class set form a production rule. For example, set 1.2.1 was created by the rule:

```
IF   A2 = "N"
AND  A5 = "N"
AND  A3 = "N"
THEN Class = D2
```

The production rules are generated in a representation compatible with the performance system.[6]

---

[6] It is worth noting that SCENIC has the facility to export either frame-intensive or rule-intensive knowledge bases into AD/Cycle The Integrated Reasoning Shell, a commercial knowledge-engineering tool distributed by IBM.

The Performance System

The performance system has two modes of operation. When it is given a file of test cases, it uses them for batch consultation. When it is not given this file, it does an interactive consultation. Only the batch consultation is required for the experiment; but the interactive consultation allowed the tool to be tested, and makes it general enough to be used outside the experimental environment.

The performance system is similar to MYCIN, but more modest in that it models a simple decision, not a complex one (ailment and treatment) as MYCIN does. It is possible, when the user answers "don't know" to some of the questions, for the system to arrive at more than one conclusion. These are all presented by the advice subfunction.

The performance system is an expert-system shell structured as shown in Figure 34 on page 166.

Subfunctions of the Performance System

Batch Consult: This subfunction will browse a test case base, taking each case as input to its inference engine. This function will determine whether the output of the inference engine matches the expected outcome stored on the test case file. The number of test cases processed and the number in which the inference agrees with the expected result are counted and passed to the elicitation tracer.

Interactive Consult: This subfunction asks the user a multiple-choice question to obtain a value for an attribute. The choices consist of all the values of the particular attribute, plus "don't know". The answer selected by the user is passed to the inference engine. The questions asked depend on the rules being evaluated. In general, not all questions are asked to arrive at a conclusion.

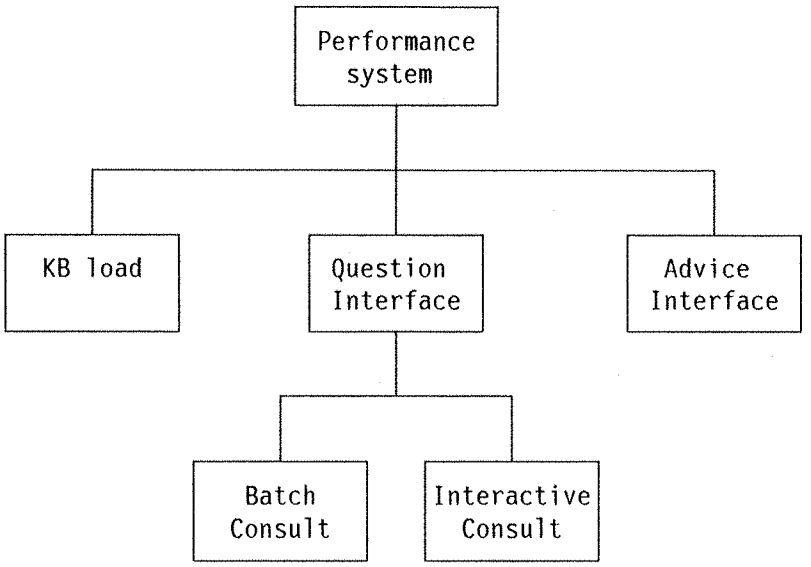

Figure 34. Structure of the performance system in SCENIC

Tracing

Throughout the elicitation process, the elicitation tracer will keep track of information such as the following:

- Date and time of event
- User identifier
- Acquisition method
- Stage of the process
- Start and end times
- Measures relevant to the stage.

This information will be used later to analyse the progress of the user through an elicitation session and to reflect the effort involved at each stage of the elicitation process. Tracing will be applied to both repertory grid technique and knowledge acquisition from examples.

Conclusions

It is possible to build a tool that administers the two knowledge-acquisition methods in the experiment, and collects the appropriate measures for later statistical analysis. Such a tool, SCENIC, described in this chapter, was constructed and tested to ensure that it functioned as intended.

Several people were asked to use SCENIC, and the comments they provided sometimes pointed to the need for minor adjustments in the user interface. All such adjustments were made. Much of the feedback was encouraging. This process continued for several months, until no further changes were being suggested.

During this testing period, the tool was used for several domains: business strategy, object identification, tool selection, identifying animals, and several small induction problems, e.g., the set of seven examples given in Hart (1986) mentioned on page 159.

SCENIC has proved adequate for its primary purpose, and includes general features (e.g., the ability to set the trace off, interactive consultation, and the generation of knowledge bases for export to other environments) that make it useful for eliciting knowledge and distilling efficient knowledge bases outside the experimental context.

The architecture used for SCENIC has also been found to be capable of extension to acquire knowledge for more complex analysis systems. This redesign is being done with the object-oriented methodology of Wirfs-Brock (1990). This approach should facilitate even further extensions, as they become desirable. In the redesign, more emphasis is being placed on domain definition, which the current version assumes complete. This will involve eliciting a skeletal plan or some skeletal advice, which will indicate what further knowledge needs to be elicited by the system.

Chapter 8. How Technique Affects Knowledge Acquisition: a Controlled Experiment

Abstract

The repertory grid technique and KAMSE follow analogous stages, and elicit essentially the same kind of knowledge. A comparison of the data elicited by the two methods, and consideration of the cognitive processes probably at work, gave rise to the expectations that neither of these two methods has more efficacy than the other. The repertory grid technique was expected to be more efficient at one stage of the knowledge-acquisition process, but it was thought that this difference would probably not be large enough to be reflected significantly in the efficiency of the entire process. This thesis has been elaborated into five hypotheses, which were tested by a controlled experiment. The two methods were shown to be equally effective ways of obtaining the information needed to build a heuristic knowledge base for classification. Differences in efficiency are attributed to possible mismatches between the repertory grid technique and the domain expert's cognitive system. These differences also point to opportunities for improving the efficiency of the repertory test.

Introduction

When a knowledge-based system (KBS) is delivered to its users and incorporated successfully into their routine operation, the system is the result of a sequence of actions in which prospective users, knowledge engineers, and domain experts participated. Such a system would normally be accepted for routine use only when it has been shown to act with a high degree of accuracy. This accuracy can indeed be improved by refining the knowledge base (Politakis, 1985; Ginsberg, 1988). But the size of the refinement task, and even

the necessity for refinement, can be reduced if the knowledge-acquisition process is itself capable of producing highly accurate knowledge bases.

Knowledge acquisition can be a difficult problem; and several methods have been used to try to solve it. Typically, much of the knowledge has to be elicited from a domain expert; and the choice of method can affect the pace and outcome of the process. Indeed, when a domain expert agrees to have his knowledge elicited, it is important, to maintain his or her enthusiasm, that this elicitation be as efficient and effective as possible. Such efficiency and efficacy are unlikely to be achieved unless the elicitation technique closely matches the cognitive processes of the domain expert.

The experiment designed and described in Chapter 6, "Design of an Experiment to Compare two Knowledge-Acquisition Techniques" was carried out as planned. This chapter describes how the experiment was conducted, and analyses the data collected.

The results indicate that, whereas the two methods produce equally accurate knowledge, the domain expert needs to expend more effort when the repertory grid technique is used. The increased effort is at two stages of the process. One of these stages is the elicitation of attributes or constructs, where shortcomings of the repertory grid technique are evident; but these are probably capable of being remedied so as to make the technique more efficient.

The hypotheses to be tested are discussed at length in Chapter 3, "Some Implications of Cognitive Psychology for Knowledge Acquisition."

Experiment Design

As discussed in Chapter 6, "Design of an Experiment to Compare two Knowledge-Acquisition Techniques" on page 123, the single independent variable in the hypotheses stated above is knowledge-acquisition method. Now, to test the hypotheses, a single-factor within-subject (repeated measures) design was used.

As has been argued in Chapter 5, "The Controlled Experiment in Knowledge-Acquisition Research" (see also Dhaliwal & Benbasat, 1990), factors other than method can affect the outcome of any knowledge-acquisition effort. So this design attempted to control for these moderating variables by randomly assigning the domain experts to method, eliminating knowledge engineers and replacing them with a tool whose behaviour is consistent across subjects, and using a single domain throughout to keep the domain characteristics constant.

One possible approach to testing the stated hypotheses was to use a repertory-grid knowledge-acquisition tool (e.g., Boose's AQUINAS) to acquire knowledge under the repertory grid technique, and to use another tool embodying KAMSE (e.g., 1st-Class, a commercially available expert-system shell distributed by Programs in Motion). However, neither of these tools measure and record the data needed to test the stated hypotheses. But perhaps more crucially, 1st-Class runs under the Disk Operating System (DOS) on a personal computer, while AQUINAS runs on the Xerox family of Lisp machines (Boose & Bradshaw, 1987). (There is also a subset that runs on a DEC Vax and a UNIX-based portable version.) So platform would have become a factor if tools like these were used.

There are, of course, repertory-grid knowledge-acquisition tools that run on personal computers (see, e.g., Garg-Janardan & Salvendy, 1988). But any two tools are markedly dissimilar in the way they look and feel. It would be

quite difficult to say whether any differences found in the dependent variables were due to knowledge-acquisition method or to subjects' differential reactions to the two dissimilar user interfaces.

Being a dual-function tool, SCENIC represents an attempt to eliminate platform as a factor. In addition, several routines are shared between the two methods implemented within SCENIC. For example, the acquisition of elements in the repertory grid uses the same routine as the acquisition of classes in KAMSE. The induction routine is also shared between methods, as is the routine for eliciting evaluation cases.

In both methods, similar keys are used for similar functions; and the same keys are used for analogous functions throughout all stages of knowledge acquisition. The same areas on the screen are used consistently for the same purpose; and colours are used in the same way throughout all the stages of knowledge acquisition, and therefore between methods. So considerable effort was invested in eliminating the user interface as a factor in the experiment. And although there has been no attempt to prove conclusively that interface is indeed constant as a factor between methods, a test has been included that gives some indication as to whether subjects found the two interfaces similar. This test is implicit in hypothesis 1.

Knowledge Domain

An experimental domain was created consisting of a package of eight objects that any university student or office worker or academic living in Britain would be expected to be very familiar with and be able to identify quite easily. Based on preliminary trials, the domain was designed so that the acquisition of the knowledge could be completed within an acceptable length of time.

Subjects

The experiment used 32 volunteer subjects (19 male and 13 female), who came from the following backgrounds:

- 11 undergraduate third-year students
- 9 information developers from a large computer company
- 6 members of the academic staff of Southampton University
- 6 masters students studying management and accounting.

The normal activities of all the subjects involved using computer terminals or personal computers routinely several times per week.

Experimental Method

Each subject was randomly assigned to both knowledge-acquisition methods from a Latin square of treatment combinations. This ensured that an equal number of subjects used each method at their first session. The design was an attempt both to counterbalance for practice effects, to minimise errors due to subject variability (Keppel, 1982), and to make as much use as possible of the available subjects. To try and attenuate any carry-over effects, each subject was allowed a period of at least one week (average 10 days) between the two sessions.

The experiment was conducted in a computer laboratory at the University of Southampton over a six-week period. The room was equipped with IBM Personal System/2 Model 55 computers running the Disk Operating System (DOS). Each subject was provided with a diskette and written instructions for the knowledge-acquisition method to be used. The instructions, which differ slightly between methods, are shown respectively in Appendix B, "Instructions for KAMSE" on page 216 and Appendix C, "Instructions for the Repertory Grid Technique" on page 220.

An important ingredient in the experiment was the knowledge-acquisition tool. The two methods have been implemented in the SCENIC knowledge-acquisition tool, which elicits knowledge directly from a domain expert. Most of the instructions were intended to help the subject become familiar with using SCENIC for an entire knowledge-acquisition process in a simple domain. The training was planned to take 10 to 15 minutes, and the actual experiment was expected to take up to 40 minutes.

The diskette given to a subject contained those portions of the software needed for the acquisition method to be used. After the training period, each subject was given a package containing the objects in the experimental domain and was asked to use the tool to elicit, record, analyse, and evaluate his or her own knowledge of the domain. At the end of the experiment, each diskette contained a representation of the knowledge acquired during the session and the measurement data gathered by the tool. The knowledge units elicited from a typical subject are shown in Appendix D.

Results

Although the experiment was designed to test the effect of knowledge acquisition method, it was still conceivable that differences observed in the dependent variables were affected by other factors as well. For the within-subjects (repeated measures) design, which was used in this experiment, Keppel (1982) argues that, with each subject receiving both treatments, it is possible that there was some practice effect: that, in spite of the precautions taken in the design, the position of a treatment (whether it was administered first or second) could have influenced the differences between group means. For example, the repertory grid technique might have a different effect when it is the first treatment than when it is the second.

In addition to these main effects, there could also be effects due to interactions. (Keppel, 1982, p 178, defines interaction as being "present when the effects of one independent variable on behavior change at different levels of the second independent variable". Keppel argues that, because the within-subjects design attempts to counterbalance any practice effects by randomly assigning treatment position to subject, it is possible to quantify the effect due to treatment position, thereby reducing the size of the error term.

Because several dependent variables were measured in the experiment, group means on all these variables had to be compared simultaneously. A multivariate analysis of variance (MANOVA) was therefore appropriate (Bray & Maxwell, 1985).

Five of the 32 subjects failed to complete their first trial. These five were all students who participated in the experiment between classes, and were unable to complete the evaluation stage of the process in the time they had available. Although all five did elicit enough knowledge to build a knowledge base, they did not generate test cases against which to evaluate these knowledge bases. None of these five subjects returned for a second trial. A further five subjects failed to return after completing their first trial. Most of these were students (3 out of 5); and it is not known why they did not come back. However, both of the other absentees later explained that they had had urgent business at the time arranged for their second trial.

Because of the missing cases mentioned above, the MANOVA lost some power by having to exclude data provided by subjects who did not do both trials.

```
* * ANALYSIS  OF  VARIANCE * *

EFFECT of METHOD
Multivariate Tests of Significance (S = 1, M = 1 1/2, N = 7 )

                                                          Sig.
Test            Value  Approx. F    Hyp. DF     Err DF   of F
Pillais        .48313    2.99117       5.00      16.00   .043
Hotellings     .93474    2.99117       5.00      16.00   .043
Wilks          .51687    2.99117       5.00      16.00   .043
Roys           .48313
```

Figure 35. Omnibus MANOVA tests of knowledge-acquisition method

As shown in Figure 35, the omnibus MANOVA indicated an overall significant effect of knowledge-acquisition method ($p < 0.05$) on the Pillai's, Hotelling's, and Wilks' tests. The significant overall MANOVA justifies a closer examination of the effect of method on each of the five dependent measures. Significant effects of method on some of these dependent variables were indicated by the univariate F-tests, and are discussed in the following sections.

Eliciting Elements and Classes

The mean number of minutes used to elicit elements under the repertory grid technique, and classes under KAMSE is shown in Figure 36 on page 176. Because these are means of the MANOVA cells, the measures from the ten subjects who did only one test are not included. This is also true of the means presented later for other dependent variables.

```
Summaries of   Minutes to acquire elements / classes
By levels of   KA method

                ---------- Mean ----------
                1st Test  2nd Test  Overall   Std Dev    Cases
   Repertory grid   2.68      2.21     2.45      0.64       22
   KAMSE            2.71      1.79     2.25      0.76       22

  Total Cases =      32
  Missing Cases =    10
```

Figure 36. Mean number of minutes to acquire elements and classes

The subjects who did both tests used a mean time of 2.45 minutes to acquire elements, and 2.25 minutes to acquire classes. The univariate F-tests of mean time to acquire elements / classes against method and treatment position are shown in Figure 37. A practice effect, present in both methods, is also evident.

```
   * * *  U N I V A R I A T E   F - T E S T S  * * *

        TIME1    Minutes to acquire elements / classes
   BY   METHOD   KA method
        POS      Treatment position

                          Hyp.  Hyp.   Err.            Signif
Source of Variation        SS    DF     DF       F      of F

Main Effects:
  METHOD                 .417    1     20      2.239   .150
  POS                   5.239    1     20     28.128   .000
  Constant            242.755    1     20    436.767   .000

Interactions:
  POS X METHOD           .557    1     20      1.002   .329

     32 Cases were processed.
     10 Cases were missing.
```

Figure 37. Univariate analysis of variance in time used to acquire elements or classes

This analysis shows a strong constant effect ($p < 0.001$) over the observed variations in time. There is also a strong main effect of treatment position, as would be expected from the means shown in Figure 36.. However, knowledge-acquisition method appears to have no significant effect on the variations in time used to acquire elements or classes. The strong constant effect reflects the fact that, whichever method is used, subjects use a certain amount of time without having any knowledge elicited.

This result supports hypothesis 1, and confirms that the two knowledge-acquisition method treatments cause subjects to respond in the same way to essentially identical stimuli.

Eliciting Constructs and Attributes

As Figure 38 shows, eliciting constructs under the repertory grid technique consumed more time (11.66 minutes) than eliciting attribute descriptors and values under KAMSE (9.34 minutes). In spite of the time allowed between treatments, differential carry-over effects seem to be present. The repertory grid technique appears to be more difficult when used after KAMSE than when the subject has no experience of knowledge acquisition.

Summaries of Minutes to acquire constructs / attributes
By levels of KA method

	Mean				
	1st Test	2nd Test	Overall	Std Dev	Cases
Repertory grid	10.53	12.79	11.66	3.55	22
KAMSE	10.93	7.74	9.34	3.24	22

Total Cases = 32
Missing Cases = 10

Figure 38. Mean number of minutes for each method to acquire constructs or attributes

Differences in the amount of time used at the attributes / constructs stage were analysed to determine the significance of effects exerted by method

and treatment position. A univariate analysis of variance on these variables indicated significant effects of method at this stage, as Figure 39 on page 178 shows.

```
* * *  U N I V A R I A T E   F - T E S T S  * * *

        TIME2    Minutes to acquire constructs / attributes
  BY    METHOD   KA method
        POS      Treatment position
```

Source of Variation	Hyp. SS	Hyp. DF	Err. DF	F	Signif of F
Main Effects:					
METHOD	59.617	1	20	8.065	.010
POS	2.341	1	20	.317	.580
Constant	4850.651	1	20	382.403	.000
Interactions:					
POS X METHOD	81.500	1	20	6.425	.020

```
    32 Cases were processed.
    10 Cases were missing.
```

Figure 39. Analysis of variance of time used to acquire constructs and attributes

This univariate analysis of variance indicates that both the knowledge-acquisition method and a constant affect time (and hence mental effort) significantly ($p < 0.05$). These main effects remain significant even when tempered by the Bonferroni procedure. The direction of the difference observed has already been seen in Figure 38 on page 177, which indicates that the repertory grid technique requires more effort than KAMSE at this stage. This result does not support hypothesis 2.

There is also a significant ($p < 0.05$) interaction effect between treatment position and method at this stage of the process, reflecting the differential carry-over effect mentioned earlier. Possible reasons are explored on page 188.

A feature of the repertory grid technique was its tendency to elicit negative constructs, e.g., "non-metallic / metallic". Whether a construct is negative or positive can affect later use of the knowledge base in which it appears. For instance, a person takes a finite amount of time to work out the meaning of "not non-metallic" when consulting with the knowledge base or classifying exemplars. Every such instance demands effortful thought (Cohen, 1989). The probable slowdown of the classification of exemplars would tend to increase the overall time used for the repertory grid technique. However, only a small, surprisingly negative, correlation (r = -.175) was found between the number of negative constructs used in a knowledge base and the total time used for knowledge acquisition.

Eliciting Ratings and Examples

As Figure 40 shows, subjects appeared to use less time to elicit ratings (7.54 minutes) than a minimal set of examples (8.29 minutes). A practice effect, affecting both methods, appears to be present and greater for KAMSE than for the repertory grid technique.

```
Summaries of   Minutes for ratings / examples
By levels of   KA method
```

	Mean				
	1st Test	2nd Test	Overall	Std Dev	Cases
Repertory grid	7.60	7.48	7.54	2.18	22
KAMSE	9.44	7.14	8.29	3.04	22

```
  Total Cases =      32
Missing Cases =      10
```

Figure 40. Mean number of minutes to acquire ratings or examples under each method

As for the hypotheses discussed above, it is again necessary to identify the main effects of method and treatment position, and any interaction between the method and treatment position, on the effort needed (and hence time used)

to elicit ratings or a minimal set of examples. These effects are shown in the univariate analysis of variance in Figure 41 on page 180.

```
* * *  U N I V A R I A T E   F - T E S T S  * * *

        TIME3    Minutes for ratings/examples
   BY   METHOD   KA method
        POS      Treatment position

                      Sum of   Hyp.   Err.          Signif
Source of Variation   Squares   DF     DF      F     of F

Main Effects:
  METHOD                6.150    1     20     1.514   .233
  POS                  16.061    1     20     3.954   .061
  Constant           2756.320    1     20   300.732   .000

Interactions:
  POS X METHOD         13.000    1     20   201.418   .248

    32 Cases were processed.
    10 Cases were missing.
```

Figure 41. Analysis of variance of time used to acquire ratings or examples

This analysis again indicates a strong constant effect ($p < .001$). There is a hint of a main effect of treatment position, but this is outside the .05 significe level ($p = 0.061$). The effect is probably an indication of the practice effect apparently present in the means. Although no systematic investigation of this was done, a few subjects did remark that the second session seemed easier than the first. However, knowledge acquisition method has no significant main effect. These results support hypothesis 3.

Efficacy of the Methods

As mentioned on page 129, when machine induction was completed, a set of exemplars was generated from random values of the attributes used in the knowledge base (see "Eliciting Test Cases" on page 151). These exemplars are descriptions of hypothetical objects, which the subject was asked to name. This

approach is simple, and feasible within the time constraints of an experiment. However, exemplars created in this way do not always describe real objects, when the complete descriptions are considered. The subject therefore identifies the object by considering one attribute at a time, in the same sequence as the performance system would request the information. The subject does not have to consider all the attributes, only enough of them to decide what the object is.

Accuracy, the main measure of efficacy, is computed as the number of evaluation cases in which the expert's classification was the same as that of the generated knowledge base, divided by the total number of evaluation cases processed. As Figure 42 indicates, the mean accuracy of the knowledge bases generated was 72% for those from repertory grids, and 77% for those under KAMSE. A practice effect again appears to be present.

Summaries of Knowledge-base accuracy
By levels of KA method

	Mean				
	1st Test	2nd Test	Overall	Std Dev	Cases
Repertory grid	0.67	0.78	0.723	0.23	22
KAMSE	0.76	0.78	0.774	0.17	22

Total Cases = 32
Missing Cases = 10

Figure 42. Mean accuracy of knowledge bases generated under each method

The significance of this difference and the presence of any main effects of method and treatment position were tested by the univariate analysis of variance of knowledge-base accuracy. This analysis is shown in Figure 43 on page 182.

```
* * *  U N I V A R I A T E   F - T E S T S  * * *

        ACCURACY Knowledge-base accuracy
  BY    METHOD   KA method
        POS      Treatment position
```

Source of Variation	Sum of Squares	Hyp. DF	Err. DF	F	Signif of F
Main Effects:					
METHOD	.029	1	20	.695	.414
POS	.043	1	20	1.038	.320
Constant	24.656	1	20	606.594	.000
Interactions:					
POS X METHOD	.020	1	20	.491	.491

Figure 43. Analysis of variance of knowledge-base accuracy

This analysis indicates a strong constant effect; but knowledge acquisition method does not have a significant main effect on knowledge-base accuracy. This result supports hypothesis 4: the knowledge bases generated by induction are equally accurate for both the repertory grid technique and KAMSE.

Efficiency of the Methods

The mental effort involved in the complete knowledge-acquisition process is made up of four components. The first three have been discussed in preceding sections. The fourth component is the effort used in classifying the evaluation exemplars. The mean time used, under each method, for the entire knowledge acquisition-process is shown in Figure 44 on page 183. A practice effect appears to be present.

Summaries of Minutes for entire KA process
By levels of KA method

	Mean				
	1st Test	2nd Test	Overall	Std Dev	Cases
Repertory grid	36.69	35.52	36.11	7.17	22
KAMSE	35.57	26.69	31.13	8.89	22

Total Cases = 32
Missing Cases = 10

Figure 44. Mean number of minutes to classify evaluation exemplars

Subjects using the repertory grid technique spent more time classifying evaluation exemplars. This contributed to the difference in total knowledge acquisition time between methods. Whether this difference is significant, and whether there are any significant main effects of method or treatment position, were determined by the univariate analysis of variance in Figure 45.

* * * U N I V A R I A T E F - T E S T S * * *

TTIME Minutes for entire KA process
BY METHOD KA method
POS Treatment position

Source of Variation	Hyp. SS	Hyp. DF	Err. DF	F	Signif of F
Main Effects:					
METHOD	272.422	1	20	5.871	.025
POS	278.260	1	20	5.997	.024
Constant	49729.138	1	20	727.470	.000
Interactions:					
POS X METHOD	163.497	1	20	2.392	.138

32 Cases were processed.
10 Cases were missing.

Figure 45. Analysis of variance of time used to classify evaluation exemplars

It is evident that there is a strong constant effect on the variations in total time used. Knowledge-acquisition method also has a significant main effect ($p < 0.05$). This result does not support hypothesis 5; the repertory grid technique requires more effort overall (and is therefore less efficient) than KAMSE.

There is also a significant main effect of treatment position. This reflects the practice effect evident in the means.

Figure 46 brings together the mean times for the four stages requiring mental effort, and one that does not. It is clear that, at the stages where knowledge acquisition method has a significant main effect on the time used, the repertory grid technique uses more time.

	Rep grid		KAMSE		Method as
	Mean	SD	Mean	SD	a Factor
Entire Process	36.1 100%	7.2	31.1 100%	8.9	Significant
Elements/ Classes	2.4 7%	0.6	2.3 7%	0.8	Not sig.
Constructs/ Attributes	11.7 32%	3.6	9.3 30%	3.2	Significant
Ratings/ Examples	7.5 21%	2.2	8.3 27%	3.0	Not sig.
Induction	.6 2%	.2	.5 2%	.3	-
Evaluation	13.9 39%	4.6	10.7 34%	4.5	Significant

Figure 46. Number of minutes used at different stages of each knowledge-acquisition method

Although time has generally been used as a proxy measure for mental effort, the induction stage of the knowledge acquisition process uses time but

demands no effort from the subject. Induction provides a short interval of about half a minute for the subject to relax before starting to classify the evaluation exemplars. This is not problematic because induction actually occupies only a very tiny fraction of the time taken by the entire process (34 minutes overall mean).

The fact that induction accounts for such a tiny (though important) portion of the time required to build a knowledge base from scratch casts fresh light on the way that the demonstration of Michalski & Chilausky (1980) should be interpreted. Induction is only the tip on an iceberg (see page 199).

Evaluation effort

The MANOVA is sensitive to the number of dependent variables, as this number provides the hypothesis degrees of freedom used to determine whether the omnibus test is significant (Bray & Maxwell, 1985). The experiment was designed to provide five dependent variables for the MANOVA. Introducing a sixth dependent variable could impair the significance of the omnibus MANOVA. So, although it is tempting to examine the significant effects on variables other than the five in the design, this can be done within the MANOVA only if each such additional variable is used as a substitute for one of the original five.

Evaluation time (TIME5) is an interesting variable, which was measured, but which was never intended to be used in the MANOVA. However, this variable is a genuine substitute for total knowledge-acquisition time (TTIME) because TTIME - TIME5 is equal to the total time used at the first four stages of knowledge acquisition. Moreover, when evaluation time is used instead of total time, the significance of the omnibus MANOVA and all the effects discussed so far in this chapter remain unchanged. Thus, conclusions

drawn from analysing TTIME could also be drawn indirectly from analysing TIME5.

Classifying evaluation examplars is the same process for both methods. Differences in the amount of time used might be due to one of two reasons:

- One method might tend to elicit a larger number of attributes, thereby giving the knowledge source more factors to consider in classifying an exemplar.
- One method might tend to elicit attribute descriptors and values that are not readily understood when combined (e.g., a negative descriptor such as "non-metallic" combined with the value "no").

The time used for classifying evaluation exemplars is interesting because a significant learning effect on it would further support the notion that knowledge sources can be primed.

The mean times used for classifying evaluation exemplars are shown in Figure 47, which appears to indicate a learning effect for both methods. It also appears that more time was used for the repertory grid technique than for KAMSE.

```
Summaries of   Minutes for evaluation
By levels of   KA method

                ---------- Mean ----------
                1st Test  2nd Test  Overall   Std Dev    Cases
  Repertory grid   15.36     12.47     13.91      4.56       22
  KAMSE            12.09      9.44     10.76      4.49       22

 Total Cases =      32
 Missing Cases =    10
```

Figure 47. Mean number of minutes to classify evaluation exemplars

The significance of any effects of method and treatment position were determined by the univariate F-tests summarised in Figure 48 on page 187. The analysis shows a strong constant effect ($p < .001$). Method also has a significant main effect ($p < .05$): KAMSE demands less effort than the repertory grid technique. However, the main effect of treatment position is marginally outside the .05 significance level.

```
* * *  U N I V A R I A T E   F - T E S T S  * * *

          TIME5    Minutes for evaluation
    BY    METHOD   KA method
          POS      Treatment position

                          Hyp.  Hyp.    Err.              Signif
Source of Variation        SS    DF      DF       F       of F

Main Effects:
  METHOD                109.200     1      20     5.248   .033
  POS                    84.153     1      20     4.044   .058
  Constant             6698.234     1      20   373.229   .000

Interactions:
  POS X METHOD            .148     1      20      .008   .929

     32 Cases were processed.
     10 Cases were missing.
```

Figure 48. Univariate analysis of variance in time to classify evaluation exemplars

Conclusions

As Figure 49 on page 188 shows, three of the five hypotheses are supported by the data. The two knowledge-acquisition stages at which the hypotheses are not supported require more effort to use the repertory grid technique than KAMSE.

Hypothesis	Supported?	Contrary Finding
1	Yes	-
2	No	Repertory grid requires more effort to acquire an adequate domain model.
3	Yes	-
4	Yes	-
5	No	Repertory grid requires more effort for the complete KA process.

Figure 49. Summary of support for the hypotheses

A feature of the repertory grid technique is its tendency to elicit negative constructs. A tool could counter this tendency by switching poles, so that the positive one is always primary. Alternatively, the question "what makes A similar to B and different from C" could be turned around so that a positive trait is always elicited first. Neither of these devices was used in SCENIC, although the integrity of the repertory grid technique seems better preserved by letting the tool detect negativity (i.e., it does not involve the subject, only the method administrator).

Although the interaction effect of treatment position with method is significant in only one of the dependent measures, there are significant main effects of treatment position. These affect the tests of hypotheses 1 and 5. A practice effect is observable in all the means, with one exception.

Knowledge sources can be primed. Those who have their knowledge elicited once tend to find a second experience of having the same knowledge elicited less demanding, even if two different methods are used. All stages of KAMSE appear to be affected by this practice effect when the knowledge

source has previously had the same knowledge elicited by the repertory grid technique.

However, knowledge sources can also be inhibited by a previous experience of having their knowledge elicited. This appears to affect triadic elicitation of constructs when the knowledge source has previously been asked directly for attributes involved in the decision being modelled. The time taken to elicit constructs is less when the repertory grid technique is used first (10.5 minutes) than when it is used second (12.8 minutes). This effect needs further investigation if it is to be explained satisfactorily. However, two possible reasons for it suggest themselves.

Subjects undergoing triadic elicitation may have been distracted by the contrast between the freedom of attribute elicitation under KAMSE and the restrictive nature of the triads. If this is so, then knowledge engineers may be well advised to find out whether a prospective knowledge source has had previous experience of knowledge elicitation. On the other hand, using the repertory grid technique may be an effective way of preparing knowledge sources to have their knowledge elicited by another method.

One criticism directed at the two methods discussed here is that they work well for simple problems, but break down when the problem assumes real-world complexity (see, e.g., Gammack, 1987; Quinlan, 1991). But the simple classification problem can be a building block for more complex problems such as planning (see, e.g., Dechter & Michie, 1984). Complex problems often involve a series of such decisions, some depending on the results of others. Garg-Janardan & Salvendy (1988) demonstrated that hierarchies of classifiers can deal with some kinds of complexity, and the repertory grid technique can be used to elicit the knowledge for each classifier.

The knowledge-acquisition methods used here can therefore be quite powerful when used to develop parts of complex knowledge bases. In such

environments, inefficiencies are multiplied; and the total effort to be saved in developing a complex knowledge base could be considerable. The acquisition of knowledge from multiple experts also involves multiplying inefficiencies; so the same considerations apply. The findings of this experiment provide some pointers to both improving and measuring the efficiency of knowledge acquisition.

Chapter 9. Discussion of the Results

Abstract

The data collected in the controlled experiment are reported and analysed in Chapter 8. This current chapter discusses the theories underlying the research, what is assumed to be true, and what was tested. The implications of the results for each hypothesis are also discussed. These include the successful control of the tool's user interface as a factor, the efficacy of triadic elicitation, and ways of improving its efficiency. Implications of the fact that the two methods have equal efficacy are also discussed. Two of the dependent measures are affected by the classifying of exemplars, which in turn seems capable of being affected by knowledge-acquisition method. Factors affecting the internal and external validity of the experiment are also discussed.

Introduction

In many advanced areas of science, a general theory combines with a specific one to produce an explanation of observed reality Bunge (1973). For example, the general theory of simple harmonic motion is applied to both the swinging of a pendulum and the flow of current in an electric circuit. But the specific theory in each case is different: in the case of the pendulum, the specific theory has to do with the length of the string and the acceleration due to gravity, whereas in the case of the electric circuit the specific theory has to do with the voltage of the power supply and the impedances in the circuit. Bunge argues strongly that specific theories can be tested, while general ones cannot, because general theories are nothing more than pure mathematics; whereas specific ones are expressed in terms of the reality that they are meant to model and explain.

Perhaps, then, it would be useful to identify specific and general theories underlying the hypotheses tested in the experiment. One general theory, on which hypothesis 3 is based, is mathematical in nature. The declarative

knowledge underlying a cognitive skill may be thought of as being represented by a matrix. One method traverses this matrix in one direction (horizontally by rows), while the other traverses it vertically by columns. If it is the same matrix that is being traversed in the two methods, then each would end up traversing the same total distance, and hence taking the same amount of time.

Another general theory is that memory is divided up in a certain way: long-term versus short-term, and that long-term memory has episodic, semantic, and procedural elements that are differentially retrievable. Short-term memory has a very restricted capacity. A related theory is that the knowledge underlying a cognitive skill is represented as productions in procedural memory, it is not amenable to conscious retrieval, but it is an experiential distillation of declarative and episodic knowledge from semantic and episodic memory respectively.

This last theory implies that, in knowledge acquisition, what is actually accessed is not the irretrievable compiled productions, but the episodes and schemata, which are retrievable. Processes are available to distill these and generate productions, presumably no less powerful than those stored in the expert's procedural memory. Although the experiment might support it, no attempt has been made to test this general theory. Testing it would probably have required an experimental condition under which the experts stated explicit rules. So the general theories have been assumed to be true. However, the research does test aspects of a specific theory; that is about what happens when the expert is:

- Asked to list elements
- Given the repertory test
- Asked to list attributes descriptors and values adequate for describing a set of examples of all the domain classes

- Asked to express a minimal set of examples in terms of attribute values
- Asked to provide evaluation cases by classifying exemplars.

Implications of the Results

This section discusses the findings related to each hypothesis, and what they suggest for both using and further researching the two knowledge-acquisition methods.

Regarding Hypothesis 1

In some ways, hypothesis 1 is a self-evident truth; but stating and testing it are invaluable as a demonstration that what is expected to happen actually does. Both methods were expected to demand the same amount of effort because the method is merely a label; the process is exactly the same for both methods at stage 1 of knowledge acquisition. Use of a dual-function tool like SCENIC is an attempt to control for the interface between the tool and the user, i.e., to keep it unchanged between methods. The fact that the data support this hypothesis gives some indication that the attempt to control for user interface was substantially successful.

Regarding Hypothesis 2

An adequate domain model is only part of what needs to be acquired in knowledge acquisition. It comprises a list of elements (or classes), coupled with a set of constructs (or attribute descriptors and values). The elements define the limits of the domain and the constructs provide a language in terms of which either the domain elements can be rated or a set of examples can be expressed. The repertory grid technique includes a stage called the repertory test, which goes to some lengths to discover constructs which the subject is assumed to have difficulty identifying and articulating when asked directly.

Perhaps that was true of Kelly's patients, and of anyone being psychoanalysed. Several researchers in knowledge acquisition have assumed[7] that it is also true of domain experts. Kelly (1955) argued strongly and persuasively in favour of the triad; and hypothesis 2 is stated on the strength of this. However, the data do not confirm hypothesis 2. And although there is little doubt (from hypothesis 4, discussed later) about the efficacy of the triad, there is still some doubt about its efficiency. Even so, the experiment has not proved that the triad is not optimal; it may or may not be. The experiment does however, raise questions about the ability of the repertory grid technique to come to a speedy completion. Future research will have to test refinements to the mechanism used to elicit distinctions between confused pairs of elements during the closing stages of a repertory test.

One shortcoming of the repertory grid technique is that the person being interviewed does not always understand what the technique or the questions are trying to get at. This is especially crucial in the later stages of the repertory test (acquisition of constructs and ratings), when a small number of confused pairs of elements remain. A "why" option could help the user understand what is required. For example, the user's "why" could evoke a response such as this: "I already know the difference between (paper clip) and the other two elements. What I really need to find out is how (cent coin) differs from (penny coin)."

Alternatively, the tool could make its own selection of the odd one out in the triad in such a way that the user would have to supply the required distinctions. Instead of using strategies of this kind, the classic repertory test

---

[7] Latta & Swigger (1992) have tested whether the technique "accurately elicits and represents commonality of understanding about communal knowledge", and validated this assumption.

proceeds relentlessly, probably getting the naive user frustrated at being repeatedly confronted with the same (or similar) triads. Other researchers (e.g, Garg-Janardan & Salvendy, 1988) seem to have encountered this same problem. Some tools therefore try to solve the problem by switching from triadic to dyadic elicitation after a certain number of triads have been presented.

The word "dyadic" actually has three meanings when used with reference to the repertory grid. Ryle & Lunghi (1970) argued that the standard technique is insensitive to information about the constructs as applied to relationships between individual elements; to remedy this, they used the relationships as elements. Keen & Bell (1981) developed a dyadic method of eliciting elements and constructs at the same time by considering two elements at a time in a variant of the repertory test. Garg-Janardan & Salvendy (1988) were mainly concerned with speeding the closure of the repertory test by using two elements instead of three in the closing stages of the test.

Knowledge acquisition from a minimal set of examples has no way of limiting the domain model to a sufficient one free of redundant attributes. But then, neither does the repertory grid technique guard against this in a foolproof way. The redundancies are removed from the generated knowledge base by machine induction, but eliciting them in the first place is largely a wasted effort. In addition, while the repertory grid technique is seeking specific types of answers to specific types of questions, it may well, like a cross-examining lawyer, be inhibiting the user's freedom to express some of the relevant ideas that it evokes. Knowledge acquisition from a minimal set of examples suffers rather less from this.

It also appears that the kind of prompting provided by the repertory grid technique does not always make the production of an adequate domain model easier than when subjects are asked directly for the model components. It is evident that, in some circumstances, a person can come up with an

adequate domain model more easily if not constrained by the repertory grid technique.

Further research could determine efficient alternatives to triadic elicitation. It is quite likely that the repertory grid technique can be modified in various ways to improve its efficiency. For example, in the later stages of a repertory test, the tool could, as discussed above, construct triads and split them itself in ways that would constrain users to provide the required information. Experiments could determine whether any improvements in efficiency are produced by the modified method.

Although subjects generally benefitted from having had their knowledge elicited previously, the elicitation of constructs appeared to be more difficult after exposure to attribute elicitation under KAMSE. There is some question as to whether triadic elicitation inhibits the knowledge source (see page 77). The differential carry-over effect at stage 2 indicates that this may be so when a less restricting method has been used previously.

However, this effect needs to be investigated further. This could be done by comparing the efficiencies of a number of possible schemes to that of standard triadic elicitation:

- Dyadic elicitation of constructs
- Triadic elicitation of constructs
- Tetradic elicitation of constructs
- Letting the knowledge source compose the triads
- Continuously showing the knowledge source a model of the knowledge so far elicited
- Mixing the initiative by enabling the knowledge source to escape from the triad, and express any other constructs that come to mind out of sequence.

Regarding Hypothesis 3

The lack of a significant effect of method on the differences between the time taken to elicit ratings in the repertory grid technique and that taken to elicit a minimal set of examples indicates that the effort used in traversing an inference matrix by rows is no different from that used in traversing it by columns. The data therefore support hypothesis 3; but this does not mean that the knowledge is indeed organised as a matrix in the expert's mind, because the matrix is simply a convenient mapping of a multi-dimensional problem or domain space. All that can be said is that, however the knowledge is represented in the expert's mind, it can be mapped into a matrix. But this mapping is only an approximation of the actual knowledge (Newell, 1982). Moreover, the knowledge may be represented differently in the expert's mind, depending on the use to which it is being put (Tulving, 1962).

Regarding Hypothesis 4

The experiment set out to acquire declarative knowledge and to distill that into accurate productions. The hypotheses predicted that these productions would be equally accurate between the two methods used, because they acquire very similar kinds of knowledge that are put through the same process of distillation. And the empirical data support this prediction, which is embodied in hypothesis 4.

But why should the two methods produce knowledge bases that are equally accurate; is the similarity of the knowledge they elicit sufficient reason for expecting them to have equal efficacy? If that reason is indeed sufficient, then it is possible to make the general statement that any two methods that elicit similar kinds of knowledge ought to have equal efficacy. And what is so similar about the knowledge? It has been shown (in Chapter 2) that there is a

one-to-one mapping between the knowledge units elicited under the two methods.

Perhaps, then, it is possible to generalise by saying that if there is a one-to-one mapping between the knowledge units acquired by two methods, then the two methods are likely to have equal efficacy provided that the knowledge under each method is subject to the same distillation processes. That seems reasonable, because what has actually been done with the repertory grid is to use it to generate a minimal set of examples, which have then been processed by machine induction to come up with a knowledge base in the same way as under the KAMSE. However, this conjecture has not been proved; the only assertion proved is that the two methods used in the experiment produce knowledge bases that have equal efficacy. To prove the more general statement, it would be necessary to experiment further with an adequate random sample of all pairs of knowledge-acquisition methods related in this way (Anderson, 1971).

On the basis of this result, we cannot advise anyone seeking differential efficacy to choose either method in favour of the other.

Regarding Hypothesis 5

Although the repertory grid technique has been shown (by support for hypothesis 4) to acquire equally accurate knowledge as KAMSE, the two methods have also been shown to differ in efficiency. Two components of total effort appear to be responsible for the repertory grid technique's inferior efficiency as revealed by the test of hypothesis 5. First, as discussed under hypothesis 2, it is easier to propose attribute descriptors and values than to think of constructs. Second, it appears to be easier to classify exemplars for evaluation of knowledge induced from a minimal set of examples.

Evaluation exemplars are more difficult to deal with under the repertory grid technique because, in general, this technique yields more attributes than KAMSE. In addition, negative constructs can slow down the recognition of an exemplar's description. These are not final flaws of the repertory grid technique; it can probably be remedied by spending more time on the clustering of constructs, and by inverting negative constructs.

Previous Results

Michalski & Chilausky (1980) demonstrated convincingly the superior efficiency and efficacy of machine induction as compared to the "hand-crafting" of rules. This research has not sought to test those findings, the essence of which is supported by the author's experience of acquiring knowledge and building knowledge bases early in this research. Both methods used in this experiment have therefore included machine induction as a step in the process.

But it is clear that machine induction by itself cannot be said to be a knowledge-acquisition method. A knowledge-acquisition method would involve gathering case data by some means, coding these cases in terms of some domain model, processing them by machine induction, and evaluating the resulting rules. Indeed, one of the shortcomings of the work of Michalski & Chilausky is that they compared the entire knowledge-acquisition effort in one method with only a part of the effort in the other method. Even so, it is likely that this weakness in internal validity affected the magnitude rather than direction of their findings.

Internal Validity

Several factors can affect the outcome of any attempt to acquire knowledge. This experiment controlled for knowledge engineer and domain by keeping them constant, randomly assigned experts to method, and used method as the single independent variable. As discussed above, it is also likely that two tools

embodying the two methods would have very different user interfaces and therefore different effects on the user's ability to work with them. The single tool used in the experiment had a consistent interface throughout, thereby minimising the effect of user interface as a factor.

Originally, it was supposed that time used and number of keystrokes would be alternative indicators of mental effort. Analysis of the keystrokes data, which have not been included, threw considerable doubt on the validity of this assumption. There is, for example, a significant main effect of method on the number of keystrokes (but not the amount of time) used to acquire ratings and examples. People appear to spend time thinking about what to key in rather that pressing keys until they are satisfied with what they have put in. What the number of keystrokes might more plausibly indicate is the keying economy that the repertory grid technique has over KAMSE at the ratings / examples stage of knowledge acquisition. It therefore seemed appropriate to abandon the idea of using number of keystrokes as a measure of effort. This is not problematic because there are strong arguments in favour of using time as a measure of mental effort (see page 126).

It can be argued that the subjects were at a disadvantage in classifying the evaluation exemplars. The exemplars were generated by combining random values of the attributes. To make sense of these exemplars, the subjects had to view them in a certain way. Other ways of looking at the examplars might result in different classifications. The researcher's *ad-hoc* conversations with the subjects led him to believe that some of them had difficulty in adhering to the prescribed strategy when deciding on classifications for the exemplars.

In addition, wide variations in the accuracy measured (lowest score 9.4%, highest score 100%) indicated that some subjects grasped more fully than others how to consider the information present in an exemplar and decide on the correct class. In some ways, deciding the exemplar classes was an exercise

in predicting what the knowledge-based system would conclude if information was presented to it in a given sequence. But there is no basis for supposing that these difficulties affected one method differently from the other.

External Validity

It is possible to boost the external validity of an experiment by varying the problem. Employing that approach in this experiment would have called for using two domains rather than one, to answer the objections of critics who might question whether similar results would be obtained for some domain other than the experimental one. However, according to Keppel (1982, p 340), "the resultant increase in external validity is really not very great in any far-reaching sense". It is a device which has not been adopted in this experiment. This is mainly because, even with an extensive search, it was difficult to find a second domain of comparable simplicity and size, in which the same subjects could be expected to have the same degree of expertise. It would be nice to test these hypotheses again with a different domain. But this is a question of content versus structure; and there is no reason to think that domains of similar structure will not yield the same results.

There may also be some question about the applicability of these results to large-scale problems. It is difficult to make categorical predictions about scaling-up, but it is worth pointing out that large-scale problems can often be structured into hierarchies of small-scale ones (see page 189). Even under the KADS methodology, there are phases (Wielinga, Schreiber, & Breuker, 1992) at which using the repertory grid technique and inductive methods is recommended. Whether these are the only stages where these techniques are useful is open to question. However, the repertory grid technique and KAMSE are evidently applicable to both the decision modelling paradigm and the physical-system modelling one.

Chapter 10. Overall Conclusions

Abstract

This chapter concludes the research by trying to assess its significance and limitations. The first section summarises the lessons learned during the research. The second section discusses the limitations of what was done, ways in which the research could be improved or made more complete, and ideas from related research that are not covered by this research. The final section discusses the importance of this work to the development of the knowledge-acquisition discipline.

Lessons from the Research

Knowledge acquisition is primarily a process of gathering, analysing, and organising enough information about a given knowledge domain to build a knowledge-based system that has accuracy adequate for its purpose. Of course, the system developed will need to "speak the users' language" and be able to provide information of a quality and in a style that the users will find acceptable and useful. But without sufficient coverage and adequate accuracy, the system is unlikely to be useful enough to be unleashed on users.

If there is any reality in the knowledge-acquisition bottleneck which so many writers mention, then it seems reasonable to expect a project-management approach to yield useful results by clever sequencing of tasks and allocation of resources. Three factors seem particularly amenable to this approach: the position of knowledge acquisition in the network of tasks involved in building a knowledge-based system, the large amount of knowledge typically required to build a useful system, and the inherently slow nature of the knowledge-acquistion process.

This research does not fundamentally alter these views. It still appears that progress can be, and is being, made by repositioning knowledge acquisition

within the development cycle. But if knowledge acquisition is the entire process of building a knowledge-based system, as Buchanan *et al* (1983) define it, then what else of significance is there against which to reposition in the development cycle? One answer lies first of all in the fact that a growing number of knowledge-based systems appear to derive much of their usefulness from the fact that they do not stand alone, but interface with conventional software systems.

In developing this kind of integrated system, it is possible for the development of both the knowledge-based component and the conventional component to proceed in parallel, if tasks are sequenced appropriately. Another fruitful approach might be to subdivide knowledge-based systems into hierarchies of smaller ones and assign these to increased knowledge-engineering resources, which can proceed in parallel with different parts of the problem. Still another approach is to use rapid prototyping in an iterative cycle of elicit-build-test. Subsequent tasks can be driven by the evolving prototype.

Regarding the large amounts of knowledge typically required to build useful systems, it still appears reasonable to think that a fruitful approach is to select problem domains carefully (a view also expressed by Frederick Hayes-Roth in an interview with Chandrasekaran, 1991). But although this approach is likely to preserve the reputation of knowledge engineers adopting it, it provides little comfort for owners of problems eschewed as too messy, unwieldy, or unstructured. Wilkins' (1987) notion of "sociopathic knowledge" also appears reasonable; and the need for inductive distillation of knowledge, as implemented in SCENIC and some other knowledge-acquisition tools, supports this view. The challenge remains how to determine when enough knowledge has been acquired; and the approach taken by SCENIC is clearly one of the ways of doing this.

A number of avenues still appear to be available for tackling the inherently slow nature of the knowledge-acquisition process. Agarwal & Tanniru (1990), for example, have shown that training knowledge engineers in the use of an effective technique can improve the productivity of even those with little experience. More fundamentally, techniques are more likely to be successful if they recognise the limitations of the domain expert's cognitive system, and seek to elicit and use knowledge that is indeed retrievable (declarative knowledge) rather than try to tease out procedural knowledge which is not amenable to conscious retrieval.

Moreover, knowledge-acquisition technology is still evolving; and some builders of knowledge-acquisition tools argue that the labour-intensive process of knowledge engineering is unnatural and goes against all the trends in computing (see, e.g., Shaw & Gaines, 1987). One solution to speeding up the process is to create tools capable of bringing about dramatic increases in knowledge-acquisition productivity. As a by-product, this research has demonstrated that people can walk up to a well designed knowledge-acquisition tool, and with little training use it to elicit their own knowledge of a simple domain. It has also shown that small knowledge-based systems can be built and tested very quickly when some kinds of tools are used.

While the project-management approach is important, it is also evident that, at a deeper level, explanations of why one knowledge-acquisition method might perform differently from another lie in the ways knowledge is represented in the human mind and the ways in which it can be retrieved. There are too many knowledge-acquistion techniques for any research of this nature to cover them all. So the work has concentrated on two techniques which have not been compared in this way before.

In particular, this research has demonstrated that the two methods can be subdivided into stages characterised by the type of knowledge unit being

elicited. This is not an innovation; it is used in other tools. But little has been written about the expert's cognitive processes during these stages. It is evident that different mental operations take place in the domain expert's mind during these different stages. But, at every stage, the limited capacity of short-term memory restricts the number of elements or constructs that can be considered simultaneously. Techniques that ignore these restrictions may demand more effort from domain experts.

Knowledge acquisition is also the creation of a continuing problem, because any knowledge-based system that is put into regular use will probably need to have its knowledge base updated as time passes and domain knowledge evolves. So knowledge acquisition should have an eye on maintenance (see, e.g., Davis, 1990; Xiafeng, 1991). It seems reasonable to suppose that knowledge-based systems will be easier to maintain if they can be modified at the knowledge level rather than the symbol level (see Newell, 1982, for an explanation of these levels).

A knowledge-acquisition tool like SCENIC appears to be a feasible way of developing a maintainable knowledge base, because it converses with the domain expert at the knowledge level. It generates a knowledge base, which can be used directly by a performance system. One important feature of using a tool like SCENIC is that subsequent changes to the knowledge can be made at the knowledge level. A new knowledge base can be generated from the updated knowledge. Careful design of the data interfaces between a knowledge-based system and the traditional systems with which it is integrated is needed to support this kind of maintainability.

But, beyond the question of maintainability, consideration of the domain expert's cognitive system gives rise to the kind of insights useful for refining knowledge-acquisition methods. This research indicates that there is room for refinement of the repertory test to increase its efficiency.

It is also evident that there are grave obstacles facing any attempt to access the compiled productions that govern a domain expert's performance. However, it is possible to retrieve declarative knowledge without undue difficulty. This retrieved knowledge can then be subjected to processes that can generate powerful, accurate productions. In the same way as a novice can generate effective primitive productions from declarative knowledge, meta-knowledge embodied in a knowledge-acquisition tool can operate on declarative knowledge to fashion productions that exhibit expert performance. If we cannot retrieve the compiled productions on which an expert's performance is based, then we have to settle for retrieving the episodic and semantic declarative knowledge that generated the compiled productions.

Protocol analysis is one method that seeks to access the compiled productions in the only way they are accessible (by performance of the skill), and to accompany that performance with the expert's account, or even *post-hoc* justifications of what s/he is attending to. But protocol analysis has been shown to be inefficient compared to other methods. And there is some doubt about which memory the knowledge acquired by this method comes from.

Limitations of the Research

Although a very large portion of the knowledge-acquisition process is covered by the experiment and the tool, there is a little more to the process than the stages in SCENIC. When a user starts using the tool, it is assumed that s/he has already thought out the objectives of the knowledge base s/he would like to build. This process of domain definition is important because it is the first hurdle. As discussed on page 167, future work on the tool will include enhancements in this area. There is also room for acquiring explanations, and the tool largely ignores this. It could be a subsequent stage after enough information has been obtained to build an accurate knowledge-based system.

The attribute values accepted by the tool are all qualitative. There is room for quantitative attribute values, if the tool is to become applicable to a wider range of problems. It is also unclear how the efficiency of the two methods would be affected if they were applied to a domain involving quantitative attribute values.

One limitation of the kind of distillation done by machine induction has been mentioned in Chapter 2. It sheds attributes that are redundant when deciding among the classes identified. But being redundant may not be sufficient reason to discard an attribute totally. Whitehall (1990) point out that the attribute shed might be more important than the parallel one used. Buchanan *et al* (1983) also argue that redundancy should be built into knowledge bases, so that they do not rely heavily on a single piece of evidence. Because the ID3 algorithm ignores this advice, a user can obtain the desired redundancy only by including training examples that have alternative patterns of attribute values. The algorithm itself could probably save users this burden, by finding disjunctive relationships and including them in the decision trees or rules generated.

Discarding attributes is clearly a problem for both the repertory grid technique and KAMSE. Until induction algorithms are improved, some expert evaluation is appropriate, to determine when the knowledge base lacks sufficient redundancy. Without such safeguards, the resulting knowledge base is most usefully applied in a closed world, where it is known that the object to be classified does indeed belong to one of the classes included in the knowledge base.[8] In domains where this is not so, there is some chance of the knowledge

---

[8] Gammack (1987) has also criticised the repertory grid technique's focus on distinctions rather than similarities.

base reaching erroneous conclusions (false positives). There are solutions to this: one of them is to restrict the application of this kind of method to closed-world domains, for example, selecting the advertising or promotion medium for a product (Chadha, Mazlack, & Pick, 1991). Another approach may be to augment the set of classes by including others outside the narrow domain that the system is meant to cover. These outsider classes should have some characteristics in common with the classes that properly belong in the domain.

It is also evident that a kind of distillation similar to that performed by machine induction can be achieved by an artificial neural network. One obvious difference is that, whereas a neural network creates a mapping between the inputs (attribute values) and outputs (classes), it is difficult to derive explanations from this mapping. That is not so when machine induction is used.

There is a growing call for users to be involved during (rather than after) the knowledge-acquisition process. This will allow them to keep an eye on the objectives of the system and the words used to express the knowledge. There are good reasons for adopting this approach aimed at increasing the chances of a successful outcome to the process. The research being reported here has thrown no light on whether user involvement affects the pace of knowledge acquisition. There do appear to be stages at which users can be usefully involved. If rapid prototyping is used, users can generate valuable feedback on the emerging knowledge base.

In such situations, the knowledge engineer is a middleman between the user and the knowledge source. But the primary difficulty in knowledge acquisition is in obtaining enough (not too much, and not extraneous) information from the knowledge source to build an accurate knowledge-based system.

Value of the Research

The contribution of this research to the discipline of knowledge acquisition is summarised in Figure 50 on page 210.

The repertory grid technique and knowledge acquisition from a minimal set of examples have been shown to be similar in many respects and to acquire similar kinds of knowledge. Although these two methods were not expected to be differentially efficacious or efficient over the entire knowledge-acquisition process, the repertory grid technique was expected to be more efficient at one stage; but it was thought that this difference would probably not be large enough to be reflected in the effort used for the complete process.

The five hypotheses into which this thesis has been elaborated have been tested by a controlled experiment. The two methods were shown to be equally effective ways of obtaining the information needed to build a knowledge-based system for classification. Differences in efficiency suggest opportunities for improving the repertory grid technique.

The focal theory behind this research includes the mapping between the two knowledge acquisition methods. It also includes the set of hypotheses that were tested empirically. The contribution of this research to the focal theory is partly in demonstrating that it is, at least sometimes, easier for a domain expert to articulate a partial domain model and a minimal set of examples than to have his or her knowledge elicited by the repertory grid technique. The results obtained have raised questions about the optimality of triadic elicitation, especially in the closing stages of the repertory test.

This research has a bearing on Lundell's (1988, p 27) principle of "elicitational congruence", which states that "the knowledge acquired from the expert will be more accurate when the acquisition method is congruent with the cognitive system of the expert". While there is clearly a connection between the

- Pioneering the use of controlled experiments with a knowledge-acquisition tool (Shows some of the issues to be faced, the measures that can be used, and ways of recording them.)
- First controlled experiment to compare the repertory grid technique and knowledge acquisition from a minimal set of examples (Many methods need to be compared in the future; this research shows one way of doing these comparisons.)
- Identifies areas where the efficiency of the repertory grid technique needs to be improved
- Makes explicit the role of machine induction in the repertory grid technique
- Explores links between cognitive processes and knowledge acquisition.

Figure 50. Contribution of the research

outcome of knowledge acquisition and the method used, it appears to be the efficiency of the method, rather than the accuracy of the knowledge acquired, that is affected — at least in the two methods used here. Based on the results of this research, it is appropriate to modify the principle to state that the process of knowledge acquisition will require less effort *to acquire knowledge of a given accuracy* "when the method is congruent with the cognitive system of the expert".

Measuring the outcome of knowledge acquisition can be difficult, especially when the acquisition is not carried through to completion (Burton *et al*, 1990). This research demonstrates that knowledge bases can be generated and tested as part of the experimental treatments, if an appropriate tool is used, and exemplars can be used for evaluating a knowledge base.

If knowledge acquisition is to be done in industrial settings, it must also be a managed task. The resources to be used in it must be capable of being estimated in advance: the domain expert's time, the knowledge engineer's time, and the user's time. As a by-product, the results of this research can help provide pointers to selecting a technique and estimating the productivity of two methods in one type of domain.

Results reported by other researchers (e.g., Burton, Shadbolt, Rugg, & Hedgecock, 1990; Agarwal & Tanniru, 1990; Rugg *et al*, 1992) provide other pointers to the effort involved in using various methods for knowledge acquisition. These methods, which include protocol analysis, laddered grid, card sorts, and structured interviews, use manual methods to transcribe and analyse the information gathered. While it is not expected that these laboratory settings will necessarily provide industry with accurate means of estimating and planning, there are few other pointers available at the moment.

A growing number of tools have been built and reported on (see, e.g., Boose, 1989). The vast majority of them specialise in knowledge acquisition for analysis applications. Both the tools themselves and the approaches that underlie them are deserving of empirical comparison to identify how and why they differ. One important contribution of this research is to point the way that this kind of comparison can be done. Another is to demonstrate the differences between the two techniques focused on.

Although the research reported here seeks to establish the relative efficacies and efficiencies of two different methods, some writers argue that it is not appropriate to hold up a single method as being superior to others. In this view, it is more useful to cultivate a portfolio of methods, the different products of which will presumably complement each other. The KADS methodology suggests different methods at different stages of the process. Outside of a methodology, using a mix of methods can be a shotgun approach containing

appreciable redundancy and duplicated effort. At the same time, it seems quite reasonable to argue that, if the use of one method does not successfully elicit the desired knowledge, it might be appropriate to try other methods until one is found to be successful. The portfolio approach is based on an analogy with stock markets. The analogy is complete only if the relative efficacies and efficiencies of different methods are known.

The use of controlled experiments in knowledge acquisition research is not yet widespread. But the large number of approaches to knowledge acquisition, coupled with the scarcity of evidence for choosing among them, are an ideal situation for this kind of research. Research like that being reported here helps to erect markers on a barren landscape. Future research can benefit from either building upon these results or challenging them. Future researchers can improve their own investigations by looking at both the weaknesses and strengths of the approach taken here.

Appendix A. Entry strategy selection: a broader view

Clearly, business strategy involves much more than is discussed in Chapter 2. Indeed, what is discussed there is a small, though vital, part of strategic decision making. Strategic decision making involves essentially the dual process of environmental scanning and organisational introspection. The environment is monitored for opportunities and threats, while the organisation itself is assessed to determine how well it can respond to the challenges and pitfalls around it.

According to Johnson & Scholes (1984, p. 243)

> strategic analysis and choice are of little value to an organisation unless the proposals are capable of being implemented. Strategic change does not take place simply because it is considered to be desirable, it takes place if it can be made to work.

Thus, the strategy process typically involves deciding on the major actions to be taken for organisational survival or growth — actions such as market development, product development, and diversification of various kinds. These strategies have to be pursued by one of several methods, which Roberts & Berry (1985) call "entry strategies". Roberts & Berry list seven entry strategies:

- Acquisition
- Joint venture
- Internal development
- Internal venture
- Licensing

- Nurturing / venture capital
- Educational acquisition.

Three of these entry strategies are included in the fragment of decision making used as illustration in Chapter 2.

One effect of induction is to eliminate, from the decision, some of the attributes that an expert might suggest. Some experts might be uneasy about the small number of attributes that an induction algorithm typically singles out for consideration. For example, Kononenko (1990, p. 193-194) noted

> Although a decision tree outperformed the physicians ... with respect to diagnostic accuracy the physicians were not prepared to use [it] in practice. The rules ... were too short, containing only few, although most informative, attributes .... The physicians typically use all available information to make a decision and they are also able to estimate the reliability of the diagnosis. If the reliability is not high enough then additional examinations are needed.

In business strategy, the stakes are also high; and it is just as important to avoid rash decisions. However, people do use redundant attributes in their decisions. For instance, Ansoff (1968) offers the following factors in deciding on the entry strategy:

- Start-up synergy
- Operating synergy
- Start-up cost
- Timing (product cycle and learning curve)
- Risk
- Price/earnings ratio in the new industry
- Availability of attractive acquisition opportunities.

This list contains factors acknowledged as important by Roberts & Berry (1985). However, Roberts & Berry argue strongly that the question of how to enter cannot be divorced from that of which product markets to enter. On this basis, they see familiarity with the market to which the development is to be targeted, the technology of the product, and the level of involvement required as factors that outweigh all others.

Even so, inspection of Ansoff's table (Ansoff, 1968, p 169) reveals the first two factors are sufficient for the decision. This conclusion is supported when a case is generated from each row in Ansoff's table. The effects of start-up and operating synergy are so decisive that an (ID3-like) induction algorithm drops all the other factors as redundant. Here are the rules induced:

```
a) If start-up synergy is 'none'
   then the best method is acquisition.

b) If start-up synergy is weak
   and operating synergy is weak
   then the best method is acquisition.

c) If start-up synergy is weak
   and operating synergy is weak
   then the best method is joint venture.

d) If start-up synergy is strong
   then the best method is internal development.
```

If the other factors are desired in the induced rules, the first two factors must be dropped. When the cases are expressed in terms of all factors except start-up and operating synergy, the rules induced are as follows:

a) If the kind of growth is conglomerate diversification
   then the best method is acquisition.

b) If the kind of growth is concentric diversification
   then the best method is acquisition.

c) If the kind of growth is unrelated
   and timing is not of the essence
   then the best method is internal development.

d) If the kind of growth is unrelated
   and timing is of the essence
   then the best method is joint venture.

e) If the kind of growth is market development
   then the best method is internal development.

This confirms the redundancy in the original set of factors. It is, of course, possible that in certain cases both start-up synergy and start-up cost, say, affect the outcome of the decision. This might be so if high start-up synergy is not always accompanied by low start-up cost. If the input to the induction process includes examples in which the presence of one condition and not the other changes the strategy decided, both factors will appear in the rules generated. But such examples may be difficult to find if the two factors are related in the way these two appear to be.

Appendix B. Instructions for KAMSE

Instructions to Participants (B)

This session is divided into two parts. During the first part of the session, you will learn how to use the knowledge acquisition software. In the second part, you will be using the software to perform a predetermined task.

Practice

1. Make sure your computer is switched on, then continue reading this.

2. You have been given a diskette containing a knowledge acquisition program. When your machine is ready, insert the diskette in the diskette drive.

3. Now type `a:` to point your operating system to the diskette drive. You should now see the a: command prompt. If you don't, ask for help.

4. Type `SCENIC` and press the Enter key to start the program. You should now see the SCENIC logo screen.

5. Press F10. The top line of the screen should now be displaying red characters.

6. Type `B`

   Your screen should now be displaying a file selection screen.

7. Select the file named FAUNA. You should now see the class entry screen on which you will enter a list of animals.

8. Start the list by adding lizard. Here is how you do it:

- Press the / key to open the field for data entry. You should now see a small cursor appear about three quarters of the way down the left side of the screen. This is where any information you type will appear.

- Type: `Lizard`

  Then press the Enter key. The list should now include your lizard.

9. Now add lion and then elephant to the list.

10. Your list of animals should now be complete with the three types of animals we are interested in. Press the Esc key to progress to the next stage.

11. You should now see the screen for entering attributes and their possible values. You will need two attributes to describe the animals you just added. Here is how you add an attribute:

    - Press the / key. You should see a small cursor appear about three quarters of the way down the left side of the screen. This allows you to enter information.

    - Type: `Body temperature`

      Press the Enter key. The new attribute should now appear in place of the large cursor.

    - Body temperature (for our purposes) can have two values. So add: `warm cold` under the new attribute. (Remember the / key!)

12. Now move the block cursor to the next column and add the

    - attribute: `Trunk on face`
    - with values: `Yes`
    - and `No`

Your list of attributes and their possible values should now be sufficient for the three types of animals we are interested in. Press the Esc key to progress to the next stage.

13. You should now be looking at a screen for entering examples of each type of animal. Use the / key to open each column for data entry. Then use the cursor arrow keys to select the attribute values and class for each example. Here are the three examples you should enter:

```
1 cold      no          Lizard
2 warm      no          Lion
3 warm      yes         Elephant
```

14. Your list of examples should now be complete with the three types of animals we are interested in. Press the Esc key to progress to the next stage.

15. Now watch the screen and wait for a moment while the program processes the information you have put in so far. At the end of this processing, you will see, near the bottom of the screen, a message should be telling you to press the Enter key to continue. Go ahead, press the Enter key.

16. You should now see the validation screen. This is where you enter test cases to validate the knowledge you have been transferring to the program. The test cases have randomly generated values for each attribute. All you have to do is to select the correct class for each case. Here is how you do this for the first test case:

    - With the large cursor in the class column, press the / key. Now use the cursor arrow keys to find the appropriate class. Each time you press a cursor arrow key, another class will be displayed on the data entry line.
    - Enter the class for all the test cases displayed.

- Now press Esc to progress to the next stage.

17. The test cases you have entered are now going through a validation process. You should see the screen changing just too quickly for you to read what is being displayed. When this ends, you will see a message telling you to press the Enter key. Do as the message says.

18. The SCENIC logo should again be showing on the screen. This indicates that your training is over. You are now ready to use SCENIC for the main task.

Prescribed Task

Now that you are familiar with the software, you will be given a package containing eight items. These items comprise a knowledge domain for object identification. Handle the objects carefully as they will be collected at the end of the session.

1. Look at the items and identify them to yourself.
2. As you did earlier, press F10 and the B key.
3. You should see the file selection screen. Move the block cursor below last name in the list. Press the / key to open the field for data entry. Then type: Objects and press Enter.
4. Now use the software to gather information about the items and distinctions between them. Your goal is to do for these items what you did for the animals in the practice example.
5. When you get to the validation screen, consider each of the 32 cases carefully. For each case, look at the attribute values from left to right. Consider only the evidence you think relevant in making your decision, and ignore the other (sometimes conflicting) evidence.

Appendix C. Instructions for the Repertory Grid Technique

Instructions to Participants (R)

This session is divided into two parts. During the first part of the session, you will learn how to use the knowledge acquisition software. In the second part, you will be using the software to perform a predetermined task.

Practice

1. Make sure your computer is switched on, then continue reading this.

2. You have been given a diskette containing a knowledge acquisition program. When your machine is ready, insert the diskette in the diskette drive.

3. Now type a: to point your operating system to the diskette drive. You should now see the a: command prompt. If you don't, ask for help.

4. Type SCENIC and press the Enter key to start the program. You should now see the SCENIC logo screen.

5. Press F10. The top line of the screen should now be displaying red characters.

6. Type R

   Your screen should now be displaying a file selection screen.

7. Select the file named FAUNA. You should now see the class entry screen on which you will enter a list of animals.

8. Start the list by adding lizard. Here is how you do it:

- Press the / key to open the field for data entry. You should now see a small cursor appear about three quarters of the way down the left side of the screen. This is where any information you type will appear.

- Type: `Lizard`

  Then press the Enter key. The list should now include your lizard.

9. Now add lion and then elephant to the list.

10. Your list of animals should now be complete with the three types of animals we are interested in. Press the Esc key to progress to the next stage.

11. You should now see the list of three animals. Near the top of the screen is a question for you. Two of the animals shown are warm-blooded and the third is not. Position the block cursor on Lizard. Press the Enter key to select Lizard.

12. The three animals should now be split into two groups. Now enter the trait that the top two share. Here is how you do this:

    - Press the / key to open the field for data entry.

      Type: `warm-blooded`

      Press the Enter key. The new trait should now appear to the right of the top two animals. The block cursor should also have moved down to the right of the bottom animal.

    - Press the / key to open the field for data entry.

      Type: `cold-blooded`

      Press the Enter key. This opposite pole of the new trait should now appear to the right of the bottom animal.

- Check the display to ensure that you have not made any errors. Then press Esc to progress to the next stage.

13. Your screen should now be displaying the rating screen. A rating of 5 means the animal is warm-blooded. A rating of 1 means that it isn't. The three animals are already rated correctly. For practice, rerate one of them as follows:

    - Move the block cursor to the line you want to change.
    - Press /
    - Type the appropriate rating (in this case, either 1 or 5).

14. Check quickly that all the animals have been rated correctly. When you are satisfied, Press the Esc key to progress to the next stage.

15. You should again be presented with the three animals and asked which one is different. This time, select the elephant and use the trait: untrunked. When you are through, press Esc to progress to the next stage.

16. Now watch the screen and wait for a moment while the program processes the information you have put in so far. At the end of this processing, you will see, near the bottom of the screen, a message should be telling you to press the Enter key to continue. Go ahead, press the Enter key.

17. You should now see the validation screen. This is where you enter test cases to validate the knowledge you have been transferring to the program. The test cases have randomly generated values for each attribute. All you have to do is to select the correct class for each case. Here is how you do this for the first test case:

- With the block cursor in the class column, press the / key. Now use the cursor arrow keys to find the appropriate class. Each time you press a cursor arrow key, another class will be displayed on the data entry line.
- Enter the class for all the test cases displayed.
- Now press Esc to progress to the next stage.

18. The test cases you have entered are now going through a validation process. You should see the screen changing just too quickly for you to read what is being displayed.

    When this ends, you will see a message telling you to press the Enter key. Do as the message says.

19. The SCENIC logo should again be showing on the screen. This indicates that your training is over. You are now ready to use SCENIC for the main task.

Prescribed Task

Now that you are familiar with the software, you will be given a package containing eight items. These items comprise a knowledge domain for object identification. Handle the objects carefully as they will be collected at the end of the session.

1. Look at the items and identify them to yourself.
2. As you did earlier, press F10 and the R key.
3. You should see the file selection screen. Move the block cursor below last name in the list. Press the / key to open the field for data entry. Then type: `Objects` and press Enter.
4. Now use the software to gather information about the items and distinctions between them. Your goal is to do for these items what you did for the animals in the practice example.

5. When you get to the validation screen, consider each of the 32 cases carefully. For each case, look at the attribute values from left to right. Consider only the evidence you think relevant in making your decision, and ignore the other (sometimes conflicting) evidence.

Appendix D. Knowledge Units Elicited

This appendix shows the knowledge units elicited by each method. The two sets of units, which were selected because they are representative of most responses, were elicited from the same subject (number 8).

Knowledge from the Repertory Grid Technique

The repertory grid technique was this subject's second treatment. It is noticeable that, although elicited one week later, the constructs are remarkably similar to the attributes elicited earlier. It is not known whether this is due to carry-over effects or the persistence of the knowledge.

Domain : Objects

		Ratings				
Elements :	Washer	4	5	5	3	1
	Metal paper clip	5	1	5	5	1
	Plastic paper clip	3	1	1	5	1
	Button	5	5	1	3	5
	Hair clip	2	2	5	5	2
	One-pence piece	1	5	5	3	5
	One-cent coin	1	5	5	3	1
	Elastic band	5	1	1	1	5
Constructs:	Holes in / Solid					
	Circular / Rectangular					
	Metallic / Non-metallic					
	Rectangular / Non-defined shape					
	Dull / Shiny					

Rules Induced

1. IF Metallic
   AND Circular
   AND Holes
   Then Object is Washer

2. IF Metallic
   AND Circular
   AND Solid
   AND Dull
   THEN One-pence piece

```
3.  IF Metallic
    AND Circular
    AND Solid
    AND Shiny
    THEN One-cent coin

4.  IF Metallic
    AND Rectangular
    AND Holes in
    THEN Metal paper clip

5.  IF Metallic
    AND Rectangular
    AND Solid
    THEN Hair clip

6.  IF Non-metallic
    AND Circular
    AND Dull
    Then Button

7.  IF Non-metallic
    AND Rectangular
    AND Dull
    THEN Elastic band

8.  IF Non-metallic
    AND Bright
    THEN Plastic paper clip
```

Knowledge from KAMSE

The meaning of a few of the attributes is not readily evident. For example, what the subject meant by solid is not the usual meaning of the word, because all the metal objects would be solid, if the word was used in its usual sense. Perhaps rigid would better describe the attribute, but the washer would have been rigid, in the normal usage of rigid. The subject used the same word in the "holes in / solid" construct in the repertory grid, which makes her meaning clearer. Solid was used to mean having no holes. If the knowledge elicited were to be put to any further use, it might be appropriate to clarify this kind of issue with the domain expert.

```
Domain     : Objects

Classes    : Metal paper clip
             Hair grip
             Plastic paper clip
             Elastic band
             Washer
             Button
             One-cent coin
             One-pence piece

Attributes: Shape          :   Circle    Rectangle  Varied
            Material       :   Metal     Plastic    Elastic
            Solid          :   Yes       No
            Circular holes:    1         4          0
            Edges          :   Straight  Wavy       Round
            Appearance     :   Dull      Bright
```

Examples

Shape	Material	Solid	Circular Holes	Edges	Appear-ance	Object
Circle	Metal	Yes	0	Round	Bright	One-pence piece
Rectangle	Plastic	No	0	Straight	Bright	Plastic paper clip
Rectangle	Metal	No	0	Straight	Bright	Metal paper clip
Circle	Metal	No	1	Round	Bright	Washer
Circle	Metal	Yes	0	Round	Dull	One-cent coin
Rectangle	Metal	No	0	Wavy	Dull	Hair grip
Circle	Plastic	No	4	Round	Bright	Button
Varied	Elastic	No	0	Wavy	Dull	Elastic band

Rules induced

```
1.  IF Material = Metal
    AND Shape = Circle
    AND Rigid = Yes
    AND Appearance = Dull
    THEN Object is One-cent coin

2.  IF Material = Metal
    AND Shape = Circle
    AND Rigid = Yes
    AND Appearance = Bright
    THEN Object is One-pence piece

3.  IF Material = Metal
    AND Shape = Circle
    AND Rigid = No
    THEN Object is Washer
```

4. IF Material = Metal
   AND Shape = Rectangle
   AND Edges = Straight
   THEN Object is Metal paper clip

5. IF Material = Metal
   AND Shape = Rectangle
   AND Edges = Wavy
   THEN Object is Metal hair grip

6. IF Material = Plastic
   AND Shape = Circle
   THEN Object is Button

7. IF Material = Plastic
   AND Shape = Rectangle
   THEN Object is Plastic paper clip

8. IF Material = Elastic
   THEN Object is Elastic band

References

Adelman, L. (1981). "The Influence of Formal, Substantive & Contextual Task Properties on the Relative Effectiveness of Different Forms of Feedback in Multiple-Cue Probability Learning Tasks". *Organizational Behavior & Human Performance*, Vol. 27, pp 423-442.

Adelman, Leonard (1989). "Measurement Issues in Knowledge Engineering". *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 3 (May/June), pp 483-488.

Agarwal, Ritu & Tanniru, Mohan R. (1990). "Knowledge Acquisition Using Structured Interviewing: An Empirical Investigation". *Journal of Management Information Systems*, Vol. 7, No. 1, pp 123-140.

Anderson, Barry F. (1971). *The Psychological Experiment: an introduction to the scientific method*, Second Edition. Belmont, CA: Brooks/Cole Publishing Co.

Anderson, John R. (1976). *Language, Memory, and Thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, John R. (1982). "Acquisition of Cognitive Skill". *Psychological Review*, Vol. 89, No. 4, pp 369-406.

Anderson, John R. (1983). "Retrieval of Information from Long-Term Memory". *Science*, Vol. 220, No. 4592 (1 April), pp 25-30.

Anderson, John R. (1989). "A Theory of the Origins of Human Knowledge". *Artif. Intell.* (Netherlands), Vol. 40, No. 1-3 (Sept), pp 313-353.

Anjewierden, Anjo (1987). "Knowledge Acquisition Tools". *AI Communications*, Vol. 0, No. 1 (Aug), pp 29-38.

Ansoff, H. Igor (1968). *Corporate Strategy*. Middlesex, England: Penguin Books.

Arblaster, A. T. (1983). "The Evaluation of a Programming Support Environment". In Green, T; Payne, S.; van der Veer, G. (Eds.) *The Psychology of Computer Use*. London: Academic Press.

Baddeley, Alan D. (1976). *The Psychology of Memory*. New York: Basic Books, Inc.

Baddeley, Alan D. (1990). *Human Memory Theory and Practice*. Hove: Lawrence Erlbaum Associates Ltd.

Bain, William M. (1986). "Judge: a Case-Based Reasoning System". In Mitchell, Tom M., Carbonell, Jaime G., & Michalski, Ryszard S. (Eds.) *Machine Learning: A Guide to Current Research.*

Bainbridge, Lisanne (1986). "Asking Questions and Accessing Knowledge". *Future Computing Systems*, Vol. 1, Part 2, pp 143-149.

Balzer, William K.; Doherty, Michael, E.; & O'Connor, Raymond (1989). "Effects of Cognitive Feedback on Performance". *Psychological Bulletin*, Vol. 106, No. 3, pp 410-433.

Bareiss, E. Ray; Porter, Bruce W; & Weir, Craig C. (1988). "PROTOS: an Exemplar-Based Learning Apprentice". *Int. J. Man-Machine Studies*, Vol. 29, pp 549-561.

Bartlett, Frederic C. (1932). *Remembering: a study in experimental and social psychology.* Cambridge: Cambridge University Press.

Bell, Michael Z. (1985). "Why Expert Systems Fail". *Journal of the Operational Research Society*, Vol. 36, No. 7, pp 613-619.

Berwick, Robert C. (1985). *The Acquisition of Syntactic Knowledge.* Cambridge, MA: MIT Press.

Birmingham, William Peter (1988). "Automated Knowledge Acquisition for a Hierarchical Synthesis System". PhD dissertation. Carnegie-Mellon University.

Bischel, Martin & Seitz, Peter (1989). "Minimum Class Entropy: A Maximum Information Approach to Layered Networks". *Neural Networks*, Vol. 2, pp 133-141.

Boose, John H. (1985). "A Knowledge Acquisition Program for Expert Systems Based on Personal Construct Psychology". *Int. J. Man-Machine Studies*, Vol. 23, pp 495-525.

Boose, John H. (1988). "Uses of Repertory Grid-Centred Knowledge Acquisition Tools for Knowledge-Based Systems". *Int. J. Man-Machine Studies*, Vol. 29, pp 287-310.

Boose, John H. (1989). "A Survey of Knowledge Acquisition Techniques and Tools". *Knowledge Acquisition*, Vol. 1, No. 1, pp 3-37.

Boose, John H. & Bradshaw, Jeffrey M. (1987). "Expertise Transfer & Complex Problems: Using AQUINAS as a Knowledge-Acquisition Workbench for Knowledge-Based Systems". *Int. J. Man-Machine Studies*, Vol. 26, pp 3-28.

Bower, G.H.; Black, J.B.; & Turner, T.J. (1979). "Scripts in Memory for Texts". *Cognitive Psychology*, Vol. 11, pp 177-220.

Bowyer, John; Markowitz, Judith; & Yusko, Jay (1987). "Preparing Your Company for Artificial Intelligence". *AFIPS Conference Proceedings*, Vol. 56, pp 3-6.

Bradshaw, J. M. & Boose, J. H. (1990). "Decision analysis techniques for knowledge acquisition: combining information and preferences using AQUINAS and AXOTL". *Int. J. Man-Machine Stud.*, (UK) Vol. 32, No.2 (Feb.) pp 121-86.

Bratko, Ivan & Kononenko, Igor (1989). "Automating Knowledge Acquisition for Expert Systems". *Electroteh. Vestn.* (Yugoslavia), Vol. 56, Part 2, pp 2225-232.

Bray, James H. & Maxwell, Scott E. (1985). *Multivariate Analysis of Variance.* Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-054. Beverley Hills, CA: Sage Publications.

Breiman, L.; Friedman, J.H.; Olshen, R.A.; & Stone, C.J. (1984). *Classification and Regression Trees.* Belmont: Wadsworth.

Breuker, J. & Wielinga, R. (1983). "Analysis Techniques for Knowledge Based Systems: Part 2". *Report 1.2, Esprit Project.* University of Amsterdam.

Breuker, J. & Wielinga, R. (1987). "Use of Models in the Interpretation of Verbal Data". In A.L. Kidd (Ed.) *Knowledge Acquisition for Expert Systems: a practical handbook.* New York: Plenum Press.

Buchanan, Bruce G. & Feigenbaum, E. A. (1982). Foreword in Davis, R. & Lenat, D. *Knowledge-Based Systems in Artificial Intelligence.* New York: McGraw-Hill International Book Co.

Buchanan, Bruce G.; Barstow, David; Bechtal, Robert; Bennett, James; Clancey, William; Kulikowski, Casimir; Mitchell, Tom; & Waterman, Donald A. (1983). "Constructing an Expert System". In Hayes-Roth, F., Waterman. D. & Lenat, D. (Eds.) *Building Expert Systems.* Reading, Mass.: Addison-Wesley.

Buchberger, E.; Zsolnai, S.; & Trost, H. (1989). "Automatic Knowledge Acquisition fro VIE LANG: A Natural Language Understanding System". *Machine and Human Learning. Advances in European Research*, pp 259-264.

Buffa, Elwood S. (1972). *Operations Management: Problems and Models* Third Edition. Santa Barbara, CA: John Wiley & Sons.

Bunge, Mario (1973). *Method, Model and Matter.* Holland: D. Reidel Publishing Co.

Buntine, Wray & Niblett, Tim (1992). "A further comparison of splitting rules for decision-tree induction". *Machine Learning*, Vol. 8, pp 75-85.

Burton, Mike; Shadbolt, Nigel; Hedgecock, A. P.; & Rugg, G. (1987). "A Formal Evaluation of Knowledge Elicitation for Expert Systems: Domain 1". In Moralee, D.S. (Ed.) *Research and Development in Expert Systems IV.* Cambridge: Cambridge University Press.

Burton, Mike & Shadbolt, Nigel (1988). "Experiments in Knowledge Elicitation". *AISB Quarterly*, Part 65 (Summer Edition), pp 11-12.

Burton, A.M.; Shadbolt, N.R.; Rugg, G.; & Hedgecock, A.P. (1990). "The Efficacy of Knowledge Elicitation Techniques: a Comparison Across Domains and Levels of Expertise". *Knowledge Acquisition*, Vol. 2, pp 167-178.

Camp, C.J.; Lachman, J.L.; & Lachman, R. (1980). "Evidence for Direct Access and Information Retrieval in Question Answering". *Journal of Verbal Learning and Verbal Behaviour*, Vol. 19, pp 583-596.

Case, Robbie (1980). "The underlying mechanism of intellectual development". In John R. Kirby & John B. Biggs (Eds.) *Cognition, Development, and Instruction.* New York: Academic Press.

Cestnik, Bojan; Kononenko, Igor; & Bratko, Ivan (1987). "ASSISTANT 86: a Knowledge-Elicitation Tool for Sophisticated Users". *Proc. 2nd European Working Session on Learning* (Bled, Yugoslavia), pp 31-45.

Chadha, Sanjay R.; Mazlack, L.J.; & Pick, R.A. (1991). "Using existing knowledge sources (cases) to build an expert system". *Expert Systems* (UK) Vol.8, No.1 (Feb.), pp 3-12.

Chandrasekaran, B. (1991). "Interviews: Fredrick Hayes-Roth and Richard Fikes". *IEEE Expert*, Vol. 6, No. 5 (Oct), pp 3-14.

Channier, T. & Fournier, C. (1988). "ACTES: knowledge acquisition from texts for a specification expert". *Machine Interaction and Artificial Intelligence in Aeronautics and Space 1988*, pp 23-34.

Chapanis, Alphonse (1959). *Research Techniques in Human Engineering.* Baltimore: The Johns Hopkins Press.

Clancey, William J. (1983). "The Epistemology of a Rule-Based System — a framework for explanation". *Artificial Intelligence* Vol. 20, pp 215-251.

Clancey, William J. (1986). "Heuristic Classification". In Kowalik, Janusz S. (Ed.) *Knowledge Based Problem Solving*. New Jersey: Prentice-Hall.

Clancey, W.J. (1991). "Implications of the system model operator metaphor for knowledge acquisition". *Knowledge Acquisition for Knowledge Based Systems 1991*, pp 65-80.

Clark, Peter & Niblett, Tim (1987). "Induction in Noisy Domains". *Proc. 2nd European Working Session-on-Learning* (Bled, Yugoslavia), pp 11-30.

Clark, Peter & Niblett, Tim (1989). "The CN2 Induction Algorithm". *Machine Learning*, Vol. 3, pp 261-283.

Claxton, Guy (1980). *Growth Points in Cognition* London: Routledge.

Clearwater, S.H. & Provost, F.J. (1990). "RL4: A Tool for Knowledge-Based Induction". *Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence*. pp 24-30.

Cohen, Jacob (1977). *Statistical Power Analysis for the Behavioral Sciences*. New York: Academic Press.

Cohen, Gillian (1989). *Memory in the Real World*. London: Lawrence Erlbaum Associates.

Cohen, Paul R. & Howe, Adele E. (1988). "How Evaluation Guides AI Research". *AI Magazine*, Vol. 9, Part 4 (Winter), pp 35-43.

Cohen, Paul R. & Howe, Adele, E. (1989). "Toward AI Research Methodology: Three Case Studies in Evaluation". *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 3 (May/June), pp 634-646.

Collins, Edward; Ghosh, Sushimoto; & Scofield, Christopher (1988). "An Application of a Multiple Neural Network Learning System to Emulation of Mortgage Underwriting Judgements". *IEEE International Conference on Neural Networks* (July 1988), pp II 459-II 466.

Connell, N.A.D. (1987). "Expert Systems in Accountancy: A Review of Some Recent Applications". *Accounting and Business Research*, Vol. 17, No. 67, pp 221-233.

Connerton, Paul (1989). *How Societies Remember*. Cambridge: Cambridge University Press.

Constant, Patrick; Matwin, Stan; & Oppacher, Franz (1988). "Knowledge Acquisition for Planning Systems". *8th International Workshop. Expert Systems and their Applications* (Avignon), Vol. 2, pp 553-565.

Cooke, Nancy Marie (1987). "The Elicitation of Units of Knowledge and Relations: Enhancing Empirically Derived Semantic Networks". PhD dissertation. New Mexico State University.

Crawford, Stuart L. (1989). "Extensions to the CART Algorithm". *Int. J. Man-Machine Studies*, Vol. 31, pp 197-217.

Dalton, Peggy (1988). "Personal Meaning and Memory: Kelly and Bartlett". In Fay Fransella & Laurie Thomas (Eds.) *Experimenting with Personal Construct Psychology*. London: Routledge & Kegan Paul.

Davis, Randall & Lenat, Douglas B. (1982). *Knowledge-Based Systems in Artificial Intelligence.* New York: McGraw-Hill International Book Co.

Davis, J. Steve (1990). "Effects of Modularity on Maintainability of Rule-Based Systems". *Int. J. Man-Machine Studies*, Vol. 32, pp 439-447.

Dechter, R. & Michie. D. (1984). *Structured Induction of Plans and Programs.* IBM Los Angeles Scientific Center, Order No. G320-2770.

Deffner, G. & Ahrens, R. (1989). "On the Use of Formal Language and ill defined Quantifiers in Knowledge Acquisition". *Proceedings of the Human Factors Society 33rd Annual Meeting. Perspectives 1989.* Vol. 1, pp 356-360.

Department of Trade & Industry (1992a). *Manufacturing Intelligence: Inside UK Enterprise.* London: DTI.

Department of Trade & Industry (1992b). *Manufacturing Intelligence*, No. 9 (Winter 91/92). London: DTI.

Dhaliwal, Jasbir Singh & Benbasat, Izak (1990). "A Framework for the Comparative Evaluation of Knowledge Acquisition Tools and Techniques". *Knowledge Acquisition*, Vol. 2, pp 145-166.

Diaper, Dan (1989). Pamphlet announcing seminar on Knowledge Acquisition for Expert Systems, organised by IBC Technical Services Ltd., London, 8th Sept., 1989. (from synopsis of presentation by Dan Diaper).

Diederich, Joachim; Ruhmann, Ingo; & May, Mark (1987). "KRITON: a Knowledge-Acquisition Tool for Expert Systems". *Int. J. Man-Machine Studies*, Vol. 26, pp 29-40.

Dilger, W. & Moller, J. (1990). "CAUSA a tool for model based knowledge acquisition". *Int. J. Pattern Recognit. Artif. Intell.* (Singapore), Vol. 4, No. 3 (Sept.), pp 489-507.

Eden, Colin; Jones, Sue; & Sims, David (1983). *Messing About in Problems.* Oxford: Pergamon Press.

Eden, Colin & Jones, Sue (1984). "Using Repertory Grids for Problem Construction". *Journal of the Operational Research Society*, Vol. 35, No. 9, pp 779-790.

Eden, Colin (1988). "Cognitive Mapping and Review". *European Journal of Operational Research*, Vol. 36, No. 1, pp 1-13.

Ellis, Andrew W. & Young, A.W. (1988). *Human Cognitive Neuropsychology.* Hove: Lawrence Erlbaum Associates Ltd.

Ericsson, K.A. & Simon, H.A. (1984). *Protocol Analysis: Verbal Reports as Data.* Cambridge, MA: MIT Press.

Farley, B. G. & Clark, W. A. (1954). "Simulation of Self-Organizing Systems by Digital Computer". *Transactions of Professional Group of Information Theory*, Vol. PGIT, Part 4, pp 76-84.

Fayyad, Usama M. & Irani, Keki B. (1992). "On the handling of continuous-valued attributes in decision tree generation". *Machine Learning*, Vol. 8, pp 87-102.

Feigenbaum, Edward A. (1977). "The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering". *Proceedings of the Fifth International Conference of Artificial Intelligence*, pp 1014-1029.

Fisher, R.A. (1936). "The Use of Multiple Measurements in Taxonomic Problems". *Annals of Eugenics*, Vol. 7, No. 1, pp 179-188.

Flavell, John H. (1985). *Cognitive Psychology* Second Edition. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Foley, J.P. & Lehto, M.R. (1989). "Models of memory: implications for knowledge acquisition". *Designing and Using Human Computer Interfaces and Knowledge Based Systems. Proceedings of the Third International Conference on Human Computer Interaction*, Vol.II, pp 814-821.

Ford, K.M.; Petry, F.E.; Adams Webber, J.R.; & Chang, P.J. (1991). "An approach to knowledge acquisition based on the structure of personal construct systems". *IEEE Trans. Knowl. Data Eng.* (USA) Vol.3, No.1 (March), pp 78-88.

Forsyth, Richard (1984a). "The Expert Systems Phenomenon". In Forsyth, Richard (Ed.) *Expert Systems: Principles & Case Studies.* London: Chapman & Hall.

Forsyth, Richard (1984b). "Machine Learning Strategies". In Forsyth, Richard (Ed.) *Expert Systems: Principles & Case Studies.* London: Chapman & Hall.

Forsyth, Richard (1988). "1st Class [Software Package]". *Expert Systems* (UK). Vol.5, No.4, pp 342-346.

Forsythe, Diana E. & Buchanan, Bruce G. (1989). "Knowledge Acquisition for Expert Systems: some Pitfalls and Suggestions". *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 3 (May/June), pp 435-442.

Freundlich, Yehudah (1990). "Transfer Pricing: Integrating Expert Systems in MIS Environments". *IEEE Expert*, Vol. 5, No. 1 (Feb.), pp 54-62.

Gaines, Brian (1987). "An Overview of Knowledge-Acquisition and Transfer". *Int. J. Man-Machine Studies*, Vol. 26, pp 453-472.

Gaines, Brian R. (1989a). "Social and Cognitive Processes in Knowledge Acquisition". *Knowledge Acquisition*, Vol. 1, No. 1, pp 39-58.

Gaines, Brian R. (1989b). "Integration Issues in Knowledge Support Systems". *Int. J. Man-Machine Studies*, Vol. 31, pp 495-515.

Gaines, B. & Linster, M. (1990). "Development of second generation knowledge acquisition systems". *Current Trends in Knowledge Acquisition*, pp 143-160.

Gaines, Brian R. & Shaw, Mildred L. (1986). "A Learning Model for Forecasting the Future of Information Technology". *Future Computing Systems*, Vol. 1, pp 31-69.

Gammack, John G. (1987). "Eliciting expert conceptual structures using converging techniques". Ph.D. thesis, University of Cambridge.

Gammack, John G. & Anderson, A. (1990). "Constructive interaction in knowledge engineering". *Expert Systems*, Vol. 7, No. 1, pp 19-26.

Gams, Matjaz & Lavrac, Nada (1987). "Review of Five Empirical Learning Systems Within a Proposed Schemata". *Proc. 2nd European Working Session on Learning* (Bled, Yugoslavia), pp 46-66.

Garg-Janardan, C. & Salvendy, G. (1988). "A Structured Knowledge Elicitation Methodology for Building Expert Systems". *Int. J. Man-Machine Studies*, Vol. 29 (Oct), pp 377-406.

Gaschnig, John; Klahr, Philip; Pople, Harry; Shortliffe, Edward; & Terry, Allan (1983). "Evaluation of Expert Systems: Issues & Case Studies". In Hayes-Roth, F.; Waterman, D.; & Lenat, D. (Eds.) *Building Expert Systems.* Reading, MA: Addison-Wesley.

Gettig, Gary A. (1989). "KAM: a Tool to Simplify the Knowledge Acquisition Process". *Fourth Conference on Artificial Intelligence for Space Applications* (NASA Conf. Publ. 3013), pp 47-55.

Gilmartin, Kevin J.; Newell, Allen; & Simon, Herbert (1975). "A Program Modeling Short-Term Memory Under Strategy Control". In Charles Norval Cofer (Ed.) *The Structure of Human Memory.* San Francisco: W.H. Freeman & Co.

Ginsberg, Allen (1988). *Automatic Refinement of Expert System Knowledge Bases.* London: Pitman.

Goodman, R.M. & Latin, H. (1991). "Automated Knowledge Acquisition from Network Management Databases". In I. Krishnan & W. Zimmer (Eds.) *Integrated Network Management, II.* Elsevier Science Publishers, B.V. (North-Holland).

Grady, Robert B. & Caswell, Deborah L. (1987). *Software Metrics: Establishing a Company-Wide Program.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

Greene, Judith (1987). *Memory, Thinking and Language.* London: Metheun.

Gruber, Thomas R. (1988). "Acquiring Strategic Knowledge from Experts". *Int. J. Man-Machine Studies*, Vol. 29, pp 579-597.

Gruber, Thomas & Cohen, Paul (1987). "Principles of Design for Knowledge Acquisition". *Proceedings of 3rd IEEE-CS Conference on Artificial Intelligence Applications (CATA).* Orlando, Florida (Feb.), pp 9-15.

Gutwald, Paul M. & Wallace, William A. (1987). "Rapid Prototyping for Knowledge Acquisition on a "Wicked" Problem: the Case of Identifying Serial Murderers". *DSS-87 Transactions: 7th International Conference on Decision Support Systems.* Institute of Management Science. pp 32-42.

Hall, Robert A. Jr. (1965). *Sound and Spelling in English.* Philadelphia: Chilton Books.

Hart, Anna (1986). *Knowledge Acquisition for Expert Systems.* London: Kogan-Page.

Hayes-Roth, Frederick (1989). "Towards Benchmarks for Knowledge Systems and Their Implications for Data Engineering". *IEEE Transactions on Knowledge and Data Engineering*, Vol. 1, No. 1 (March), pp 101-110.

Hayes-Roth, Frederick; Waterman, D.A.; & Lenat, D.B. (1983). "An Overview of Expert Systems". In Hayes-Roth, F.; Waterman, D.; & Lenat, D. (Eds.) *Building Expert Systems.* Reading, MA: Addison-Wesley.

Hoffman, Robert R. (1987). "The Problem of Extracting the Knowledge of Experts from the Perspective of Experimental Psychology". *AI Magazine*, Vol. 8, Part 2 (Summer), pp 53-67.

Hofstede, Geert (1980). *Culture's Consequences — International Differences in Work-Related Values.* Beverly Hills, CA: Sage Publications.

Holland, John H. (1975). *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press.

Hunt, E.B.; Marin, J.; & Stone, P. (1966). *Experiments in Induction.* New York: Academic Press.

Jacobson, Chris & Freiling, Michael. J. (1988). "ASTEK: A Multi-Paradigm Knowledge Acquisition Tool for Complex Structured Knowledge". *Int. J. Man-Machine Studies*, Vol. 29, pp 311-327.

Johnson, Gerry & Scholes, Kevan (1984). *Exploring Corporate Strategy*. London: Prentice-Hall International.

Johnson-Laird, P.N. (1983). *Mental Models: towards a cognitive science of language, inference and consciousness*. Cambridge University Press.

Joseph, A.T. & Davies, B.J. (1990). "EXCAP an expert process planning system for turned components". *First International Conference on Expert Planning Systems* (Conf. Publ. No. 322), pp 130-135.

Jung, John (1969). "Current Practices and Problems in the Use of College Students for Psychological Research". *The Canadian Psychologist*, Vol. 10, No. 3 (July), pp 280-290.

Keen, Terry & Bell, Richard (1980). "One Thing Leads to Another". *Int. J. Man-Machine Studies*. Vol. 13, No. 1 (July), pp 25-38.

Kelly, George (1955). *The Psychology of Personal Constructs*. New York: W. W. Norton & Co., Inc.

Keppel, Geoffrey (1982). *Design and Analysis: A Researcher's Handbook* (Second Edition). New Jersey: Prentice-Hall.

Kitto, Catherine M. & Boose, John H. (1989). "Selecting Knowledge Acquisition Tools and Strategies Based on Application Characteristics". *Int. J. Man-Machine Studies*, Vol. 31, pp 149-160.

Klein, Jonathan & Cooper, Dale F. (1981). "Assessing Problem Complexity". *European Journal of Operational Research*, Vol. 6, pp 243-247.

Klein, Jonathan & Cooper, Dale F. (1982). "Cognitive Maps of Decision-Makers in a Complex Game". *Journal of the Operations Research Society*, Vol. 33, No. 1 (Jan).

Klinker, G.; Bentolila, J.; Genetet, S.; Grimes, M.; & McDermott, J. (1987). "KNACK - Report-Driven Knowledge Acquisition". *Int. J. Man-Machine Studies*, Vol. 26, No. 1, pp 65-79.

Klinker, G.; Genetet, S.; McDermott, J. (1988). "Knowledge Acquisition for Evaluation Systems". *Int. J. Man-Machine Studies*, Vol. 29, No. 6, pp 715-731.

Kolodner, Janet L. (1983). "Towards an understanding of the role of experience in the evolution from novice to expert". *Int. J. Man-Machine Studies*, Vol. 19, pp 497-518.

Kononenko, I. (1990). "Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition". *Current Trends in Knowledge Acquisition*, pp 190-197.

Koontz, Harold & O'Donnell, Cyril (1972). *Principles of Management: An Analysis of Managerial Functions* Fifth Edition. Tokyo: McGraw-Hill.

Kors, J.A.; Sittig, A.C.; & van Bemmel, J.H. (1990). "The Delphi Method to Validate Diagnostic Knowledge in Computerized ECG Interpretation". *Methods Inf. Med.* (Germany), Vol. 29, No. 1 (Jan), pp 44-50.

Kotler, Philip (1984). *Marketing Management: Analysis, Planning and Control* Fifth Edition. London: Prentice-Hall International.

Kraemer, Helena Chmura & Thiemann, Sue (1987). *How Many Subjects? Statistical Power Analysis in Research.* California: Sage Publications.

LaFrance, Marianne (1987). "The Knowledge Elicitation Grid: a Method of Training Knowledge Engineers". *Int. J. Man-Machine Studies*, Vol. 26 (Feb), pp 245-255.

Larichev, O.I. & Morgoev, V.K. (1991). "Problems, methods, and systems for elicitation of expert knowledge". *Autom. Remote Control* (USA) Vol.52, No.6, pt.1 747-64 (June), pp 3-27.

Latta, Gail F. & Swigger, Keith (1992). "Validation of the Repertory Grid for Use in Modelling Knowledge". *Journal of the American Society for Information Science*, Vol. 32, No. 2, pp 115-129.

Lee, Robert B. (1989). "Division of Attention: the Single-Channel Hypothesis Revisited". *The Quarterly Journal of Experimental Psychology*, Vol. 41A, No. 1, pp 1-17.

Lehner, Paul E. (1989). "Toward an Empirical Approach to Evaluating the Knowledge Base of an Expert System". *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, No. 3 (May/June), pp 658-662.

Levi, Keith (1989). "Expert Systems Should be More Accurate than Human Experts: Evaluation Procedures from Human Judgement and Decisionmaking". *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, No. 3 (May/June), pp 647-657.

Liebowitzi, Jay (1986). "Useful Approach for Evaluating Expert Systems". *Journal of Expert Systems*, Vol 3, pp 86-96.

Lippmann, Richard P. (1987). "An Introduction to Computing with Neural Nets". *IEEE ASSP Magazine*, April 1987, pp 4-22.

Lundell, James Walfred (1988). "Knowledge Extraction and the Modelling of Expertise in a Diagnostic Task". PhD dissertation. University of Washington.

Lydiard, T.J. (1992). "Overview of Current Practice and Research Initiatives for the Verification and Validation of KBS". *Knowledge Engineering Review* Vol. 7, No. 2 (June), pp 101-113.

Mandler, George (1986). "Reminding, Recalling, Recognizing: Different Memories?". In F. Klix & H. Hagendorf (Eds.) *Human Memory and Cognitive Capabilities.* Amsterdam: Elsevier Science Publishers B.V. (North-Holland).

Marcus, M. (1980). *A Theory of Syntactic Recognition for Natural Language.* Cambridge, MA: MIT Press.

Marcus, Sandra (1987). "Taking Backtracking with a Grain of SALT". *Int. J. Man-Machine Studies* Vol. 26, pp 383-398.

Marcus, Sandra (1989). "SALT: a Knowledge Acquisition Language for Propose-and-Revise Systems". *Artificial Intelligence* Vol. 9, pp 1-37.

Marcus, S. & McDermott, J. (1989). "SALT: a Knowledge Acquisition Language for Propose and Revise Systems". *Artif. Intell.* (Netherlands), Vol 39, No. 1 (May), pp 1-37.

Marr, D. (1980). "Visual Information Processing: The Structure and Creation of Visual Representation". *Philosophical Transactions of the Royal Society* (London), Vol. B290, pp 199-218.

Marshall, Judi & McLean, Adrian (1985). "Exploring Organizational Culture as a Route to Organizational Change". In Valerie Hammond (Ed.) *Current Research in Management.* Francis Pinter (Publishers).

Marshall, G.; Kellett, J. M.; Lim, B. S.; & Boardman, J. T. (1987). "PIPPA: an Expert Project Planning System in Manufacturing Engineering". *KBS87: Proceedings of the International Conference*, pp 199-205.

Matheson, Douglas W.; Bruce, Richard L.; & Beauchamp, Kenneth L. (1978). *Experimental Psychology: Research Design and Analysis* (3rd Edition). New York: Holt, Rinehart & Winston.

Matwin, S. & Oppacher, F. (1988). "Learning by Watching: An Incremental Machine Learning Method that Acquires Rules by Conceptaul Clustering". *Methodologies for Intelligent Systems 3. Proceedings of the 3rd International Symposium* pp 363-373.

Matwin, Stan; Oppacher, Franz; & Constant, Patrick (1988). "Knowledge Acquisition by Incremental Learning from Problem-Solution Pairs". *Comput. Intell.* (Canada), Vol. 5, Part 2 (May), pp 58-66.

McClanahan, Anne & Luce, Thom (1988). "Tactics for Creating an Example Based Knowledge Base for the 1st-Class Expert System Shell". *ISECON '88.- 7th Annual Information Systems in Education Conference. Proceedings of the Conference* (Dallas, TX, USA, 29-30 Oct). pp 111-117.

McGraw, K.L. (1989). "Knowledge acquisition for intelligent instructional systems". *J. Artif. Intell. Educ.* (USA) Vol. 1, No. 1 (Fall), pp 11-26.

Michalski, R. S. & Chilausky, R. (1980). "Knowledge Acquisition by Encoding Expert Rules Versus Computer Induction from Examples: a Case Study Involving Soybean Pathology". *Int. J. Man-Machine Studies*, Vol. 12, pp 63-87.

Michalski, R. S. & Chilausky, R. L. (1980). "Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis". *International Journal of Policy Analysis and Information Systems*, Vol. 4, No. 2, pp 125-161.

Michalski, R.S. & Larson, J.B. (1983). "Incremental Generation of VL1 Hypotheses: the underlying methodology and the description of program AQ11". Department of Computer Science, University of Illinois at Urbana-Champaign (Report No. ISG 83-5).

Michie, Donald (1982). "Measuring the Knowledge-Content of Expert Programs". *Bulletin of the Institute of Mathematics and its Applications*, Vol. 18, pp 216-220.

Michie, Donald (1991). "Rules that are meant for breaking". *Computing*, 24 January, 1991; p 15.

Mingers, John (1989). "An Empirical Comparison of Selection Measures for Decision-Tree Induction". *Machine Learning*, Vol. 3, pp 319-342.

Minsky, Marvin L. (1975). "A Framework for Representing Knowledge". In P.H. Winston (Ed.) *The Psychology of Computer Vision.* New York: McGraw Hill.

Minsky, Marvin L. & Papert, Seymour (1969). *Perceptrons.* Cambridge, MA: MIT Press.

Minsky, Marvin L. & Papert, Seymour (1988). *Perceptrons: An Introduction to Computational Geometry.* (Expanded Edition) Cambridge, MA: MIT Press.

Mitchell, Tom M.; Carbonell, Jaime G.; & Michalski, Ryszard S. (Eds.) (1986). *Machine Learning: A Guide to Current Research.* Palo Alto, CA: Morgan Kaufman.

Morecroft, J. D. W. (1988). "System Dynamics and Microworlds for Policymakers". *European Journal of Operational Research,* Vol. 35, No. 3, pp 301-320.

Morik, Katharina (1987). "Acquiring Domain Models". *Int. J. Man-Machine Studies,* Vol. 26, pp 93-104.

Morris, Peter (1988). "Memory Research: Past Mistakes and Future Prospects". In Guy Claxton (Ed.) *Growth Points in Cognition.* London: Routledge.

Morris, P.E.; Tweedy, M.; & Gruneberg, M.M. (1985). Interest, Knowledge and the Memorising of Soccer Scores". *British Journal of Psychology,* Vol. 76, pp 415-425.

Muggleton, Stephen R. (1986). "Inductive Acquisition of Expert Knowledge". PhD dissertation. University of Edinburgh.

Mumford, Enid & MacDonald, W. Bruce (1989). *XSEL's Progress: the Continuing Journey of an Expert System.* Chichester: John Wiley & Sons Ltd.

Murphy, G.L. & Wright, J.C. (1984). Changes in Conceptual Structure with Expertise: Differences Between Real-world Experts and Novices". *Journal of Experimental Psychology: Learning, Memory and Cognition,* Vol. 10, pp 144-155.

Musen, Mark (1989). "An Editor for the Conceptual Models of Interactive Knowledge-Acquisition Tools". *Int. J. Man-Machine Studies* Vol. 31, pp 673-698.

Naylor, Chris (1985). *Build Your Own Expert System.* New York: Halsted Press (also Wilmslow: Sigma Technical Press).

Neale, Ian M. (1987). "Knowledge Acquisition for Expert Systems: a Review and Case Study". MSc dissertation. Loughborough University of Technology.

Neale, Ian M. (1988). "First Generation Expert Systems: a Review of Knowledge Acquisition Methodologies". *Knowledge Engineering Review* (UK), Vol. 3, No. 2 (June), pp 105-145.

Newell, Allen (1982). "The Knowledge Level". *Artificial Intelligence*, Vol. 18, pp 87-127.

Niblett, Tim (1987). "Constructing Decision Trees in Noisy Domains". *Proc. 2nd European Working Session-on-Learning* (Bled, Yugoslavia), pp 67-78.

Nicholson, Clive (1988). "Boon or Bandwagon? A Study of Some British Organizations that Have Acquired Expert-Systems Development Tools". Master's Dissertation, School of Management, University of Bath.

Nicholson, Clive (1992a). "Learning Without Case Records: a mapping of the repertory grid technique onto knowledge acquisition from examples". *Expert Systems* Vol. 9, No. 2 (May), pp 79-87.

Nicholson, Clive (1992b). "The Controlled Experiment in Knowledge-Acquisition Research". *IBM Journal of Research and Development*, Vol. 36, No. 6 (in press).

Nickerson, R.S. (1977). "Some Comments on Human Archival Memory as a Very Large Data Base". *Proceedings of the Third International Conference on Very Large Data Bases.* (Tokyo, October).

Noble, David F. (1989). "Schema-based knowledge elicitation for planning and situation assessment aids". *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.19, No.3 (May/June), pp 473-482.

Norris, D. E. (1986). "Machine Learning Using Fuzzy Logic with Applications in Medicine". PhD dissertation. University of Bristol.

O'Leary, Daniel E. & Watkins, Paul R. (1990). "A portfolio of knowledge acquisition approaches for a knowledge based system". *Second International Symposium on Expert Systems in Business, Finance and Accounting*, Vol.3, pp. 264-268.

Partridge, D. (1986). *Artificial Intelligence: Applications in the Future of Software Engineering*. Chichester: Ellis Horwood Ltd.

Pazzani, Michael & Kibler, Dennis (1992). "Utility of Knowledge in Inductive Learning". *Machine Learning* Vol 9, No. 1 (June), pp 57-94.

Politakis, Peter G. (1985). *Empirical Analysis for Expert Systems*. Boston, MA: Pitman Advanced Publishing Program.

Posner, Michael I. (1973). *Cognition: An Introduction*. Glenview, IL: Scott, Foresman & Co.

Posner, M.I. & Klein, R. (1973). "On the functions of consciousness". In Kornblum (Ed.) *Attention and Performance, Vol. IV*. New York: Academic Press.

Quinlan, J.R. (1979). "Discovering Rules by Induction from Large Collections of Examples". In Donald Michie (Ed.). *Expert Systems in the Micro-Electronic Age*. Edinburgh: Edinburgh University Press.

Quinlan, J. R. (1986). "Induction of Decision Trees". *Machine Learning*, Vol. 1, pp 81-106.

Quinlan, J. R. (1987). "Generating production rules from decision trees". *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pp 304-307.

Quinlan, J. R. (1989). "Unknown Attribute Values in Induction". *Proceedings of the Sixth International Machine Learning Workshop*. Cornell, NY: Morgan Kaufman.

Quinlan, J.R. (1991). "Inductive knowledge acquisition from structured data". *Knowledge Acquisition for Knowledge Based Systems 1991*, pp 97-112.

Quinlan, J. R.; Compton, P. J.; Horn, K. A.; & Lazarus, L. (1986). "Inductive Knowledge Acquisition: a Case Study". *Proceedings of 2nd Australian Conference on the Application of Expert Systems*, pp 157-173.

Rada, Roy (1984). "Automating Knowledge Acquisition". In Forsyth, Richard (Ed.) *Expert Systems: Principles & Case Studies*. London: Chapman & Hall.

Rappaport, A. T. & Gaines, B. R. (1990). "Integrated Knowledge Base Building Environments". *Knowledge Acquisition*, Vol. 2, No. 1 (March), pp 51-71.

Rasmussen, J. (1980). "The Human as a Systems Component". In Smith, H. T. & Green, T. R. G. (Eds.) *Human Interaction with Computers*. London: Academic Press.

Rau, L.F.; Jacobs, P.S.; & Zernik, U. (1989). "Information extraction and text summarization using linguistic knowledge acquisition". *Inf. Process. Manage.* (UK) Vol. 25, No. 4, pp 419-428.

Reimer, Ulrich (1990). "Automatic acquisition of terminological knowledge from texts". *ECAI 90. Proceedings of the 9th European Conference on Artificial Intelligence* pp 547-549.

Reitman Olson, Judith & Rueter, Henry H. (1987). "Extracting Expertise from Experts: Methods for Knowledge Acquisition". *Expert Systems*, Vol. 4, No. 3 (August), pp 152-168.

Roberts, Edward B. & Berry, Charles A. (1985). "Entering New Businesses: Selecting Strategies for Success". *Sloan Management Review*, Vol. 26, No. 3 (Spring).

Rosenblatt, F. (1958). "The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain". *Psychological Review*, Vol. 65, No. 6, pp 386-408.

Roskar, Egidija (1988). "Synthesis of Expert Knowledge in Medicine Using Machine Learning Applied to Data in a Database". *Medical Informatics: Computers in Clinical Medicine; British Medical Informatics Society*, Part 13-15 (Sept.), pp 77-82.

Roskar, Egidija; Bratko, Ivan; Kononenko, Igor; Cuk, Miran; & Abrams, Paul (1985). "An Application of Computer Assisted Multivariate Statistical Methods and Artificial Intelligence to the Diagnosis of Lower Urinary Tract Disorders". *Automatika*, Vol. 26, pp 177-181.

Rowley, D.T. (1990). "PC/Beagle". *Expert Systems*, Vol. 7, No. 1 (Feb), pp 58-60.

Rugg, G.; Corbridge, C.; Major, N.P.; Burton, A.M.; & Shadbolt, N. (1992). "A Comparison of Sorting Techniques in Knowledge Acquisition". *Knowledge Acquisition*, Vol. 4, No. 3 (Sept.), pp 279-292.

Rumelhart, D.E. (1975). "Notes on a Schema for Stories". In D.G. Bobrow & A. Collins (Eds.) *Representation and Understanding*. New York: Academic Press.

Rumelhart, D.E. & Norman, D.A. (1985). "Representation of Knowledge". In A.M. Aitkenhead & J.M. Slack (Eds.) *Issues in Cognitive Modelling*. London: Lawrence Erlbaum Associates Ltd.

Ryle, A. & Lunghi, M. (1970). "The Dyad Grid: A Modification of Repertory Grid Technique". *Brit. J. Psychiat*, Vol. 117, pp 323-327.

Sager, Naomi (1981). *Natural Language Information Processing: A Computer Grammar of English and its Applications*. Reading, MA: Addison-Wesley Publishing Co., Inc.

Saito, Kazumi & Nakano, Ryohei (1988). "Medical Diagnostic Expert System Based on PDP Model". *IEEE International Conference on Neural Networks*, Vol. 1, pp I-255 to I-262.

Sansone, Carol (1989). "Competence Feedback, Task Feedback, & Intrinsic Interest: an Examination of Process & Content". *Journal of Experimental Social Psychology*, Vol. 25, pp 343-361.

Schank, Roger C. (1975). *Conceptual Information Processing*. Amsterdam: North-Holland.

Schank, Roger C. (1982). *Dynamic Memory: a theory of reminding and learning in computers and people*. New York: Cambridge University Press.

Schmidt, Gabriele & Schmalhofer, Franz (1990). "Case oriented knowledge acquisition from texts". *Current Trends in Knowledge Acquisition* pp 302-312.

Schroder, O.; Frank, K.D.; Kohnert, K.; Mobus, C.; & Rauterberg, M. (1990). "Instruction-based Knowledge Acquisition and Modification: the operational knowledge for a functional, visual, programming language". *Comput. Hum. Behav.* (USA), Vol.6, No.1, pp 31-49.

Selkirk, Elisabeth (1983). *The Syntax of Words.* Cambridge, MA: The MIT Press.

Shadbolt, Nigel (1990). "Knowledge Based Knowledge Acquisition". *IEE Colloquium on Knowledge Engineering* (Digest No. 77), pp 5/1-5/3.

Shalin, Valerie L.; Wisniewski, Edward J.; & Levi, Keith R. (1988). "A Formal Analysis of Machine Learning for Knowledge Acquisition". *Int. J. Man-Machine Studies*, Vol. 29, pp 429-446.

Shaw, Michael J. & Gentry, James A. (1990). "Inductive Learning for Risk Classification". *IEEE Expert*, Vol. 5, No. 1 (Feb.), pp 47-53.

Shaw, Mildred (1979). "Conversational Heuristics for Eliciting Shared Understanding". *Int. J. Man-Machine Studies*, Vol. 11, No. 5 (Sept), pp 621-634.

Shaw, Mildred L. (1981). *Recent Advances in Personal Construct Technology.* London: Academic Press.

Shaw, Mildred L. G. (1988). "Knowledge Elicitation Techniques for Knowledge-Based Systems". *AI & Simulation: the Diversity of Applications. Proceedings of the SCS Multiconference on AI and Simulation*, San Diego, CA, 3-5 Feb, pp 19-24.

Shaw, Mildred L. G. & Thomas, Laurie F. (1978). FOCUS on Education - an Interactive Computer System for the Development and Analysis of Repertory Grids". *Int. J. Man-Machine Studies*, Vol. 10, pp 139-173.

Shaw, Mildred L. & Gaines, Brian R. (1987). "KITTEN: Knowledge Initiation and transfer Tools for Experts and Novices". *Int. J. Man-Machine Studies*, Vol. 27, No. 3, pp 251-280.

Shaw, Mildred & Woodward, Brian (1988). "Validation in a Knowledge Support System: Construing and Consistency with Multiple Experts". *Int. J. Man-Machine Studies*, Vol. 29, pp 329-350.

Shema, David B. & Boose, John H. (1988). "Refining Problem-Solving Knowledge in Repertory Grids Using a Consultation Mechanism". *Int. J. Man-Machine Studies*, Vol. 29, pp 447-460.

Shwartz, Steven P. & Kosslyn, Stephen M. (1982). "A Computer Simulation Approach to Studying Mental Imagery". In Jacques Mehler, Edward C.T. Walker, & Merrill Garrett (Eds.) *Perspectives on Mental Representation.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Smith, Stephen F. (1980). "A Learning Systems Based on Genetic Adaptive Algorithms". PhD dissertation. University of Pittsburgh.

Smith, Stephen F. (1984). "Adaptive Learning Systems". In Forsyth, Richard (Ed.) *Expert Systems: Principles & Case Studies.* London: Chapman & Hall.

Snodgrass, Joan Gay (1989). "How many memory systems are there really? Some evidence from the picture fragment completion task". In Chizuko Izawa (Ed.) *Current Issues in Cognitive Processes: The Tulane Flowerree Symposium on Cognition.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Spackman, K.A. (1990). "A comparison of two methods of inductive knowledge acquisition for medical knowledge based systems". *Fourteenth Annual Symposium on Computer Applications in Medical Care. Standards in Medical Informatics. A Conference of the American Medical Informatics Association 1990*, pp 641-644.

Stanley, William B. (1989). "Insight without Awareness: on the Interaction of Verbalization, Instruction & Practice in a Simulated Process Control Task". *The Quarterly Journal of Experimental Psychology*, Vol. 41A, No. 3, pp 553-577.

Steels, L. & Campbell, J.A. (1985). *Progress in Artificial Intelligence.* Chichester: Ellis Horwood Ltd.

Stefik, Mark; Aikens, J.; Balzer, R.; Benoit, J.; Birnbaum, L.; Hayes-Roth, F.; & Sacerdoti, E. (1983). "Basic Concepts for Building Expert Systems". In Hayes-Roth, F., Waterman. D. & Lenat, D. (Eds.) *Building Expert Systems.* Reading, Mass.: Addison-Wesley.

Stevens, Antony (1984). "How Shall We Judge an Expert System?". In Forsyth, R. (Ed.) *Expert Systems Principles & Case Studies.* London: Chapman & Hall.

Stevens, James P. (1980). "Power of the Multivariate Analysis of Variance Tests". *Psychological Bulletin*, Vol. 88, pp 728-737.

Stevenson, R. J.; Manktelow, K. I.; & Howard, M. J. (1988). "Knowledge Elicitation: Dissociating Conscious Reflections from Automatic Processes". In Jones, D. M. & Winder, R. (Eds.) *People & Computers IV.* Cambridge: Cambridge University Press (on behalf of British Computer Society).

Storrs, Graham (1989). "The Alvey DHSS Large Demonstrator Project Knowledge ANalysis Tool: KANT". In McAleese, Ray (Ed.) *Hypertext Theory into Practice.* London: Intellect.

Stout, Jeffrey; Caplain, Gilbert; Marcus, Sandra; & McDermott, John (1988). "Toward Automating Recognition of Differing Problem-Solving Demands". *Int. J. Man-Machine Studies*, Vol. 29, pp 599-611.

Sykes, David John (1987). "A Methodology for Developing Second Generation Expert Systems using Qualitative Simulation". PhD dissertation. Arizona State University.

Szolovits, Peter (1986). "Knowledge-Based Systems: a Survey". In Brodie, M. & Mylopoulos, J. (Eds.). *On Knowledge-Base Management Systems: Integrating Artificial Intelligence and Database Technologies.* Berlin: Springer-Verlag.

Tanyi, E. & Linkens, D.A. (1989). "Addition of a knowledge acquisition facility to a knowledge based environment for modelling and simulation (KEMS)". *ESC 89. Proceedings of the 3rd European Simulation Congress*, pp 193-197.

Taylor, Talbot J. & Cameron, Deborah (1987). *Analysing Conversation: Rules and Units in the Structure of Talk.* Oxford: Pergamon Press.

Tompsett, C. P. (1989). "Knowledge Acquisition for Intelligent Tutoring Systems". *Fifth International Expert Systems Conference*, pp 19-94.

Tulving, Endel (1962). "Subjective Organization in Free Recall of 'Unrelated' Words". *Psychological Review*, Vol. 69, pp 344-354.

Tulving, Endel (1985). "How many memory systems are there?". *American Psychologist*, Vol. 4, pp 385-398.

van Melle, William J. (1981). *System Aids in Constructing Consultation Programs.* Ann Arbor, MI: UMI Research Press.

Vessey, I. (1988a). "Expert Novice Knowledge Organization - an Empirical Investigation Using Computer-Program Recall". *Behav. Inf. Tech.*, Vol. 7, No. 2, pp 153-171.

Vessey, I. & Weber, R. (1988). "Conditional Statements & Program Coding - an Experimental Evaluation". *Int. J. Man-Machine Studies*, Vol. 21, No. 2, pp 161-190.

Vinze, Ajay S. (1992). "Empirical Verification of Effectiveness for a Knowledge-Based System". *Int. J. Man-Machine Studies* Vol. 37, No. 3 (Sept.), pp 309-334.

Voss, Angi (1990). "Model-based knowledge acquisition". *Information Systems and Artificial Intelligence: Integration Aspects. First Workshop Proceedings* (Ulm, Germany), pp 256-272. Springer-Verlag.

Waterman, Donald A. (1985). *A Guide to Expert Systems.* Reading, MA: Addison Wesley.

Weiss, Sholom M. & Kulikowski, Casimir A. (1984). *A Practical Guide to Designing Expert Systems.* London: Chapman & Hall.

Weizenbaum, Joseph (1976). *Computer Power and Human Reason: From Judgement to Calculation.* San Francisco, CA: Freeman (also Penguin Books).

Welbank, Margaret (1990). "An Overview of Knowledge Acquisition Methods". *Interacting with Computers*, Vol. 2, No. 1, pp 83-91.

Welch, J.C. (1898). "On the measurement of mental activity through muscular activity and the determination of a constant of attention". *American Journal of Physiology*, Vol. 1, pp 283-306.

Whipple, C.; Davis, L.; Kam, J.; & Needham, J. (1989). "Knowledge acquisition for an Internal Revenue Service classification system". *Proceedings of the Annual AI Systems in Government Conference*, pp 281-288. Washington, DC: IEEE Comput. Soc. Press.

Whitehall, Bradley Lane (1990). Knowledge-based Learning: Integration of Deductive and Inductive Learning for Knowledge Base Completion". PhD thesis. University of Illinois at Urbana-Champaign.

Wielinga, B.J.; Bredeweg, B.; Breuker, J.A. (1988). "Knowledge acquisition for expert systems". *Advanced Topics in Artificial Intelligence. 2nd Advanced Course*, ACAI '87, pp 96-124.

Wielinga, B.J.; Schreiber, A.Th.; & Breuker, J. (1992). "KADS: a Modelling Approach to Knowledge Engineering". *Knowledge Acquisition* Vol. 4, No. 1 (March), pp 5-53.

Wilkins, David Chester (1987). "Apprenticeship Learning Techniques for Knowledge Based Systems". PhD dissertation. University of Michigan.

Wirfs-Brock, Rebecca (1990). *Designing Object-Oriented Software.* Englewood Cliffs, NJ: Prentice-Hall.

Wrobel, Stefan (1988). "Design Goals for Sloppy Modeling Systems". *Int. J. Man-Machine Studies*, Vol. 29, pp 461-477.

Xiafeng, Li (1991). "What's so Bad about Rule-Bases Programming?". *IEEE Software*, Vol. 8, No. 5, pp 103-105.

Yih, Yuehwern (1988). "Trace-Driven Knowledge Acquisition for Expert Scheduling System". PhD dissertation. University of Wisconsin.

Yoneda, T.; Minagawa, M.; & Kakazu, Y. (1992). "Development of an expert system for metal shaft grinding a genetic approach". *IFIP Trans. B, Appl. Technol.* (Netherlands) Vol. B 3, pp 665-672.

Zhang, Wen Ran; Chen, Su Shing; & Berzek (1989). "Pool2: A Generic System for Cognitive Map Development and Decision Analysis". *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. 19, No. 1 (Jan/Feb), pp 31-39.

Index

A

absentees 174

accuracy 94

ACT* theory 10, 64, 67, 114

Adelman, Leonard 90, 117

Agarwal, Ritu 118, 204

Anderson, Barry 198

Anderson, John R. 10, 57, 64, 67, 126

Anjewierden, Anjo 11, 17, 39

AQ11 algorithm 104

AQUINAS 7, 37, 71

  automated guidance facility 15

Arblaster, A.T. 107

attributes

  eliciting 75, 177

  redundant 195, 207

B

Baddeley, Alan 59, 62

Bain, William 32, 53

Bartlett, F.C. 62

Bell, Michael 107

Berwick, Robert 42, 90

Birmingham, William 3, 92

Bischel, Martin 44, 45

Bonferroni procedure 178

Boose, John 7, 32, 33, 40, 71, 160

Bower, G.H. 62

Bowyer, John 4

Bradshaw, Jeffrey 32

Bratko, Ivan 89

Bray, James 174

Breiman, L. 85

Breuker, Joost 3, 6, 7

brittleness of knowledge base 100

Buchanan, Bruce 18, 96, 203

Buchberger. E. 42

Buffa Elwood 4

Buntine, Wray 50

Burton, Mike 82, 85, 105, 110, 116, 130

C

Camp, C.J. 10

card sort 117
CART algorithm 41
certification for operation 86
CGEN 92
Chadha, Sanjay 208
Chandrasekaran, B. 203
Channier, T. 20
Chapanis, Alphonse 95
Chilausky, R.L. 46
Clancey, William 3, 14, 15, 27
classes
  eliciting 73
Claxton, Guy 60
cognitive mapping 22
Cohen, Gillian 61, 67
Cohen, Paul 92
Collins, Edward 89
comparison standards 89
completely randomised
  designs 118
computing costs 98
concept learning system 46
confused pairs 39
congruence of representation 111
Connell, N.A.D. xvi, 98
Connerton, Paul 63
consciousness 59
constructs 34
  range of convenience 74
COPE 22
Crawford, Stuart 41
cultural knowledge 62

D

Dalton. Peggy 61
Davis, J. Steve 205
Davis, Randall 93
Dechter, R. 133, 189
Deffner, G. 115
Delphi method 89
DENDRAL 7
dependent variables 125
Dhaliwal, Jasbir 82
diagnostic accuracy 94
Diederich, Joachim 16, 46
Dilger, W. 6
dyadic elicitation 195

E

Eden, Colin 22

efficacy 81
efficiency 82, 198
elements 34
  eliciting 73
eliciting 74
eliciting classes 73
eliciting elements 73
eliciting ratings 80
Ellis, Andrew 68
EMYCIN 33
episodic memory 61, 62
Ericsson, K.A. 16
ETS 107, 160
evaluation 86, 140, 180
  by novices 89
  by panel 89
exemplars 29, 180
experiment subjects 129, 172, 174
explanation-based learning 42
external validity 201

F

factorial designs 117
Farley, B.G. 44
Fayyad, Usama 47
Feigenbaum, Edward 2, 96, 101
Fisher, R.A. 85
FMS-Aid 37, 71
focal theory 209
FOIL 48
Ford, K.M. 74
formal interviews 117
Forsyth, Richard 4, 42
Forsythe, Diana 12, 89, 107
forward scenario simulation 29
frames 62
Freundlich, Yehudah 2, 7

G

Gaines, Brian 32, 37
Gammack, John 39, 73, 79, 149, 207
Gams, Matjaz 51
Garg-Janardan, Chaya 32, 36, 144, 150
Gaschnig, John 86
genetic algorithms 42
Gettig, Gary 20
Gilmartin, Kevin 59
Ginsberg, Allen xii, 87, 168

Gruber, Thomas 18, 29

H

Hall, Robert 134
Hart, Anna 2, 9, 16, 33, 159
Hayes-Roth, Frederick 87, 89, 203
heuristic knowledge 6
Hofstede, Geert 63
Holland, John 42
human costs 98
Hunt, E.B. 46
hypermedia 16, 17
hypertext 16, 17
hypotheses 124

I

ID3 algorithm 50
ill-defined quantifiers 115
incremental learning 41
independent variables 125
inference matrix 78
intelligibility 99
interactions 174
interviewing 12

J

Johnson-Laird, P.N. 58, 63
Jones, Sue 22
Joseph, A.T. 2

K

KADS methodology 6, 211
KAM 20
KAMSE 29, 75, 84, 143
  subfunctions 152
  tools
    1st-Class 52, 170
KEATS 107
Keen, 195
Kelly, George 32, 35
Keppel, Geoffrey 128, 132, 174, 201
KITTEN 71, 157, 160
Kitto, Catherine 3, 15
Klein, Jonathan xvi, 22, 126
knowledge compilation 67
knowledge composition 67
knowledge engineers 208
Kolodner, Janet 11, 53, 57, 61
Kononenko, Igor 214
Koontz, Harold 4

Kors, J.A. 89
Kraemer, Helena 130
KREME 107
KRITON 16, 46
KSSO 37

L

labour-intensive process 204
laddered grid 117
LaFrance, Marianne 13
Latta, Gail 194
Lenat, Douglas 93
LEW 41
Liebowitz, Jay 86
Lippmann, Richard 45
Lisp 170
long-term memory 60
Lundell, James 11, 29, 111, 209
Lydiard, T.J. 94

M

machine induction 46
MACSYMA 8
Mandler, George 59, 61
MANOVA 174
manual induction 49
Marcus, Sandra 3
Marr, D. 68
Marshall, Judi 63
Matheson, Douglas 128
Matwin, Stan 41
McClanahan, Anne 31, 47
measuring mental effort 126
memory
  episodic 61, 62
  long-term 60
  semantic 62
  short-term 59
mental effort 126, 200
methods of knowledge
  acquisition 12
Michalski, R.S. 46, 95, 99, 103, 129, 185
Michie, Donald 100, 133, 189
milking of triads 149
Mingers, John 50
Minsky, Marvin 45, 62
missing cases 174
Mitchell, Tom 42
model-based systems 6, 100

Morecroft, J.D.W. 22
Morik, Katharina 30
Morris, Peter 61, 67
Muggleton, Stephen 92, 105
MUGOL 105
multivariate analysis 174
Mumford, Enid 97
Murphy, G.L. 67
Musen, Mark 16
MYCIN 71, 165

N

natural language 18
Naylor, Chris 9
Neale, Ian 7, 11, 12, 16, 29
neural networks 43, 67, 71, 111
Newell, Allen 59, 197
NEXTRA 157
Nickerson, R.S. 10
Noble, David 62
novices 89, 111
  evaluation by 89

O

O'Leary, Daniel 55
operation, certification for 86

P

Papert, Seymour 45
Partridge, Derek 8, 9
PED 17
PEGASUS 33
perceptrons 43, 44
PLANET 37
Politakis, Peter xii, 47, 168
Pool2 22
Posner, Michael 59, 126
power of the data 130
process of knowledge
  acquisition 4
PROLOG 19, 48
PROTEGE 16
protocol analysis 16, 114, 117
prototypes 29, 35
pseudo-English 18, 116
psychometric testing 118
PUPS 67

Q

quality analysis
  See evaluation

Quinlan, J.R. 47, 49, 50, 51, 162

R

Rada, Roy 44
range of convenience 74
Rappaport, A.T. 160
Rasmussen, J. 97, 107
ratings
  conversion to attribute
    values 151
  eliciting 80
Rau, L.F. 20
redundant attributes 195, 207
refinement 115
refinement of knowledge bases xii
Reimer, Ulrich 21
Reitman Olson, Judith 12
repeated-measures designs 114
repertory grid 32
  modifications 196
  tools
    AQUINAS 33, 37
    FMS-Aid 37
    KITTEN 157
    KRITON 16, 46
    KSSO 37
    PEGASUS 33
    PLANET 37
    SCENIC 143
representational congruence 111
ROGET 107
Rosenblatt, Frank 43, 44
Roskar, Egidija 18, 28
Rugg. G. 123
Rumelhart, D.E. 62
Ryle, 195

S

Sager, Naomi 19
Saito, Kazumi 85
SALT 15
scaling up 201
scenario simulation 29
SCENIC 130, 131, 143, 157, 167, 188, 205
Schank, Roger 19, 62
Schmidt, Gabriele 21
Schreiber, A. 3
Schroder, O. 70
scripts 62
secondary task method 126
Seitz, Peter 44, 45
semantic memory 62
semantic networks 17

Shadbolt, Nigel 82, 85, 105, 110, 116, 130
Shaw, Michael 2
Shaw, Mildred 32, 33, 71, 160
short-term memory 59
similarity-based learning 46
Simon, Herbert 59
single-factor designs 118
size of a knowledge base 99
sloppy modelling 23
Smith, Stephen 42
sociopathic knowledge bases 8
Spackman, Kent 99
standards of comparison 89
Steels, Luc 96
Stevens, Antony 89, 95
Stevens, James 132
Stevenson, R.J. 114
Storrs, Graham 17
Stout, Jeffrey 3, 15
structured interviews 118
subjective organisation 79
subjects 129, 172, 174
  number needed 130
Sykes, David 6, 100
Szolovits, Peter 100, 101

T

tacit knowledge 10
Tanniru, Mohan 118
Tanyi, E. 19
Taylor, Talbot 136
thinking-aloud protocols 114
Tompsett, C.P. 15
triads, milking 149
Tulving, Endel 60, 79, 197

U

unstructured interviews 118
user interface 107, 144
user involvement 208

V

validation
  See evaluation
validity, external 201
variables 125
variance, multivariate analysis 174
verbal protocol analysis 16
verification
  See evaluation

VIE LANG 42
Vinze, Ajay 86, 97
Voss, Angi 6

W

Waterman, Donald 4, 87
Weiss, Sholom 4, 85
Weizenbaum, Joseph 8
Welch, J.C, 126
Whipple, C. 7
Whitehall, Bradley 42, 207
Wielinga, Bob 3
Wilkins, David 8, 41, 82, 203
WIT 21
within-subject analysis 128
Wrobel, Stefan 23

X

Xiafeng, Li 205

Y

Yih, Yuehwern 90
Yoneda, T. 42

Z

Zhang, Wen Ran 22