AI and Simulation 1988
©1988 by The Society for Computer Simulation
ISBN 0-911801-30-8

Knowledge acquisition techniques for knowledge-based systems

Mildred L. G. Shaw
Dept of Computer Science
University of Calgary
Calgary, Alberta
Canada T2N 1N4

ABSTRACT

Growing recognition of the significance of knowledge-based computing systems has focused attention on processes of knowledge acquisition and transfer. Commercial applications of expert systems are being impeded by the knowledge engineering bottleneck and have led to the development of rapid prototyping tools. These have proved practically useful but the range of application of existing tools is limited, and it is not clear what limitations are inherent in the prototyping tools, what arises from the shells and their knowledge representation and inferencing procedures, and what arises from our lack of understanding of the underlying processes of knowledge acquisition and transfer.

This paper introduces the concept of knowledge support systems — integrated interactive knowledge acquisition systems and expert systems shells. A prototype knowledge support system that has been implemented is described with examples of some of the knowledge acquisition and application tools provided. It includes knowledge acquisition and transfer through interactive repertory grid elicitation; exchange techniques for understanding and agreement between experts; and different forms of analysis that have been widely used for comparing data between experts and for rapid prototyping of expert systems.

INTRODUCTION

Much successful work has been done in the area of knowledge representation, but knowledge acquisition and transfer has long been thought of as the bottleneck in the process of building expert systems. It was possible that knowledge engineering might develop as a profession on a par with systems analysis and programming, and that an initial shortage of skilled knowledge engineers would cause problems which would be overcome eventually as the profession developed. However, this scenario now appears less and less likely. There is certainly a shortage of knowledge engineers and problems in developing applications, but doubts have been cast on the notion that human labor is the appropriate solution to the knowledge engineering problem.

The technology is now in a mass-market situation where many organizations see the need for expert systems. This has led to a growth in demand that is far more rapid than the growth in supply of trained and experienced knowledge engineers. In addition, the role of the knowledge engineer as an intermediary between the expert and the technology is being questioned not only on cost grounds but also in relation to its effectiveness. Knowledge may be lost through the intermediary and the expert's lack of knowledge of the technology may be less of a detriment than the knowledge engineer's lack of domain knowledge. Full exploitation of the potential of expert systems depends on the development of rapid prototyping systems directly usable by experts with the knowledge engineer acting only as manager, not intermediary.

KNOWLEDGE SUPPORT SYSTEMS

Knowledge support systems can be seen as integrated interactive knowledge acquisition systems and expert systems shells. They are seen as having a broad scope, encompassing both aids to knowledge acquisition and transfer, knowledge representation, and support of human knowledge processes (Shaw and Gaines 1987). It is desirable that the tools in a knowledge support system are domain independent and directly applicable by experts without intermediaries. They should be able to access a diversity of knowledge sources including text, interviews with experts, and observations of expert behavior and be able to encompass a diversity of perspectives including partial or contradictory input from different experts. Moreover, the knowledge support system should be able to encompass a diversity of forms of knowledge and relationships between knowledge and be able to present knowledge from a diversity of sources with clarity as to its derivation, consequences and structural relations. Users of the knowledge support system should be able to apply the knowledge in a variety of familiar domains and freely experiment with its implications. It should make provision for validation studies and as much of the operation of the knowledge support system as possible should be founded on well-developed and explicit theories of knowledge acquisition, elicitation and representation. As the overall knowledge support system develops it should converge to an integrated system.

Clearly, these are general requirements and may not apply in particular cases. For example, some domain dependence may be appropriate for efficiency in specific knowledge support systems; some human intervention may be helpful or necessary when an expert is using a knowledge support systems. However, in general these requirements capture the key issues in knowledge support system design.

PLANET (Shaw 1982, Shaw and Gaines 1986) has been developed and widely used for some years for knowledge elicitation. This paper briefly reviews the techniques involved and describes the most recent developments of a comprehensive integrated rapid prototyping system, KITTEN, which attempts to satisfy the above design requirements for a knowledge support system.

RELATED WORK

ETS and Aquinas (Boose 1984, Boose and Bradshaw 1987) are related quite closely to PLANET and KITTEN, having been developed in parallel. All four systems use repertory grid techniques to elicit the expert's conceptual structures, and use different versions of ENTAIL (Gaines and Shaw 1986) to produce rules for an expert system shell.

MOLE (Eshelman, Ehret, McDermott, & Tan, 1987) is an expert system shell that has been used effectively to build heuristic classification systems. It interviews domain experts, exploits assumptions of exhaustiveness and exclusivity to determine the most likely candidates among competing hypotheses, and has many methods for analyzing and refining the knowledge base for consistency and adequacy. It distinguishes between three types of knowledge—covering, differentiating, and combining knowledge—in its problem-solving, and uses a system of "support values" to represent uncertainty.

CSRL (Bylander & Mittal, 1986) is another successful tool with a similar problem-solving approach, although it is difficult for domain experts to learn and use. Gruber and Cohen (1987) have developed a system called MUM that embodies architectural principles that facilitate knowledge acquisition. It allows experts to specify local evidence combining knowledge; however it is limited to a symbolic representation of uncertainty.

An innovative "learning apprentice" program, ODYSSEUS, has been developed by Wilkins (1987) to help experts refine and debug knowledge bases for the HERACLES heuristic classification shell.

However, its use is appropriate only during the knowledge acquisition "end-game"; that is, it requires that a reasonable knowledge base has already been created.

The following section describes the work on KITTEN, a knowledge support system that draws on many concepts and techniques for knowledge engineering to begin to encompass some of the requirements discussed above, attempting to relating them all through the theoretical base of personal construct psychology, and building a workbench of tools around a common database.

KITTEN: A KNOWLEDGE SUPPORT SYSTEM

KITTEN: Knowledge Initiation & Transfer Tools for Experts and Novices consists of a: knowledge base; various analytical tools for building and transforming the knowledge base; and a number of conversational tools for interacting with the knowledge base. The KITTEN implementation is written in Pascal and currently runs on a network of Apollo workstations. The KITTEN structure is best understood by following sequences of activity that lead to the generation of a rule base and its loading into an expert system shell.

A typical sequence is text input followed by text analysis through TEXAN which clusters associated words leading to a schema from which the expert can select related elements and initial constructs with which to commence grid elicitation. 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 (Gaines & Shaw 1986).

An alternative route is to monitor the expert's behavior through a verbal protocol giving information used and decisions resulting and analyze this through ATOM which induces structure from behavior and again generates production rules (Gaines 1977).

These two routes can be combined. KITTEN attempts to make each stage as explicit as possible, and, in particular, to make the rule base accessible as natural textual statements rather than technical production rules. The expert system shell being used in KITTEN currently is Nexpert (Roy 1986) which gives a variety of textual and graphical presentations of the rule base enabling the expert to see the impact of different fragments of knowledge.

The group problem-solving component of KITTEN is particularly important because it goes beyond the stereotype of an **expert** and **users**, and allows the system to be used to support an interactive community in their acquisition and transfer of knowledge and mutual understanding. The SOCIO analysis allows members of a community to explore their agreement and understanding with other members, and to make overt the knowledge network involved (Shaw 1980, 1981, 1987a).

The KITTEN implementation is an initial prototype offering a workbench with minimal integration of the knowledge base, but each of tools has already proven effective, and their combination is proving very powerful in stimulating experts to think of the knowledge externalization process from a number of different perspectives. The following sections describe and illustrate some of the tools.

Elicitation

The following screens show KITTEN eliciting data on staff appraisal using techniques that contrast with the prompt/response style of other systems (Shaw & Gaines 1986). When starting a new topic the working window must first be completed. A new user will have a topic with a specific purpose which should be entered in the appropriate box, and kept in mind as the interaction proceeds. In this example Bill is construing **managers** to see which of the ones he knows are **effective** and why. Figure 1 shows the screen after the user has entered his name, purpose, and the type of element that he will be using.

The next thing for Bill to do is to think of some managers and add these as elements. He does this by clicking on the **ELEMENTS** button, then typing in the names of the managers he would like to think about one on each line. Bill knows that about six is the best number to start with, and that less than three will make triadic elicitation impossible. If a typing error is made he can just go back and correct it by selecting that name and clicking on **Edit Name** as shown in Figure 2.

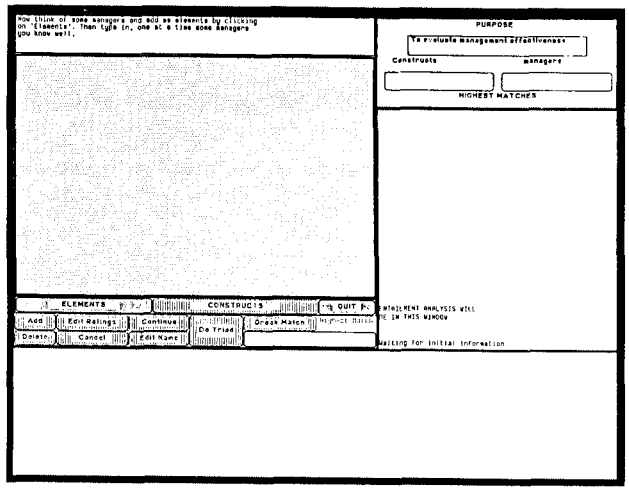

Fig 1. Initial screen

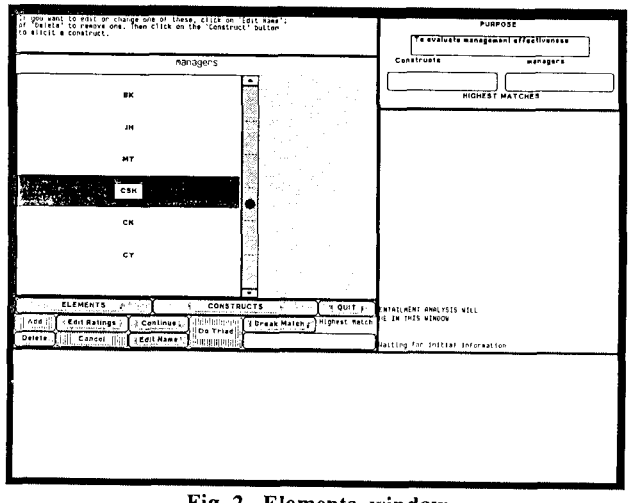

Fig 2. Elements window

As Bill is just starting his grid, he decides to elicit a construct from a triad, so he clicks on the **CONSTRUCTS** button, then on the **Do Triad** button. Bill sees the three elements in the triad, and decides that CY is the one which is different from CN and MT. This is shown in Figure 3.

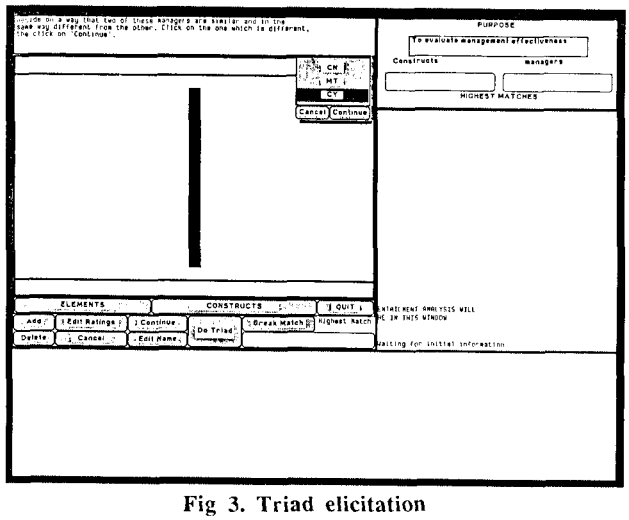

Fig 3. Triad elicitation

Now Bill has to decide why he thinks CN and MT are alike and different from CY. He types in the description of the pair, **intelligent**, on one pole of the construct as shown in Figure 4.

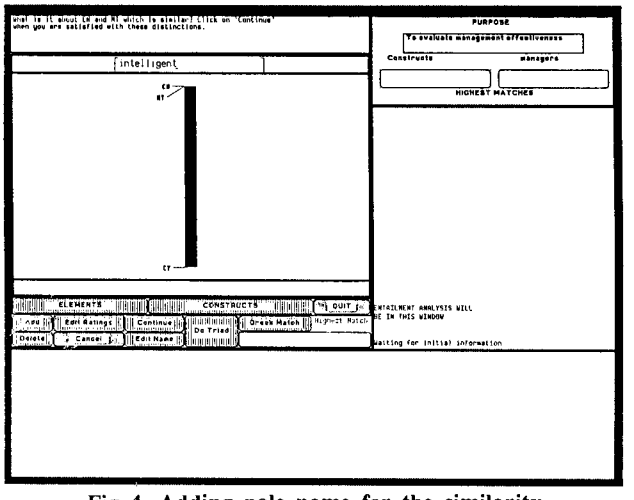

Fig 4. Adding pole name for the similarity

Then Figure 5 shows how he names the other end of the construct. That is, the way in which CY is different is that he is **dim.**

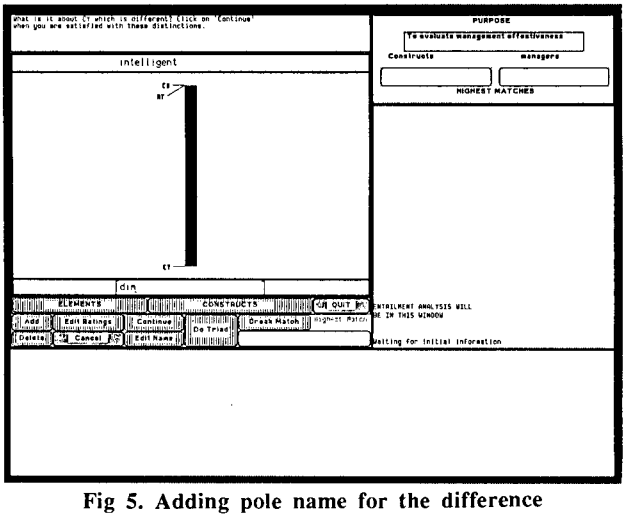

Fig 5. Adding pole name for the difference

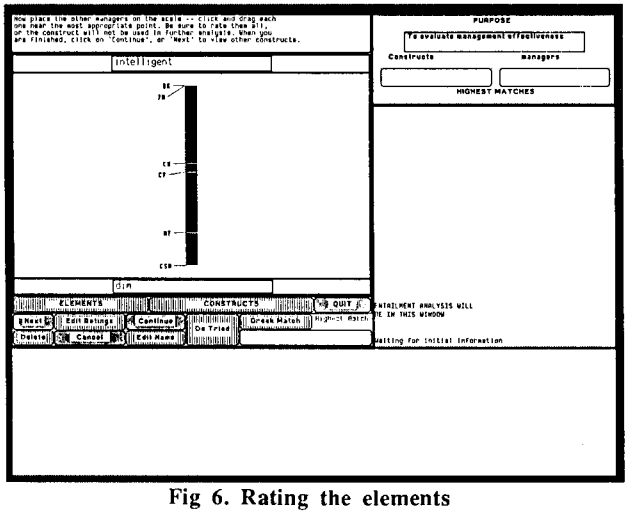

Fig 6. Rating the elements

This defines Bill's scale from **intelligent**, to **dim.** He now has to rate all his other elements on this scale too. If he finds that his original triad CN, MT and CY need to be moved to accommodate the others he can do that. Figure 6 shows that he moves CY up to the middle of the scale, and puts CSH at the bottom. He also moves CN and MT as he re-thinks how the scale should be. Now BK and JH are at the top, CN and CY towards the centre, and MT toward the bottom. However, MT is not quite so dim as CSH.

Now Bill has got his first construct. He now continues adding several more constructs from triads before he decides to look at high matches. Figure 7 shows the first construct recorded in the bottom window, and the dragging of an element onto the construct bar for the second one.

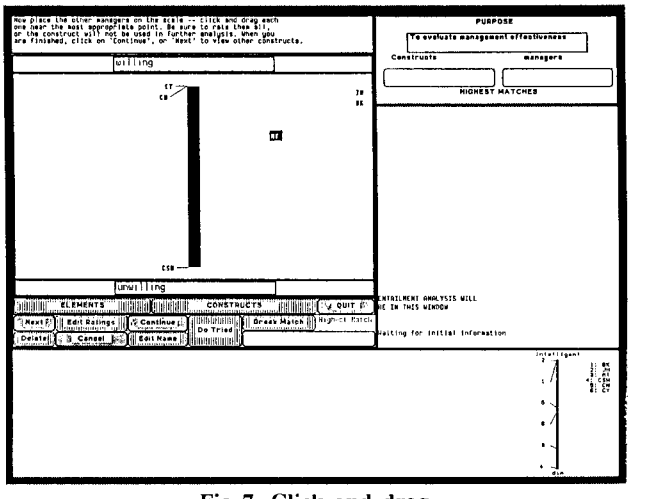

Fig 7. Click and drag

After the next construct shown in Figure 8, Bill notices that the top right window is showing a high match between two of the managers. He goes to the **ELEMENTS** window by clicking on that button, and chooses to split them by clicking on **Break Match.** This takes him to a new screen (Figure 9) showing which elements are matched, and placing one at either end of the construct bar. He then names the poles and rates the other elements on this new construct in the usual way. He also notices the first entailment in the right-hand window: dim implies unwilling, but realises that it is formed on very little data and chooses to ignore it for now.

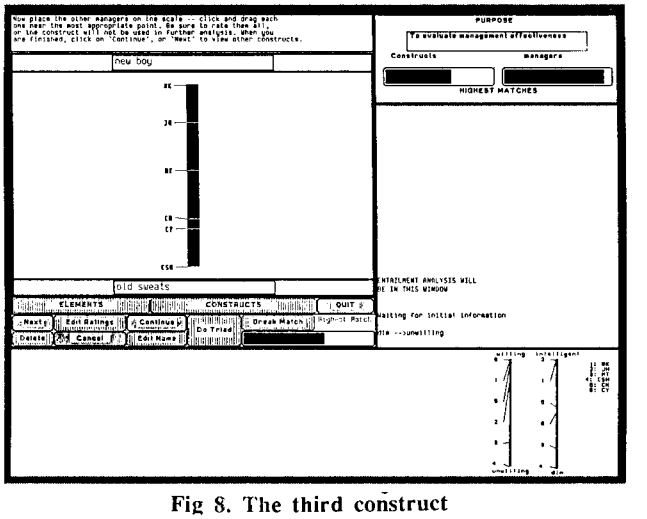

Fig 8. The third construct

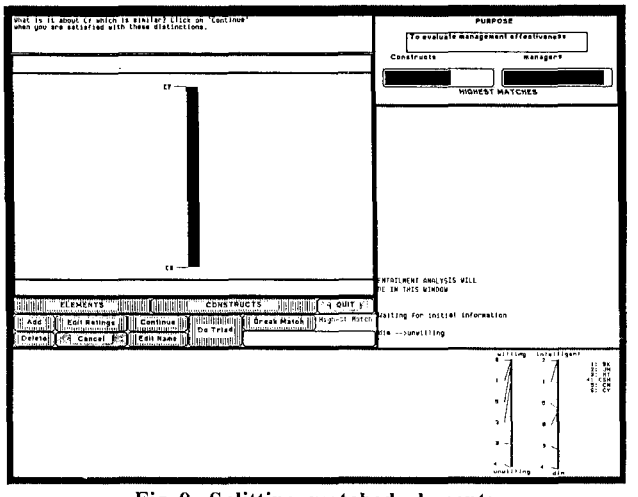

Fig 9. Splitting matched elements

Bill then looks at the construct match by clicking on **Break Match** box when he is in the **CONSTRUCTS** window. This takes him to a new screen showing which constructs are matched, shown in Figure 10. This screen also shows the constructs elicited so far in the bottom window, and a new set of entailments in the right-hand window.

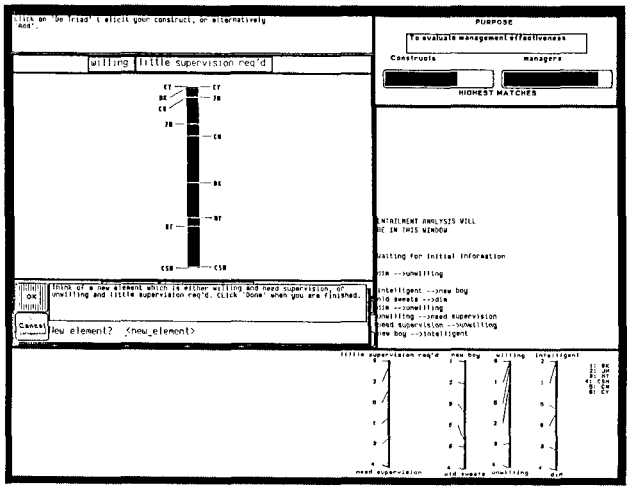

Fig 10. Construct match

Bill decides to add an element so he types in a new manager **NTM** (Figure 11). Now he will have to rate **NTM** on each of his constructs in turn.

He then has to go through each construct already elicited, adding NTM on to each scale in turn where Bill thinks he best fits. He does this by clicking and dragging that manager on the construct, then clicking on the **Next** button to get the next construct until he has done them all. Figure 12 shows one of these. Note that he may still choose not to rate NTM on opposite ends of the two matched constructs, but if he does not then the constructs will still be highly matched.

Figure 13 shows the constructs Bill has got so far. If he wishes to change the pole name of any one, or edit the ratings he could click on that one and then on the **Edit Name** or **Edit Rating** button. It also shows that the new manager NTM has been added to all the constructs elicited previously, as shown in the bottom window. Bill sees that a new set of entailments has been produced in the right-hand window.

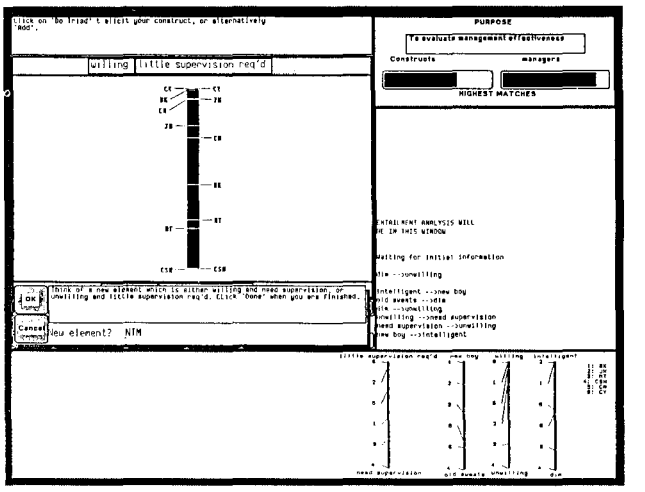

Fig 11. Adding an element to break the construct match

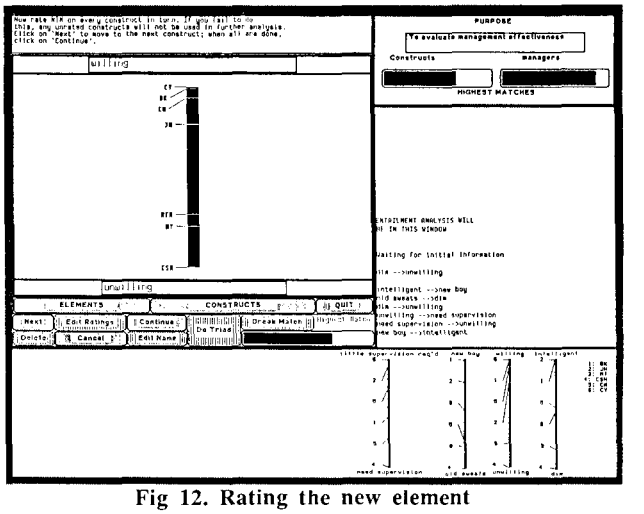

Fig 12. Rating the new element

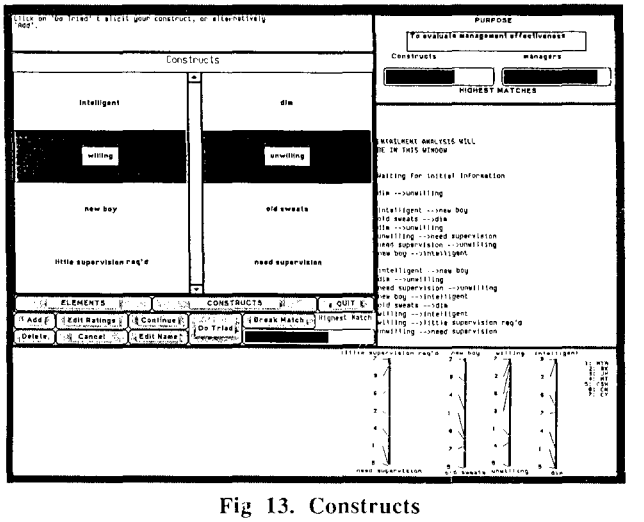

Fig 13. Constructs

Now Bill wishes to see how he has rated JH on each construct. He does this by going to the bottom window and clicking on the name JH. The number representing this manager is the highlighted on each construct, showing the relative positions which can be seen in Figure 14.

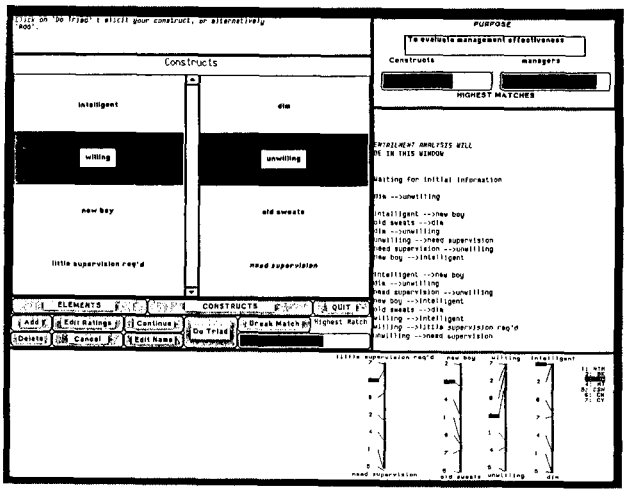

Fig 14. Highlighting an element

Figure 15 shows him looking at the positions of another manager CN. Any one of the elements can be highlighted in this way.

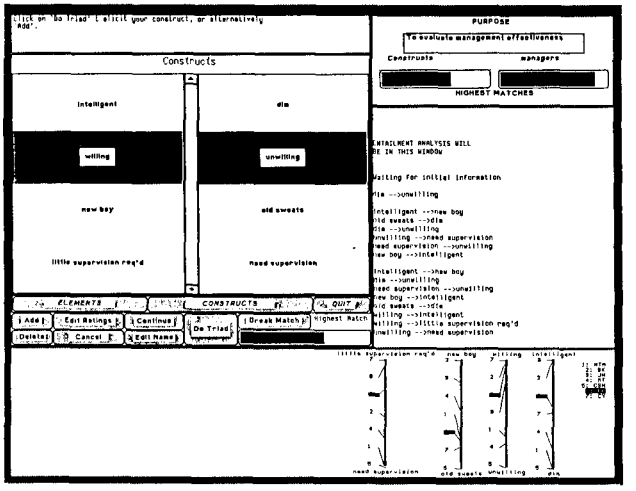

Fig 15. Highlighting another element

Figures 16 and 17 show how Bill can add a new manager anytime he chooses by clicking first on the ELEMENTS button to bring up the screen in Figure 16, and then just typing in the name of the new element. In Figure 17 the manager is listed with all the others.

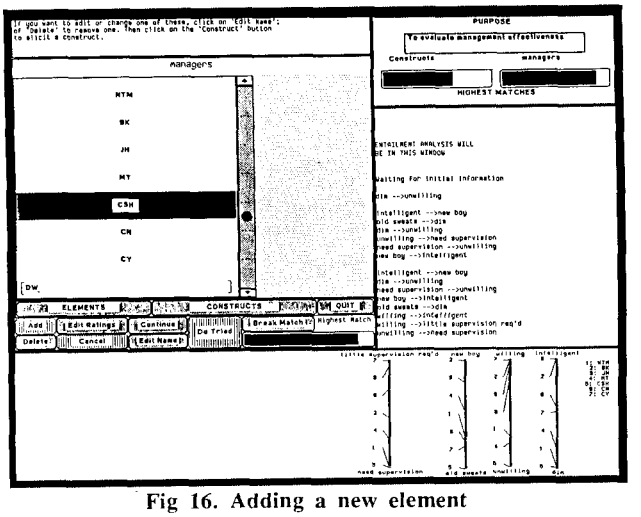

Fig 16. Adding a new element

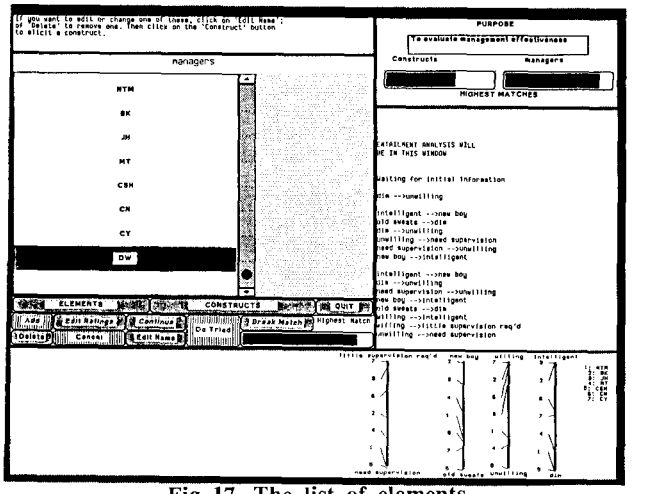

Fig 17. The list of elements

Bill continues to elicit his grid with the options available as described. At any time he can add or delete elements or constructs, use triads or just type in names and ratings. When he chooses to finish he can **QUIT** the elicitation by clicking on the button in any window. The grid will then be stored so that he can print it out or analyse it.

Cluster Analysis

Repertory grids in themselves encode information about an expert's way of looking at the world. This information can be used in its own right for some purposes since it is an aid to remembering the basis for decisions and actions. It can also be analysed in a variety of ways to bring out possible underlying structures, or construct systems, in the expert's world view and its relationship to those of others. There are a number of forms of analysis that are widely used for different purposes and KITTEN offers all the commonly used techniques plus new developments in recent years. What form of analysis should be used in a particular case is partly a matter of personal preference and partly a matter of purpose. Comparisons have been made in the literature of different analyses with the same data (Shaw 1981). In exploring the use of repertory grids for knowledge engineering in a specific domain it is worth repeating such comparisons with familiar data to determine what are the most applicable analyses and presentations.

The FOCUS algorithm (Shaw 1980) is a distance-based hierarchical cluster analysis technique that sorts the constructs into a linear order such that constructs closest together in the space are also closest together in the order. It provides a hierarchical clustering of an expert's construct system that preserves the data elicited from him so that the sources of the analysis are evident and can be discussed. Standard principal component analysis techniques give a non-hierarchical cluster analysis based on principal components that can be used to gauge the major dimensions along which an expert is making distinctions.

Entailment Analysis

The entailment analysis of a repertory grid treats each pole of a construct as a fuzzy predicate to which the elements have degrees of membership given by their ratings, and induces the logical implications between these predicates. ENTAIL rank orders entailments in terms of the uncertainty reduction they induce in the distribution of the data, and hence tends to reject spurious entailments (Gaines & Shaw 1986).

Interaction with Nexpert enables the expert to see the derived rules in action. He can determine their consequences with test data, analyze new hypothetical cases, and see the inter-relations between rules presented graphically. The logging and explanation facilities of Nexpert enable him to track down spurious inferences that may arise with the rules derived by ENTAIL, or proper inferences that are missing. He can then edit the rules and test the revized system using Nexpert's facilities (Shaw 1987b).

SOCIO: Mutual Agreement and Understanding Analysis

It has already been emphasized that knowledge elicitation methodologies to cope with a group of experts as well as the individual is needed. Much expertise only resides within the social context of cooperating individuals and requires elicitation across the group. The SOCIO analysis program supports group elicitation techniques in which the construct systems of a number of users are compared. Grids are elicited separately but then exchanged in two ways: a user can place elements on a colleague's constructs from his own point of view, and the analysis system then allows him to explore their agreement; or he can attempt to place them from his colleague's point of view and hence explore his understanding. This has been implemented in KITTEN, and studies are starting to determine the various ways to combine expertise from a number of experts.

VALIDATION OF A KNOWLEDGE SUPPORT SYSTEM

A three-stage model for KITTEN evaluation and validation has been discussed by Shaw and Woodward (1987). KITTEN has been evaluated against the first stage of the model and the results reported in the two domains of spatial mapping of physical surfaces and in trouble-shooting and maintenance of valves for oil and gas pipelines. Some preliminary results are described on validation experiments to show the extent to which this system can replace standard interviewing techniques. Some of the question asked were:

1) To what extent does an expert find the generated rules meaningful?
2) To what extent do other experts agree with the expert's ratings of her/his elements on her/his constructs?
3) Can the same constructs be obtained from the expert after a significant time lapse?
4) Can the same ratings of the elements on the constructs be obtained from the expert after a significant time lapse?

Intra-subjective validity for the rules, that is, whether the expert indicated the same significance to the rules as the ENTAIL algorithm, averaged around 88% and Inter-subjective validity ratings between experts averaged arounf 72%. Reliability measures for constructs ranged between 50% and 100% with the average around 85%, and for ratings were all around 86%. These are preliminary results, and there is a great deal more data to analyse.

CONCLUSIONS

Growing recognition of the significance of knowledge-based computing systems has focused attention on processes of knowledge acquisition and transfer. Commercial applications of expert systems are being impeded by the knowledge engineering bottleneck and have led to the development of rapid prototyping tools. These have proved practically useful but the range of application of existing tools is limited, and it is not clear what limitations are inherent in the prototyping tools, what arises from the shells and their knowledge representation and inferencing procedures, and what arises from our lack of understanding of the underlying processes of knowledge acquisition and transfer.

This paper has introduced the concept of knowledge support systems as integrated interactive knowledge acquisition systems and expert systems shells. A prototype knowledge support system that has been implemented on a network of Apollo workstations has been described with examples of some of the knowledge acquisition and application tools provided. It includes knowledge acquisition and transfer through interactive repertory grid elicitation; exchange techniques for understanding and agreement between experts; and different forms of analysis that have been widely used for comparing data between experts and for rapid prototyping of expert systems. Such systems offer the capability to expedite the prototyping stage of complex knowledge-based systems, motivating the experts to be closely involved in all aspects of the system development by giving them a supportive and comprehensible environment.

ACKNOWLEDGEMENTS

Financial assistance for this work has been made available by the National Sciences and Engineering Research Council of Canada. The KITTEN system is made available by the Centre for Person Computer Studies; the initial Apollo implementation is being carried out at the Knowledge Sciences Institute, University of Calgary.

REFERENCES

Boose, J.H. (1984). Personal construct theory and the transfer of human expertise. **Proceedings AAAI-84**, 27-33. California: American Association for Artificial Intelligence.

Boose, J.H. & Bradshaw, J.M. (1987). Expertise transfer and complex problems: Using AQUINAS as a knowledge acquisition workbench for knowledge-based systems. **International Journal of Man-Machine Studies,** in press.

Bylander, T. & Mittal, S. (1986). CSRL: A language for classificatory problem-solving and uncertainty handling, **AI Magazine** (August).

Eshelman, L., Ehret, D., McDermott, J. & Tan, M. (1987). MOLE: A tenacious knowledge acquisition tool. **International Journal of Man-Machine Studies,** in press.

Gaines, B.R. (1977). System identification, approximation and complexity. **International Journal of General Systems, 3**, 145-174.

Gaines, B.R. & Shaw, M.L.G. (1986). Induction of inference rules for expert systems. **Fuzzy Sets and Systems**, 8(3), 315-328 (April).

Gruber, T. & Cohen, P. (1987). Design for acquisition: Principles of knowledge system design to facilitate knowledge acquisition. **International Journal of Man-Machine Studies,** in press.

Roy, J. (1986). Expert systems in Nexpert. **MacTutor**, 2(2), 48-51 (February).

Shaw, M.L.G. (1980). **On Becoming a Personal Scientist**. London: Academic Press.

Shaw, M.L.G. (1981). Conversational heuristics for eliciting shared understanding. Shaw, M.L.G., Ed. **Recent Advances in Personal Construct Technology**. pp. 31-44. London: Academic Press.

Shaw, M.L.G. (1982). PLANET: some experience in creating an integrated system for repertory grid applications on a microcomputer. **International Journal of Man-Machine Studies, 17**(3), 345-360.

Shaw, M.L.G. (1987a). An interactive knowledge-based system for group problem-solving. **IEEE Transactions on Systems, Man and Cybernetics**, (to appear).

Shaw, M.L.G. (1987b) Knowledge support systems. Luker, P.A. & Birtwistle, G., Eds. **Simulation and AI**, 18(3) 58-63 (January). San Diego, California: Society for Computer Simulation.

Shaw, M.L.G. & Gaines, B.R. (1986). Interactive elicitation of knowledge from experts. **Future Computing Systems, 1**(2), 151-190.

Shaw, M.L.G. & Gaines, B.R. (1987). Techniques for knowledge acquisition and transfer. **International Journal of Man-Machine Studies,** in press.

Shaw, M.L.G. & Woodward, J.B. (1987) Validation of a knowledge support system. **Proceedings of Workshop on Knowledge Acquisition for Knowledge-Based Systems**, (in press) (October). Banff, Canada.

Wilkins, D.C. (1987). Knowledge base debugging using apprenticeship learning techniques. **International Journal of Man-Machine Studies,** in press.