Int. J. Man–Machine Studies (1987) **27**, 251–280

KITTEN: Knowledge initiation and transfer tools for experts and novices

MILDRED L. G. SHAW AND BRIAN R. GAINES

Department of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4

This paper gives a state-of-the-art report on the use of techniques based on personal construct psychology to automate knowledge engineering for expert systems. It presents the concept of knowledge support systems as interactive knowledge engineering tools, states the design criteria for such systems, and outlines the structure and key components of the KITTEN implementation. KITTEN includes tools for interactive repertory grid elicitation and entailment analysis that have been widely used for rapid prototyping of industrial expert systems. It also includes tools for text analysis, behavioral analysis and schema analysis, that offer complementary and alternative approaches to knowledge acquisition. The KITTEN implementation integrates these tools around a common database with utilities designed to give multiple perspectives on the knowledge base.

Knowledge support systems for automating knowledge engineering

Problems of knowledge engineering have been recognized since the early days of 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 to 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 decline in costs of both hardware and software support for expert systems has brought the technology into a mass-market situation far more rapidly than originally envisioned;
- This has led to a growth in demand for expert systems that is proceeding far more rapidly than the growth in supply of trained and experienced knowledge engineers;
- The declining costs of expert system technology are also making the expense of human labour in tailoring the technology for particular applications appear to be the dominating constraint and an excessive cost;
- A move towards a labor-intensive activity such as knowledge engineering is contrary to all trends in industry;
- In particular it is contrary to the trend towards automatic programming techniques in the computing industry;
- 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

0020-7373/87/090251 + 30$03.00/0

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.

The considerations of the previous section have heightened interest in the possibility of providing knowledge support systems (KSSs) to automate knowledge engineering as a process of direct interaction between domain experts and the computer. In 1980 we proposed that personal construct psychology (Kelly, 1955; Shaw, 1980) could provide foundations for expert systems, particularly in systems that combined interactivity with database access and expert advice to provide decision support, and gave examples of algorithms and programs that extracted entailment rules from repertory grid data (Gaines & Shaw, 1980). In 1983 we reported further enhancements of these techniques and a preliminary experiment to validate them empirically as a knowledge engineering technique for priming expert systems (Shaw & Gaines, 1983). This work led to industrial studies of the methodology applied to the development of expert systems: Boeing Computer Services (Boose, 1984, 1985, 1986) and Lockheed Software Technology Center (Wahl, 1986) have reported success in applications; and validation has been reported in a statistics domain (Gammack & Young, 1985).

This paper gives a state-of-the-art report on the use of techniques based on personal construct psychology to automate knowledge engineering for expert systems. It is based on four areas of advance since the previous paper:

- Improved techniques for the derivation of rules from repertory grid data which give: a natural knowledge representation for uncertain data combining fuzzy and probabilistic logics; and an information-theoretic measure of the significance of a derived rule (Gaines & Shaw, 1986*a*);
- Widespread applications experience in prototyping expert systems using the methodology (Boose, 1985; Gaines & Shaw, 1986*b*);
- Improved interactive techniques for on-line knowledge engineering from groups of domain experts interacting through a computer network (Shaw, 1986; Shaw & Chang, 1986);
- The KITTEN implementation, a knowledge engineering workbench that provides next generation KSS facilities including textual analysis, induction of models from behavior, multi-level and multi-expert repertory grid elicitation, and hierarchical construct laddering, to automate knowledge engineering for a wide range of problem domains.

Personal construct psychology

Kelly developed a systemic theory of human cognition based on the single primitive of a construct, or dichotomous distinction. For an individual, constructs are:

> transparent templets which he creates and then attempts to fit over the realities of which the world is composed (Kelly 1955).

He proposes that all of human activity can be seen as a process of anticipating the future by construing the replication of events:

> Constructs are used for predictions of things to come, and the world keeps rolling on and

revealing these predictions to be either correct or misleading. This fact provides a basis for the revision of constructs and, eventually, of whole construct systems (Kelly 1955).

Hence his psychological model of man is strongly epistemological and concerned with the way in which man models his experience and uses this model to anticipate the future. The anticipation may be passive, as in prediction, or active, as in action.

Kelly developed his theory in the context of clinical psychology and hence was concerned to have techniques which used it to by-pass cognitive defenses and elicit the construct systems underlying behavior. This is precisely the problem of knowledge engineering noted above. His repertory grid (Shaw, 1980) is a way of representing personal constructs as a set of distinctions made about elements relevant to the problem domain. In clinical psychology this domain will often be personal relationships and the elements may be family members and friends. In the development of expert systems the elements will be key entities in the problem domain such as oil-well sites or business transactions.

Repertory grids have been widely used: in clinical psychology (Shepherd & Watson, 1982); to study processes of knowledge acquisition in education (Pope & Shaw, 1981); and to study decision making by individuals and groups in management (Shaw, 1980). PLANET (Shaw, 1982) is an integrated suite of programs that operationalizes Kelly's work and may be used for the interactive elicitation and analysis of repertory grids. These programs have been widely used internationally in clinical psychology, education and management studies (Shaw, 1981), and this paper describes their application to knowledge engineering for expert systems.

Kelly's personal construct psychology is important because it develops a complete psychology of both the normal and abnormal, which has strong systemic foundations. In the long term these foundations may be more important to knowledge engineering than the techniques currently based on them. However, this paper concentrates on the repertory grid as a technique for eliciting information from an expert.

Repertory grids

A repertory grid is a two-way classification of data in which events are interlaced with abstractions in such a way as to express part of a person's system of cross-references between his personal observations or experiences of the world (elements), and his personal constructs or classifications of that experience.

The elements are the things which are used to define the area of the topic, and can be concrete or abstract entities. For example, in the context of: inter-personal relations the elements might be people; attitudes to life the elements might be critical events; job change the elements might be careers; expertise about metal joining the elements might be types of rivet; expertise about medical diagnosis the elements might be symptoms. Before choosing the set of elements, the user must think carefully about the area of the topic and relate the elements to his purpose. The elements should be of the same type and level of complexity, and span the topic as fully as possible. It is usual to start with about six to twelve elements.

The universe of discourse is determined by the elements. The elements originally suggested by Kelly in his work as a psychotherapist were role titles such as: Self, Mother, Father, Best Friend, Threatening Person, Rejected Teacher. This has been carried over into industry with such role titles as: Myself, My Boss, My Boss's Boss, Subordinate, Person Likely to Get On, Person Not Likely to Get On, and so on. The subject in both cases is required to supply names of individuals well known to her/him to fit these and other roles as closely as possible. When choosing elements care must be taken to ensure that each one is well known and personally meaningful to the subject. Each element must be significant to the person in the context of the particular problem.

The constructs are the terms in which the elements are similar to or different from each other. Each construct therefore has two poles, each of which has a meaning with respect to its opposite. Any construct or dimension of thinking which is important to the subject is a valid construct. For example, to distinguish between people by saying that x and y are **blue-eyed** whereas b and c are **brown eyed** may be trivial, and not concerned with the important qualities of x, y, b, and c. However, if you are an eye specialist concerned with prescribing tinted contact lenses, this may be a significant construct. Thoughts and feelings, objective and subjective descriptions, attitudes and rules-of-thumb all constitute valid constructs. The verbal description of the construct and the labelling of the poles need not be a publically agreed meaning in the outside world, but only a memory aid to the thinking process. The mapping of the elements onto the constructs produces the two-dimensional grid of relationships.

ELICITING CONSTRUCTS BY TRIADS

The most common method used for eliciting a construct is the minimal context form or triad method. The elements are presented in groups of three—this being the least number which will produce both a similarity and a difference. The subject is asked to say in what way two are alike and thereby different from the third. This is the emergent pole of the construct. The implicit pole may be elicited by the difference method (in what way does the singleton differ from the pair) or by the opposite method (what would be the opposite of the description of the pair).

As an example, thinking of the three Artificial Intelligence books *Handbook of AI*, *Winston's AI*, and *AI Applications for Business* in what way are two alike and thereby different from the other one? We might first of all say that *Handbook of AI*, and *AI Applications for Business* are alike since they are **multi-authored**, whereas

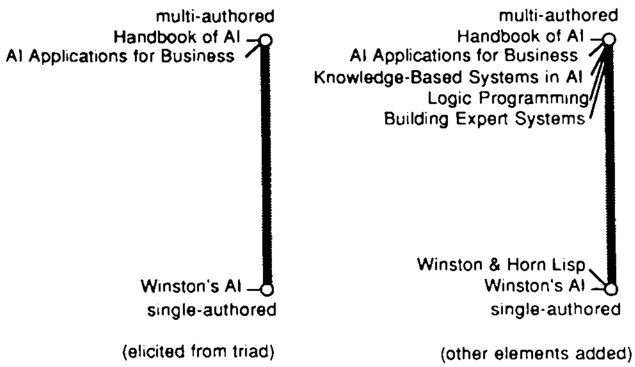

FIG. 1. Triadic elicitation of a construct about AI books.

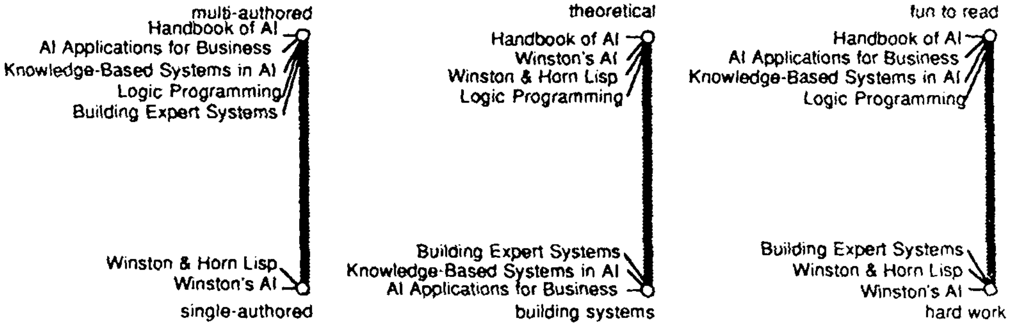

FIG. 2. Three constructs applied to AI books.

Winston's AI is **single-authored**. This is, then, the first construct with its two poles or opposite descriptions. Now all the elements in the set must be rated on this dimension as either 1 being **multi-authored**, or 2 being **single-authored** as shown in Fig. 1. This also shows the significance of the term personal in personal construct since it would not obviously be a publicly agreed description that *Winston and Horn's LISP* is single-authored whereas Davis and Lenat's book is multi-authored. In this case it reflects a single concerted effort as opposed to more than one topic.

Then the second and subsequent constructs are elicited in exactly the same way using different triads each time. Figure 2 shows the grid of Fig. 1 with a further two constructs elicited. The third construct shown here illustrates that constructs can be factual, imaginary, emotional, or whatever is important to the person generating the grid.

ELICITING CONSTRUCTS USING RATING SCALES

A scale allowing more distinctions than the pair 1 and 2 may be used as required. If a 1 to 5 scale is used then the above example might become the grid shown in Fig. 3. Thus, in this case, the third construct means that *Logic Programming* and *Knowledge-Based Systems in AI* are considered the most **fun to read** books, *Winston's AI* is both **fun to read** and **hard work**, and *Building Expert Systems* and *Winston & Horn LISP* are the most **hard work**.

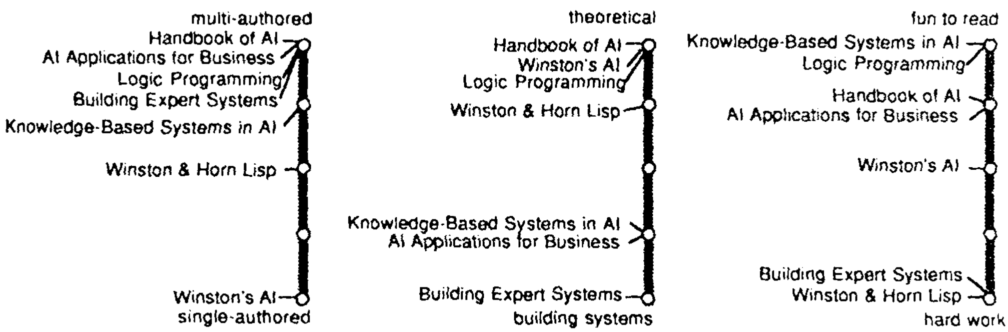

FIG. 3. Three multi-point scale constructs applied to AI books.

1	1	2	3	4	5	6	7	5
multi-authored	1	5	1	3	2	1	1	single-authored
theoretical	1	1	4	2	4	1	5	building systems
fun to read	2	3	2	5	1	1	5	hard work

Building Expert Systems
Logic Programming
Knowledge-Based Systems in AI
Winston and Horn LISP
AI Applications for Business
Winston's AI
Handbook of AI

FIG. 4. Repertory grid data structure for three multi-point constructs applied to AI books.

Note how 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. It may be desirable to apply more discrimination to this central rating and allow the subject the choice of the two possibilities: neither, that the element belongs to neither pole; or both, that the element belongs to both poles (Landfield, 1976). It is also possible to extend these possibilities to allow separate ratings on each pole (Shaw & Gaines, 1980; Gaines & Shaw, 1981*b*).

The mapping of the elements onto the constructs produces the two-dimensional grid of relationships which can be represented as a numeric data structure as shown in Fig. 4. This structure may be viewed as a component of a database in entity-antribute form (Chen, 1980): a repertory grid has elements as entities, constructs as attributes and allocations of elements to poles of constructs as values.

Knowledge support system design considerations

We see knowledge engineering in very broad terms as: the acquisition, elicitation, structuring and encoding of knowledge for application in inferential, goal-directed, explanatory, decision and action support systems. We see knowledge support systems as having even broader scope, encompassing both aids to knowledge engineering and support of human knowledge processes—in the long term the division between knowledge engineering tools and expert system shells will break down, and integrated systems will be necessary. The general requirements for a KSS are:

1. The KSS tools should be domain independent;
2. The KSS tools should be directly applicable by experts without intermediaries;
3. The KSS tools should be able to access a diversity of knowledge sources including text, interviews with experts, and observations of expert behavior;
4. The KSS system should be able to encompass a diversity of perspectives including partial or contradictory input from different experts;
5. The KSS system should be able to encompass a diversity of forms of knowledge and relationships between knowledge;

6. The KSS system should be able to present knowledge from a diversity of sources with clarity as to its derivation, consequences and structural relations;
7. Users of the KSS should be able to apply the knowledge in a variety of familiar domains and freely experiment with its implications;
8. The KSS should make provision for validation studies;
9. As much of the operation of the KSS as possible should be founded on well-developed and explicit theories of knowledge acquisition, elicitation and representation;
10. As the overall KSS develops it should converge to an integrated system.

All of these requirements are subject to caveats—some domain dependency may be appropriate for efficiency in specific KSSs—some human intervention may be helpful or necessary when an expert is using a KSS—and so on. However, the broad design goals stated capture the key issues in KSS design currently.

The PLANET system for repertory grid elicitation and analysis (Shaw, 1980, 1982; Shaw & Gaines, 1986*b*,*c*) is a primitive KSS satisfying requirements 1 and 2 for domain independence and direct use. Its foundations in personal construct psychology, which itself has strong systemic and cognitive science foundations (Gaines & Shaw, 1981*a*; Shaw & Gaines 1986*a*), are attractive in terms of requirement 9. Boose (1985) in evaluating ETS has noted the limitations of basic repertory grid techniques in terms of requirement 5—that the methodology is better suited for analysis than for synthesis problems, for example, debugging, diagnosis, interpretation and classification rather than design and planning, and that it is difficult to apply to deep causal knowledge or strategic knowledge—and is attempting to overcome these use grid hierarchies in NeoETS (Bradshaw & Boose, 1986). The TEIRESIAS extension to MYCIN is an early form of KSS providing debugging support for an expert system using basic analogical reasoning (Davis & Lenat, 1982). The development of KSSs has become a major area of activity recently, for example, MORE (Kahn, Nowlan & McDermott, 1985), MDIS (Antonelli, 1983), DSPL (Brown, 1984), MOLE (Eshelman *et al.* 1987), SALT (Marcus, McDermott & Wang, 1985; Marcus & McDermott, 1987, Marcus, 1987), SEAR (van de Brug, Bachant & McDermott 1985), and TKAW (Khan *et al.*, 1987).

The following section describes our work on KITTEN, a knowledge support system that draws on many concepts and techniques for knowledge engineering to begin to encompass requirements 3 through 8, while attempting to satisfy 9 by relating them all through personal construct psychology, and 10 by building a workbench of tools around a common database.

KITTEN: a knowledge support system

Figure 5 shows the structure of KITTEN: Knowledge Initiation & Transfer Tools for Experts and Novices. KITTEN 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.

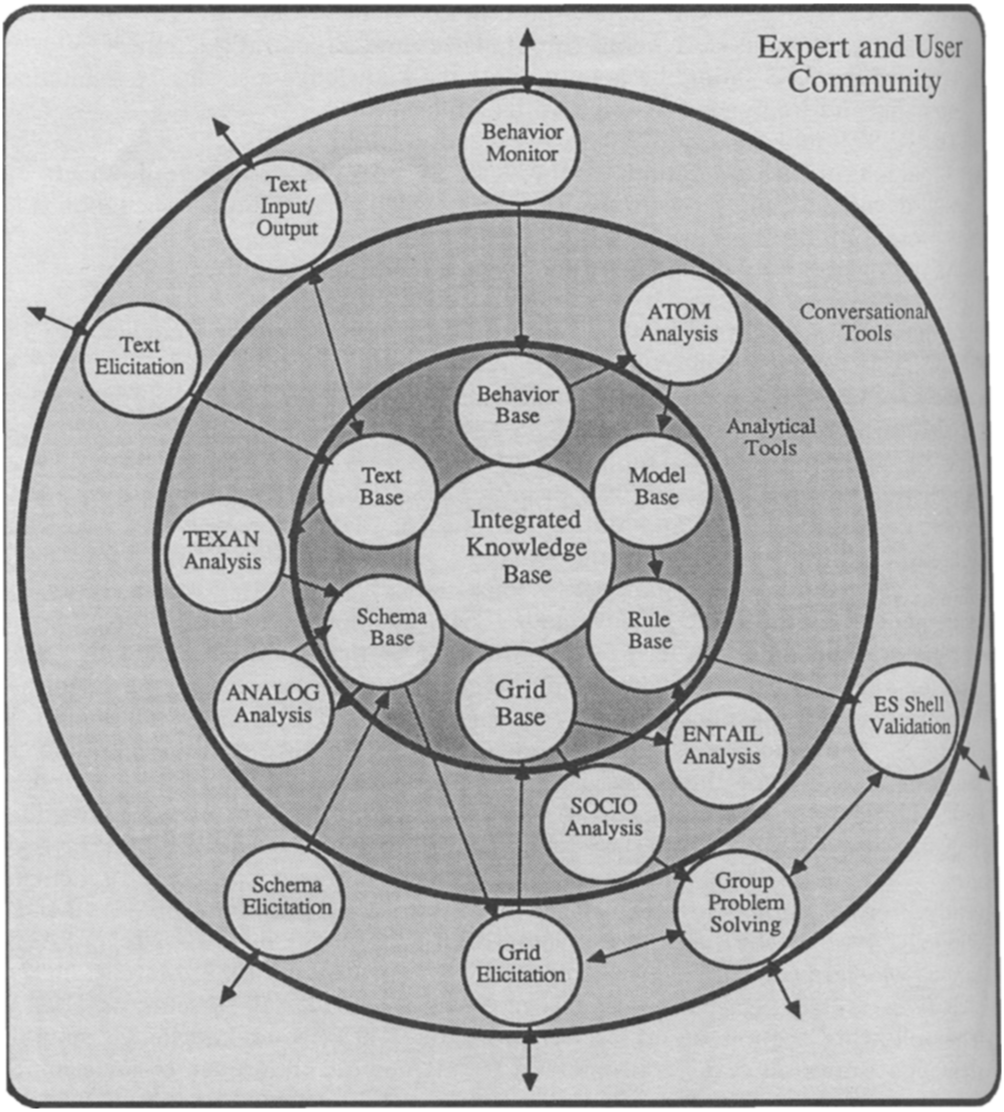

FIG. 5. KITTEN: Knowledge Initiation and Transfer Tools for Experts and Novices.

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*a*).

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, 1986).

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*b*). 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 6 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 Fig. 7.

As Bill is just starting his grid, he decides to elicit a construct from a triad, so he clicks **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 Fig. 8. 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 Fig. 9. Then Fig. 10 shows how he names the other end of the construct. That is, the way in which CY is different is that he is **dim**. 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 11 shows that he moves

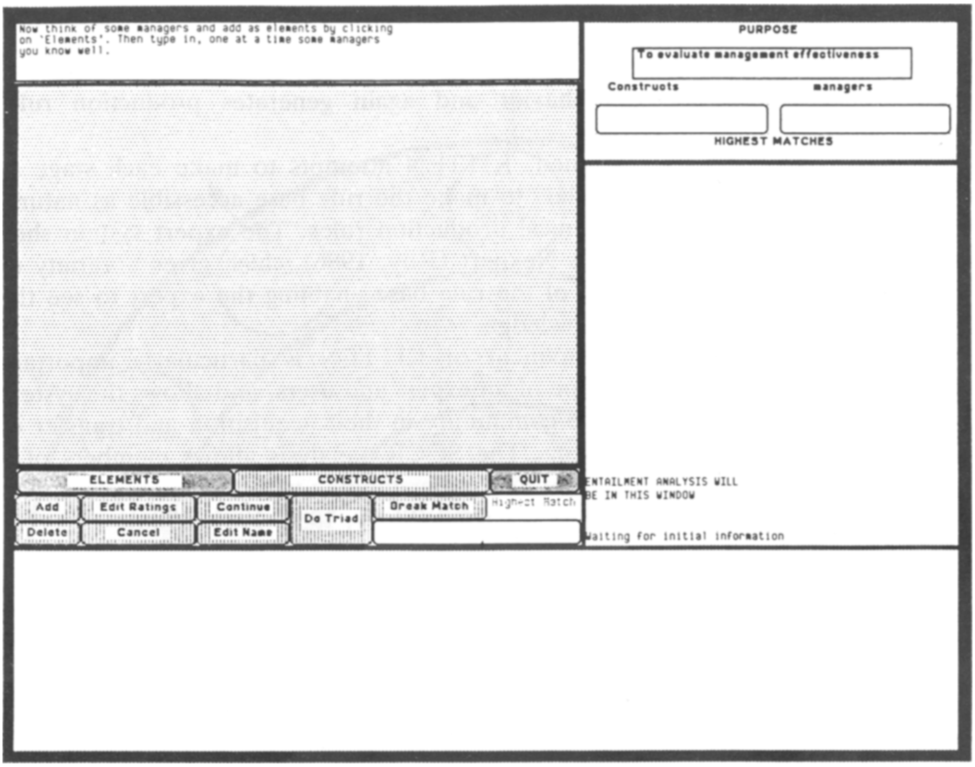

FIG. 6. KITTEN initial screen.

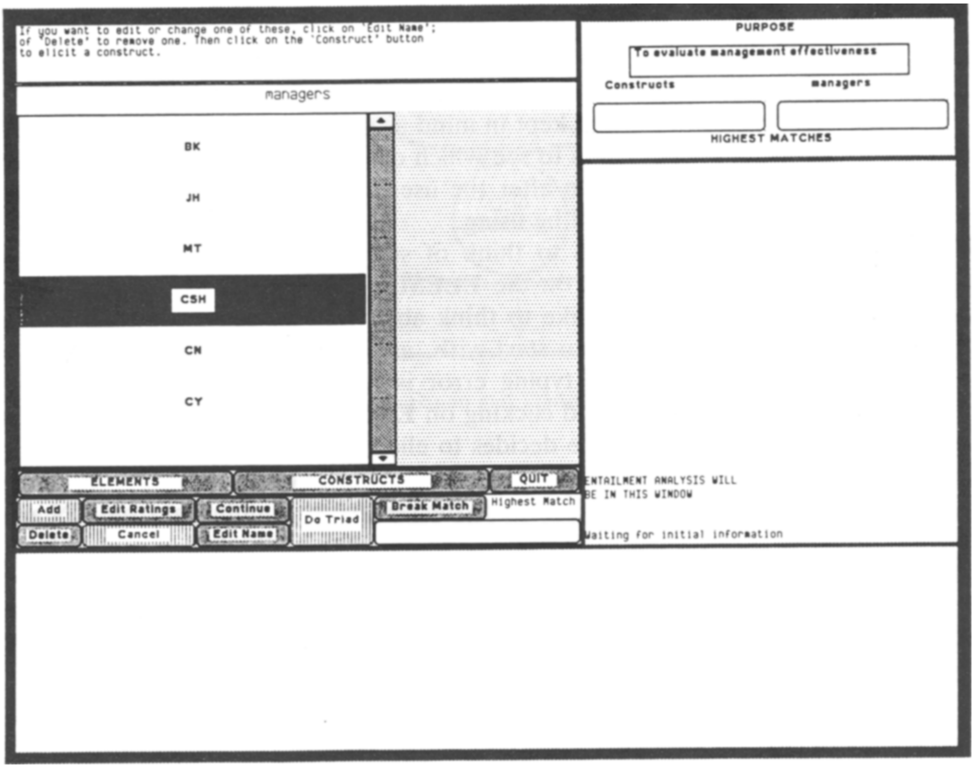

FIG. 7. KITTEN elements window.

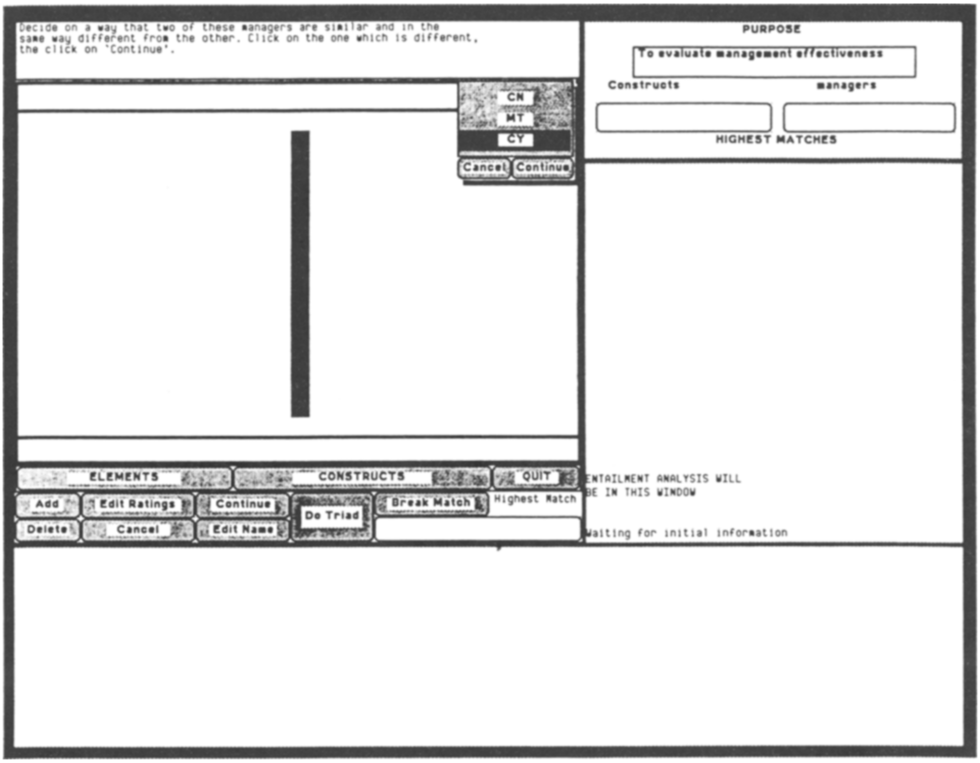

FIG. 8. KITTEN triad elicitation.

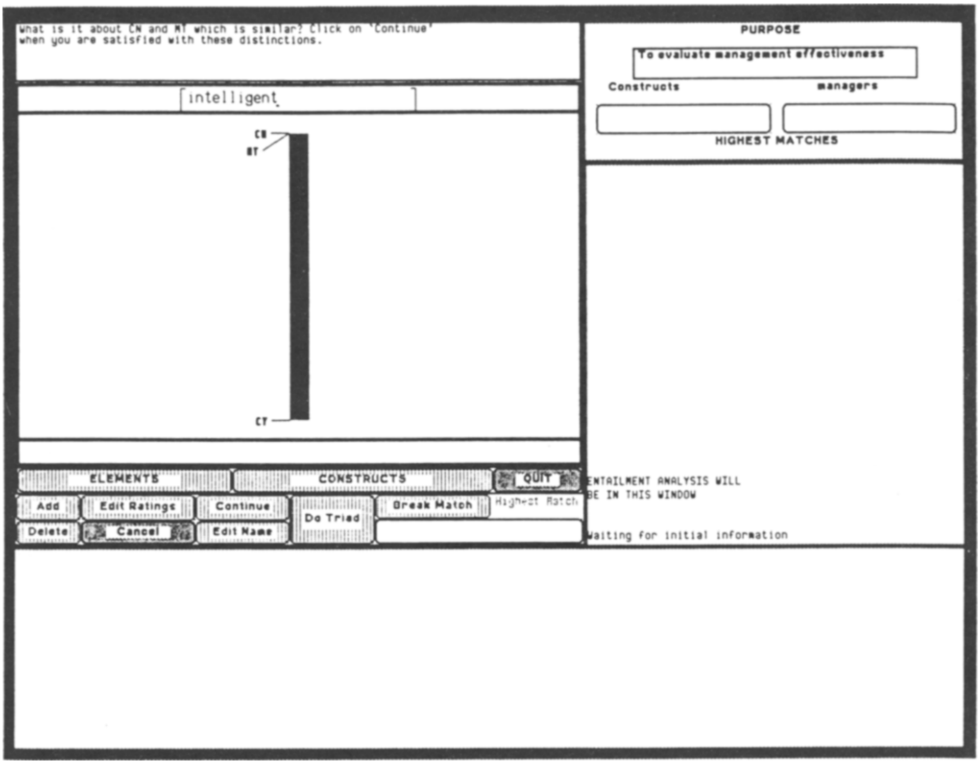

FIG. 9. KITTEN adding pole name for the similarity.

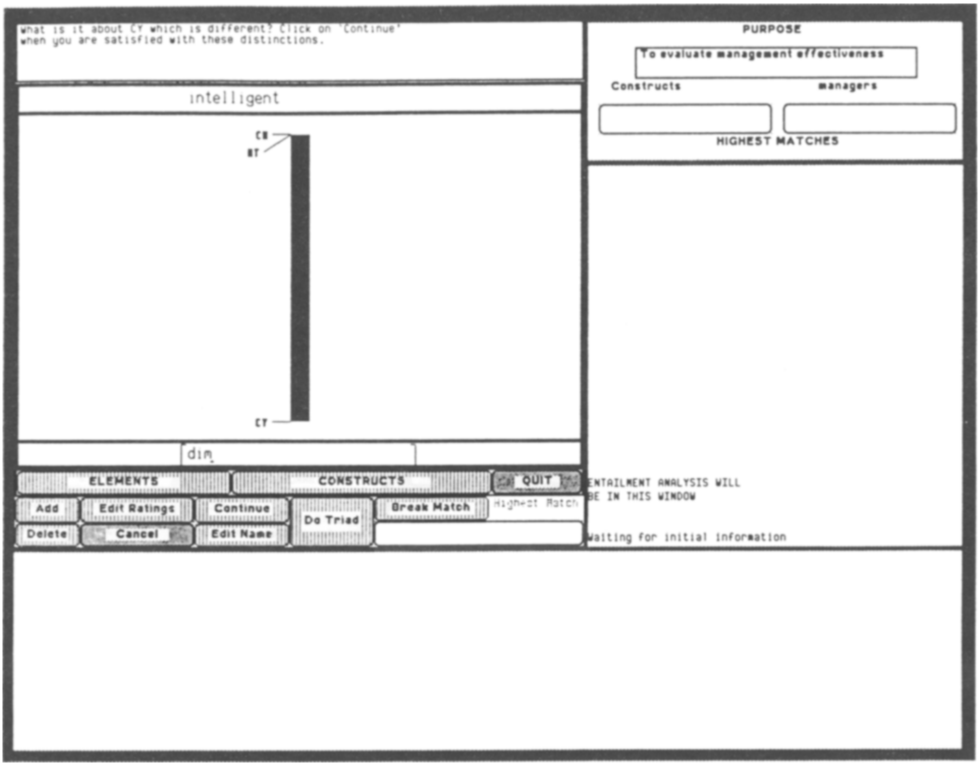

FIG. 10. KITTEN adding pole name for the difference.

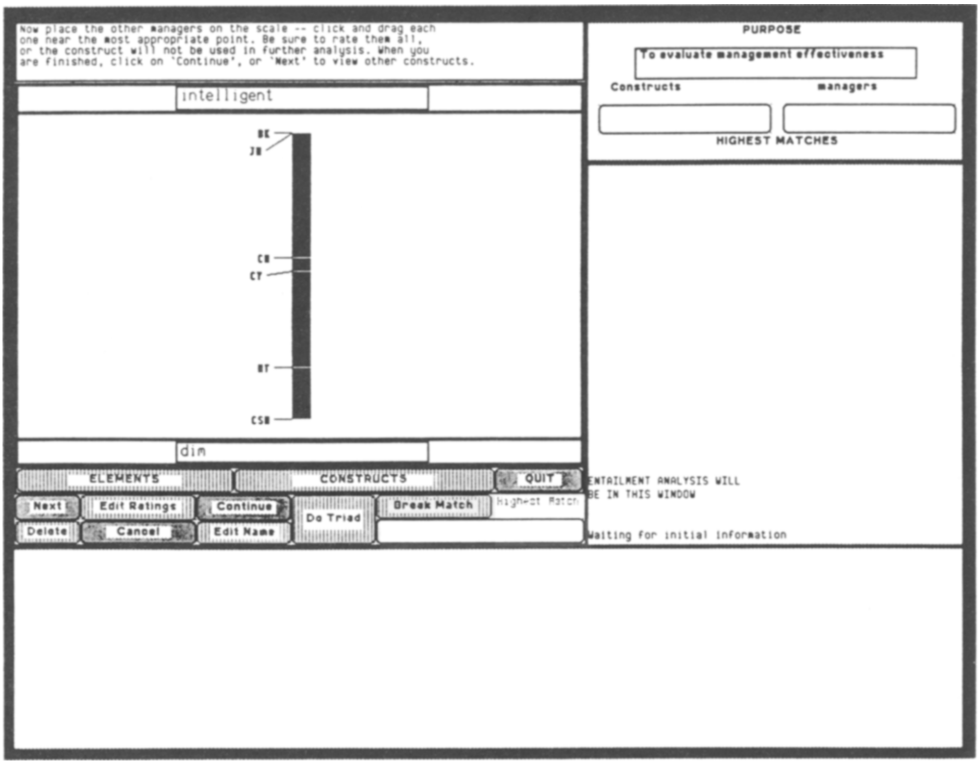

FIG. 11. KITTEN rating the elements.

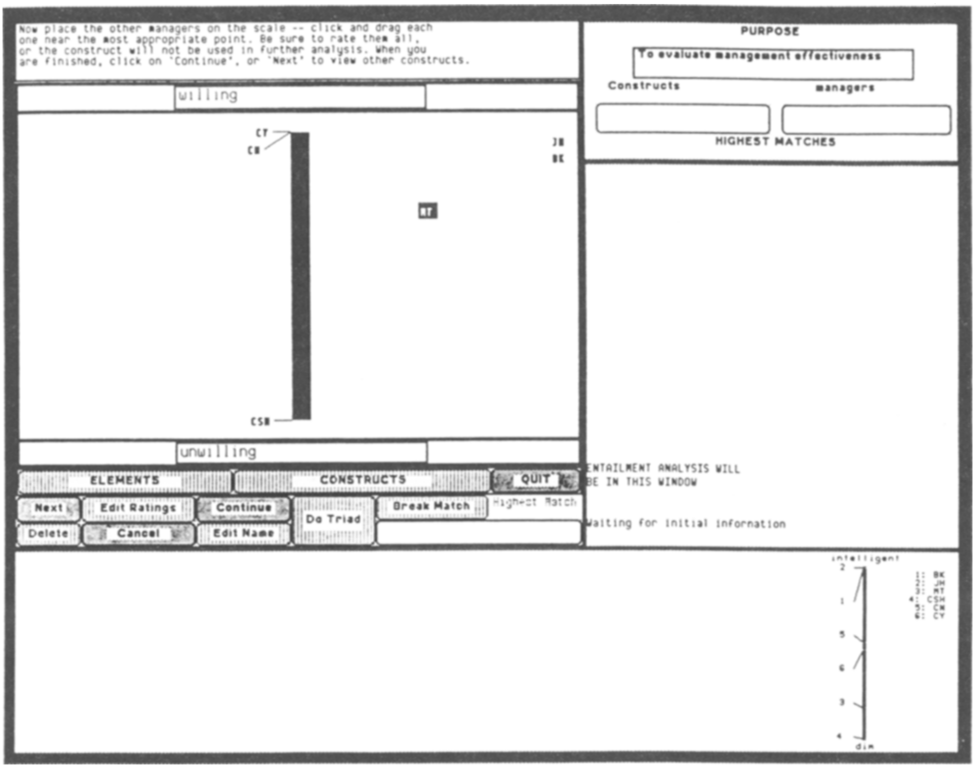

FIG. 12. KITTEN click and drag.

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.

Bill has now got his first construct. He continues adding several more constructs from triads before he decides to look at high matches. Figure 12 shows the first construct recorded in the bottom window, and the dragging of an element onto the construct bar for the second one. Figure 13 shows the completed construct.

After the next construct shown in Fig. 14, 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 (Fig. 15) showing which elements are matched, and placing one at either end of the construct bar. 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. The first thing he does after naming the poles, is to move one of the elements back close to the other, indicating that he cannot easily distinguish between them, as shown in Fig. 16. Figure 17 then shows the completed construct.

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

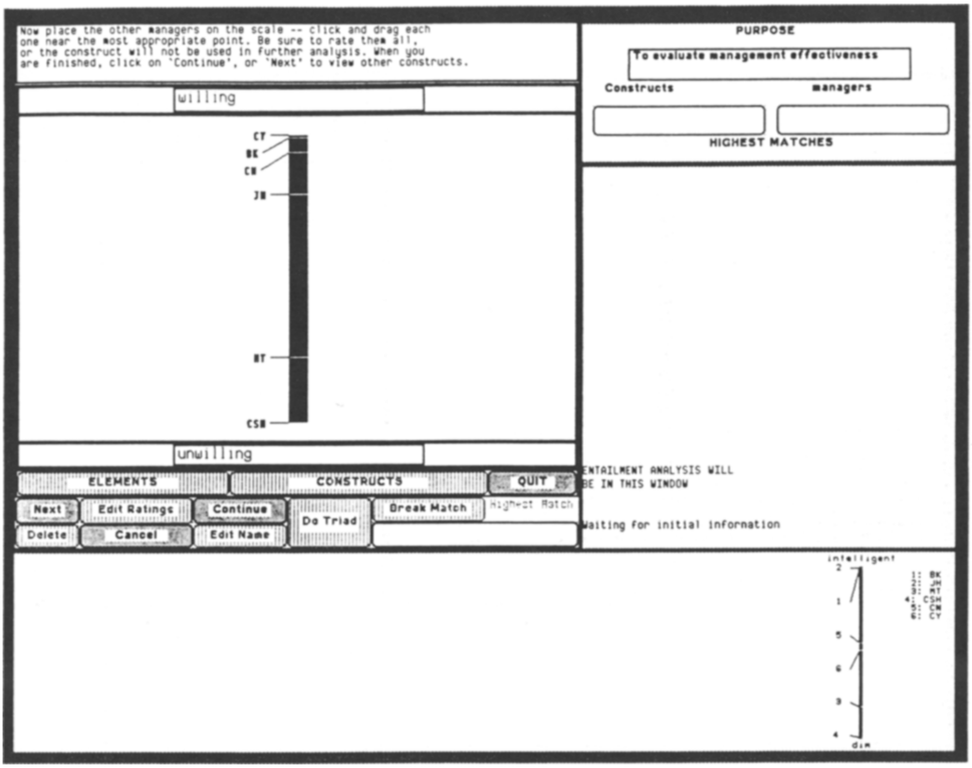

FIG. 13. KITTEN the second construct.

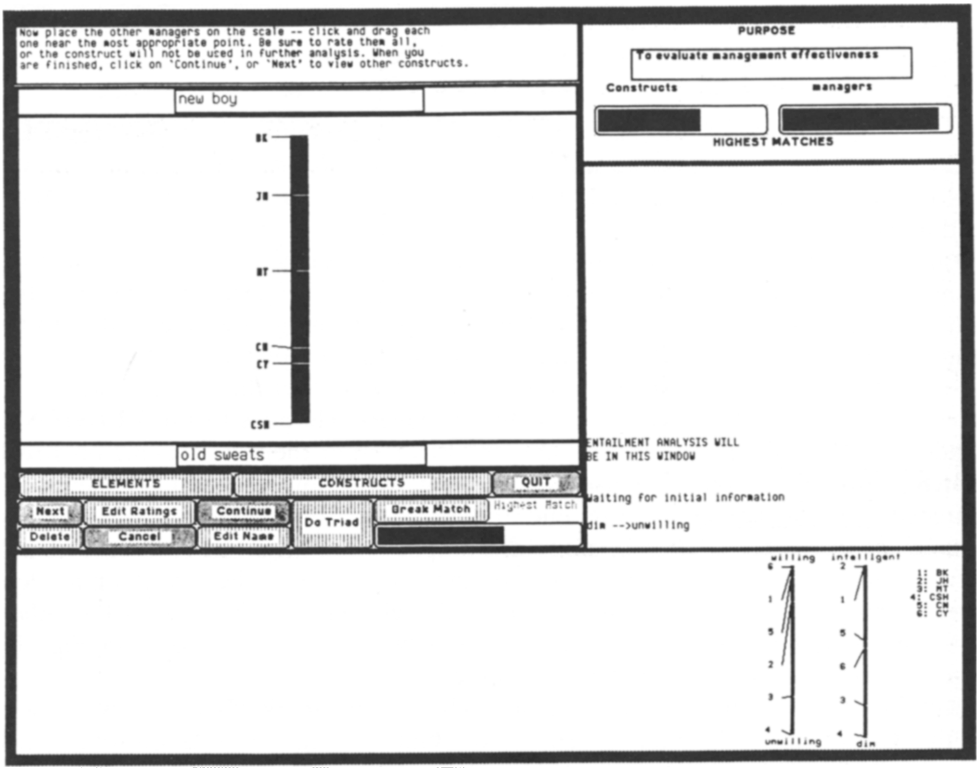

FIG. 14. KITTEN the third construct.

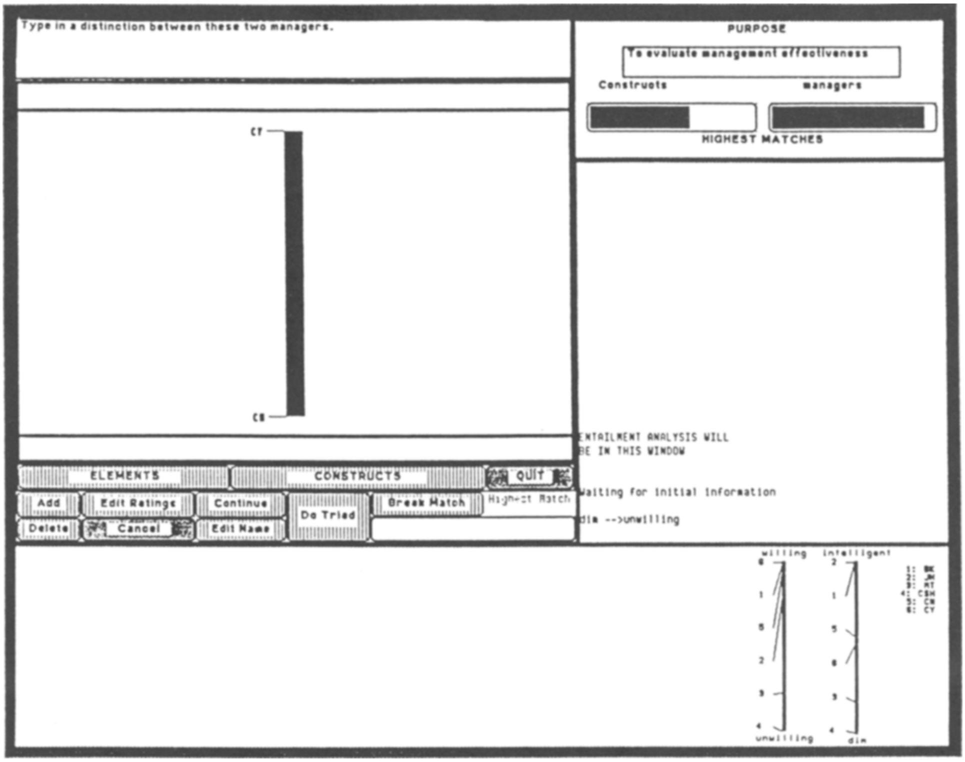

FIG. 15. KITTEN splitting matched elements.

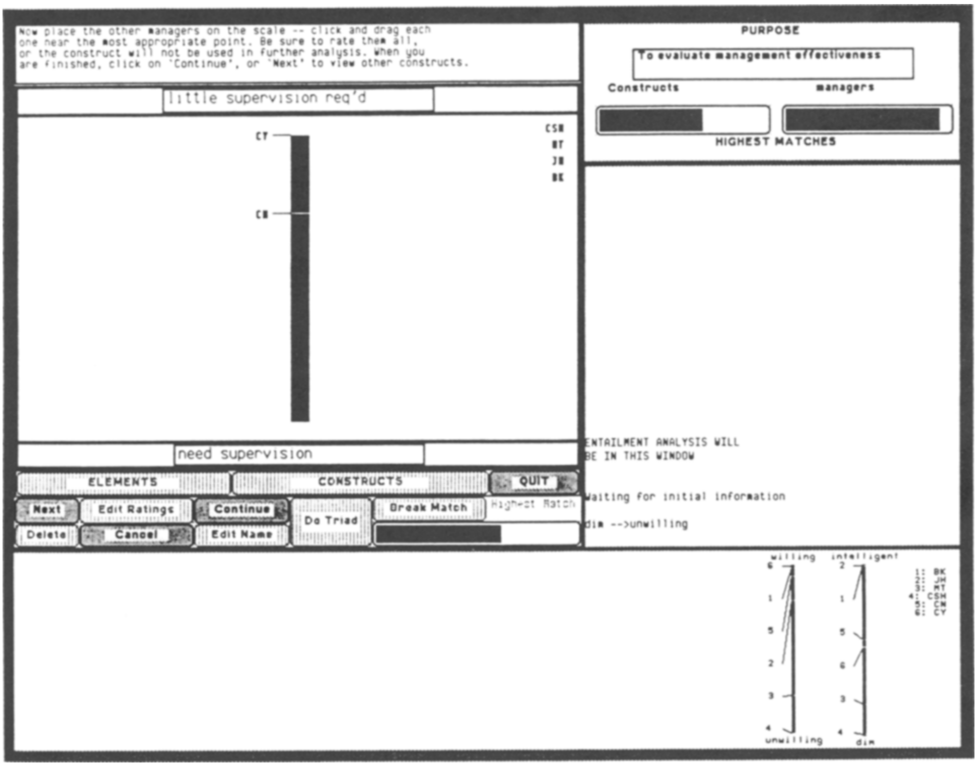

FIG. 16. KITTEN revising the ratings.

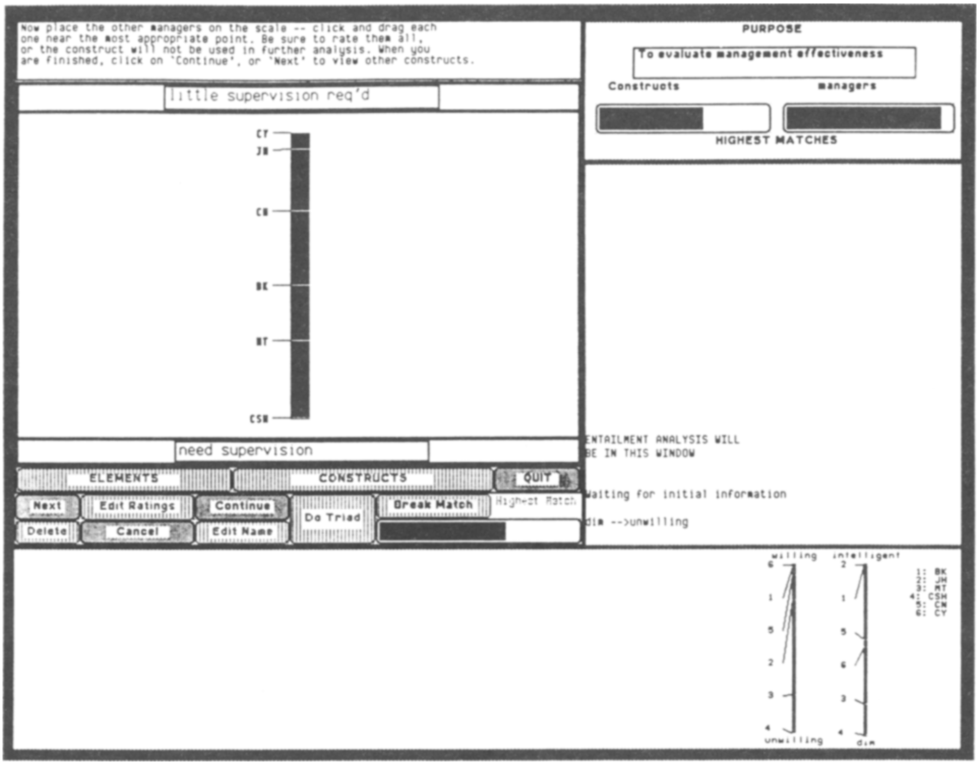

FIG. 17. KITTEN the fourth construct.

constructs are matched, shown in Fig. 18. This screen also shows the constructs elicited so far in the bottom window, and a new set of entailments in the right-hand window. Bill decides to add an element so he types in a new manager **NTM** (Fig. 19). 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 20 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 21 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. 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 Fig. 22. Figure 23 shows him looking at the positions of another manager CN. Any one of the elements can be highlighted in this way. Figures 24

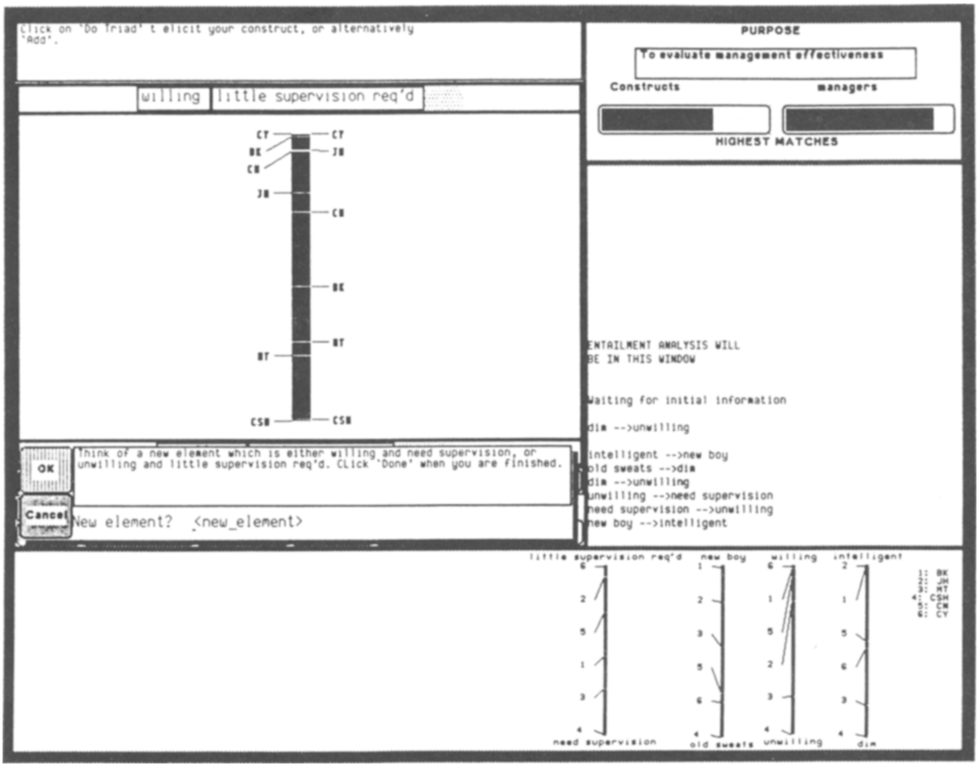

FIG. 18. KITTEN construct match.

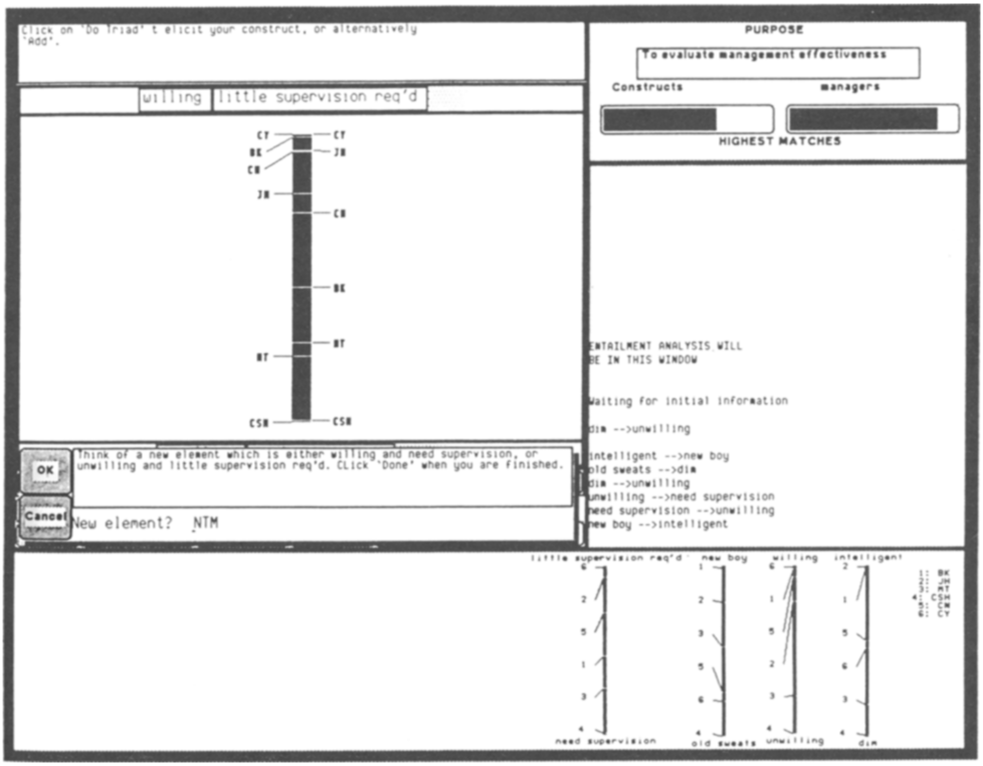

FIG. 19. KITTEN adding an element to break the construct match.

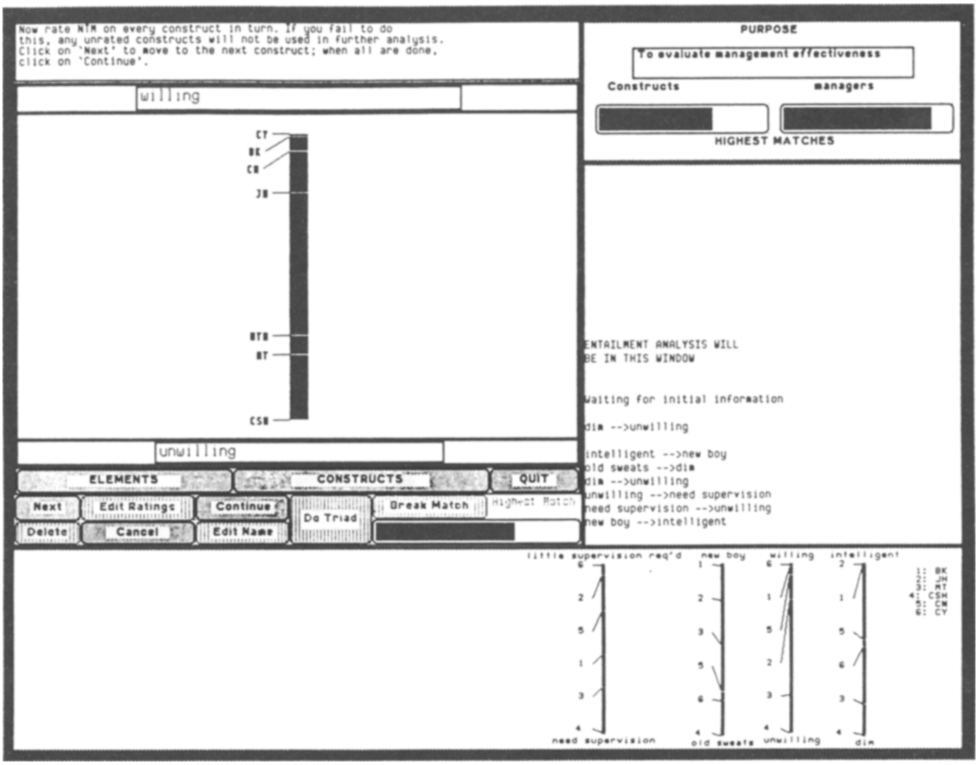

FIG. 20. KITTEN rating the new element.

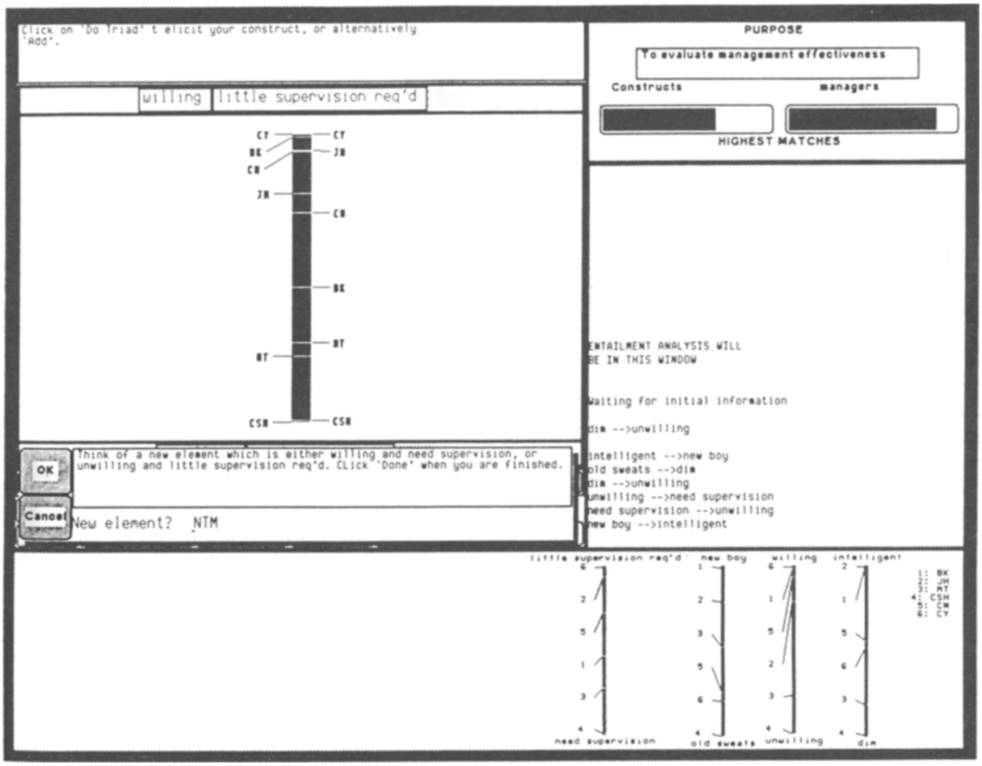

FIG. 21. KITTEN constructs.

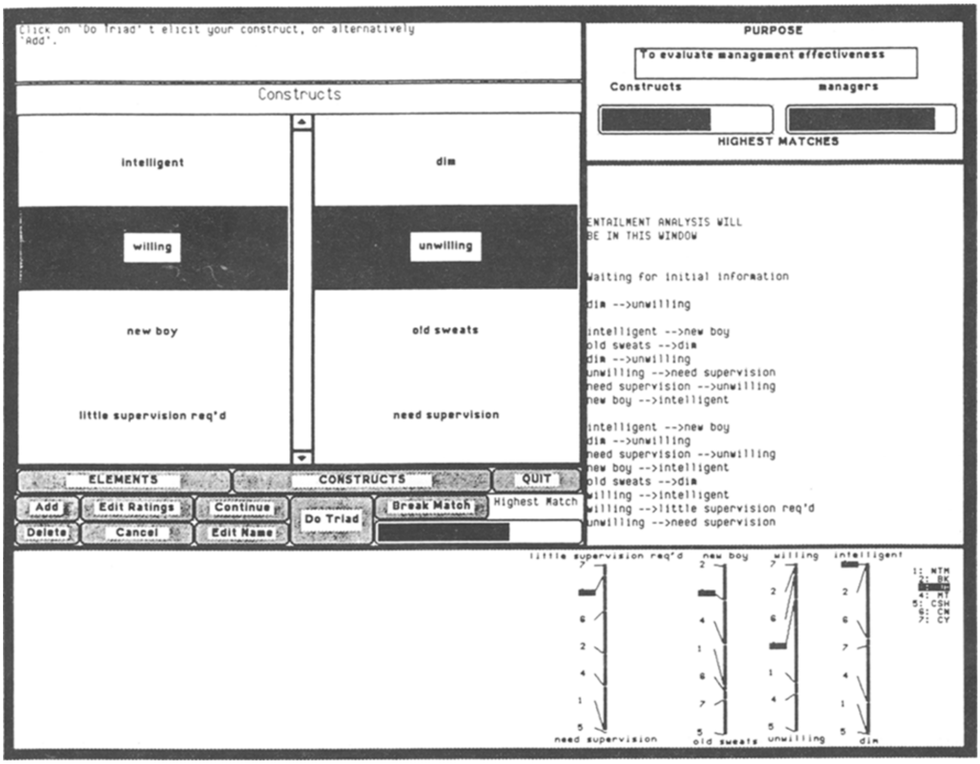

FIG. 22. KITTEN highlighting an element.

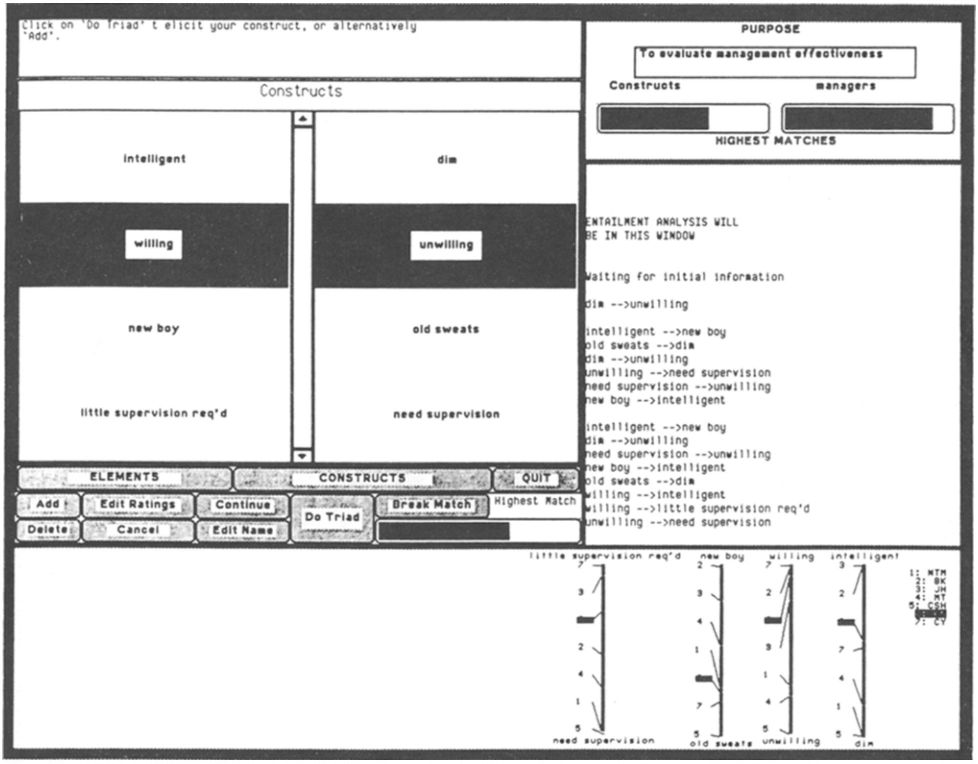

FIG. 23. KITTEN highlighting another element.

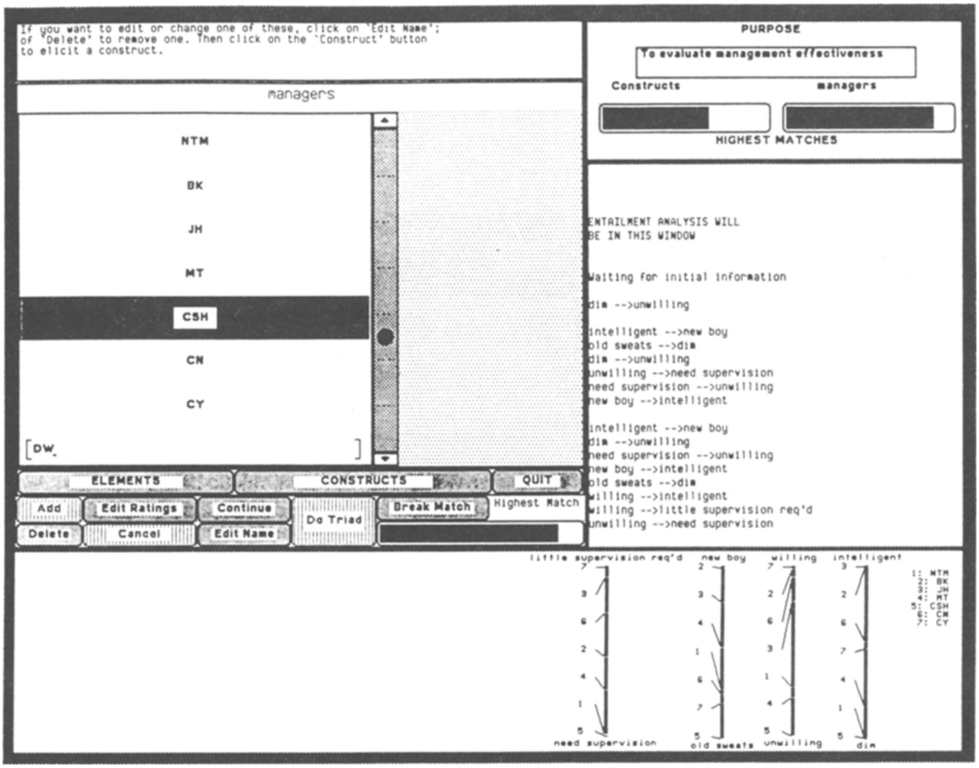

FIG. 24. KITTEN adding a new element.

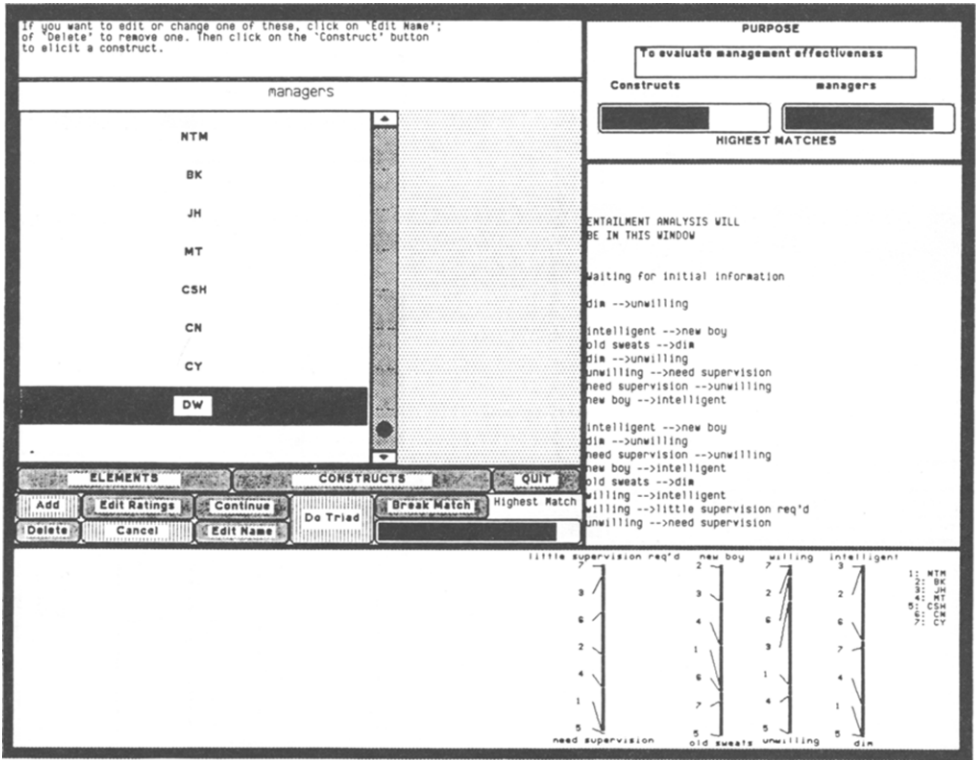

FIG. 25. KITTEN the list of elements.

and 25 show how Bill can add a new manager anytime he chooses by clicking first on the ELEMENTS button to bring up the screen in Fig. 24, and then just typing in the name of the new element. In Fig. 25 the manager is listed with all the others.

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 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.

ENTAIL: ENTAILMENT ANALYSIS

PLANET and KITTEN access the expert's personal construct system by interactively eliciting a repertory grid of constructs classifying elements characterizing to part of the domain of expertise. Figure 26 shows the resultant grid elicited in a study of personnel selection.

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. The original ENTAIL program produced all entailments consistent with the grid and allowed the expert to prune any that seemed spurious before using them as inference rules in an expert system. ENTAIL II 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*a*).

		1	2	3	4	5	6	7	8	9	10		
intelligent	1	1	1	4	5	3	3	5	2	3	5	1	dim
willing	2	1	2	4	5	1	1	4	3	1	2	2	unwilling
new boy	3	1	2	3	5	4	4	4	1	4	3	3	old sweats
little supervision reqd	4	3	1	4	5	2	1	5	2	2	3	4	need supervision
motivated	5	1	1	4	5	2	2	5	3	3	2	5	less motivated
reliable	6	3	2	2	5	1	1	5	1	2	3	6	not so reliable
mild	7	3	4	5	2	2	3	1	5	4	5	7	abrasive
ideas men	8	1	1	5	4	2	3	1	3	4	4	8	staid
self starters	9	2	1	5	5	1	3	5	3	4	5	9	need a push
creative	10	1	1	5	5	2	3	4	3	4	5	10	non-creative
helpful	11	4	3	4	2	3	5	1	4	5	5	11	unhelpful
professional	12	1	2	3	3	2	1	5	2	4	4	12	less professional
overall rating high	13	2	1	3	4	1	2	5	2	3	4	13	overall rating low
messers	14	2	2	5	4	3	5	1	5	3	1	14	tidy
		S1	S2	S3	S4	S5	S6	S7	S8	S9	S10		

FIG. 26. Repertory grid elicited on staff appraisal.

Figure 27 is an ENTAIL II analysis of the grid of Fig. 26. The entailments are shown with three values in the range from 0 to 1: first, the truth value of the hypothesis; second, the probability of the hypothesis being true; and third, the information content (uncertainty reduction generated) of asserting the hypothesis. For example, L1→L9 has a truth value of 0·80, a probability of 1·00, and an information content of 0·29. The information content measures the significance of the hypothesis and is used to ensure that trivial entailments consistent with the data are pruned.

The data of Fig. 26 may be regarded as that of an expert on staff appraisal concerned with deriving his overall rating (construct 13) from behavioral assessments such as intelligent and creative. The ENTAIL analysis of Fig. 27 shows that L1, L4, L6, L9, L10 and L12 imply L13, that intelligent, creative, reliable and professional self-starters requiring little supervision receive a high overall rating, whereas R2, R4, R5, R6, R9 and R12 imply R13, that being unwilling, less motivated, not so reliable, less professional, needing supervision and needing a push leads to a low overall rating.

Figure 28 shows Nexpert in operation loaded with the entailments of Fig. 27. 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,

Entail	Truth	Prob.	Inf.	(Cutoff 0·17) Implication usually
L1→L9	0·80	1·00	0·29	intelligent→self starters
L9→L13	1·00	1·00	0·29	self starters→overall rating high
R9→R1	0·80	1·00	0·28	need a push→dim
L10→L8	1·00	1·00	0·28	creative→ideas men
L1→L10	0·80	1·00	0·26	intelligent→creative
R8→R10	1·00	1·00	0·26	staid→non-creative
L10→L9	0·80	1·00	0·26	creative→self starters
R13→R6	0·80	1·00	0·26	overall rating low→not so reliable
L9→L10	0·80	1·00	0·24	self starters→creative
R10→R1	0·80	1·00	0·24	non-creative→dim
L10→L1	0.80	1·00	0·23	creative→intelligent
R13→R9	1·00	1·00	0·23	overall rating low→need a push
R4→R13	0·80	1·00	0·22	need supervision→overall rating low
R5→R4	0·80	1·00	0·22	less motivated→need supervision
R5→R13	0·80	1·00	0·22	less motivated→overall rating low
R9→R10	0·80	1·00	0·22	need a push→non-creative
L1→L3	0·80	1·00	0·21	intelligent→new boy
L6→L13	0·80	1·00	0·21	reliable→overall rating high
R10→R9	0·80	1·00	0·20	non-creative→need a push
R1→R6	0·60	1·00	0·19	dim→not so reliable
R1→R10	0·80	1·00	0·19	dim→non-creative
R9→R4	0·60	1·00	0·19	need a push→need supervision
R9→R12	0·60	1·00	0·19	need a push→less professional
R9→R13	0·60	1·00	0·19	need a push→overall rating low
R12→R13	0·80	1·00	0·19	less professional→overall rating low
R13→R4	0·80	1·00	0·19	overall rating low→need supervision
R13→R12	0·80	1·00	0·19	overall raing low→less professional
L4→L5	0·80	1·00	0·18	little supervision reqd→motivated
L4→L9	0·60	1·00	0·18	little supervision reqd→self starters
R6→R4	0·80	1·00	0·18	not so reliable→need supervision
R6→R13	0·80	1·00	0·18	not so reliable→overall rating low
L12→L9	0·60	1·00	0·18	professional→self starters
L13→L4	0·80	1·00	0·18	overall rating high→little supervision reqd
L13→L5	0·80	1·00	0·18	overall rating high→motivated
L13→L9	0·60	1·00	0·18	overall rating high→self starters
L1→L8	0·80	1·00	0·17	intelligent→ideas men
R2→R4	0·80	1·00	0·17	unwilling→need supervision
R2→R5	0·80	1·00	0·17	unwilling→less motivated
R2→R13	0·80	1·00	0·17	unwilling→overall rating low
R3→R1	0·80	1·00	0·17	old sweats→dim
L4→L13	0·80	1·00	0·17	little supervision reqd→overall rating high
L12→L13	0·80	1·00	0·17	professional→overall rating high
L13→L12	0·80	1·00	0·17	overall rating high→professional

FIG. 27. ENTAIL analysis of repertory grid on staff appraisal.

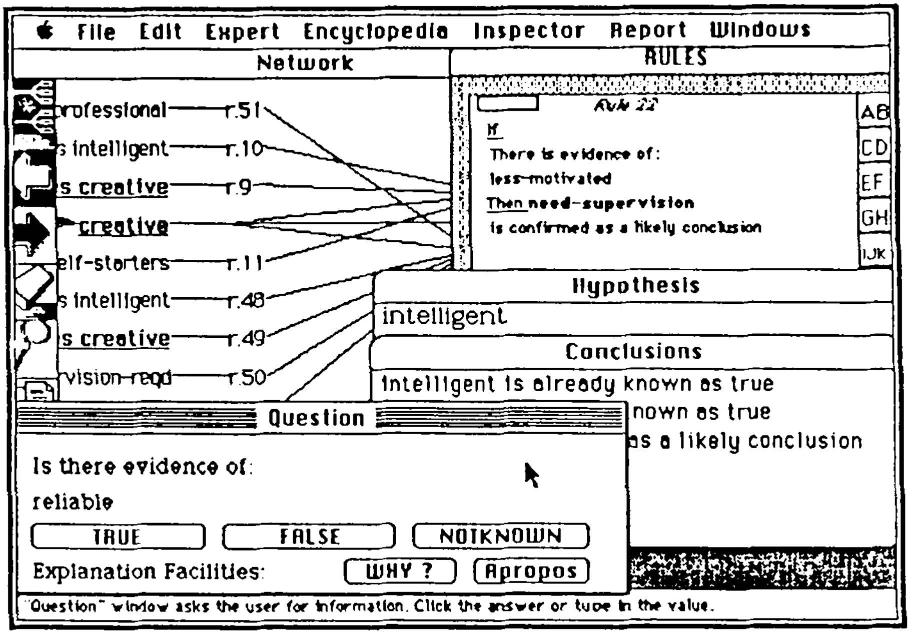

FIG. 28. Inference rules derived by ENTAIL in use in Nexpert expert system shell.

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.

TEXAN: TEXT ANALYSIS

Repertory grid techniques depend on eliciting elements and constructs from experts that are representative of a domain and comprehensive in their classification. The interactive elicitation program PEGASUS in PLANET uses online analysis of the grid to feed back comments to the expert which stimulate the addition of elements and constructs to achieve comprehensiveness (Shaw, 1980). However, this structural feedback is only applicable when a grid has been partially completed and the initial selection of elements has had no computer-based support.

TEXAN is a text analysis program designed to pump-prime the grid elicitation process when a manual or text book is available that the expert regards as having reasonable coverage of the domain. It uses techniques that were originally designed to map subject matter concepts against student concepts in computer-managed instruction systems (Smith, 1976). The text is fully indexed by all non-noise words grouped by their stems, and a coupling matrix of word associations is calculated using a simple distance-in-text measure. The high-frequency associations in the text are clustered and presented to the expert as a prototypical schema for the subject area which he can edit for spurious words and associations, and then use to suggest knowledge islands and associated elements and constructs.

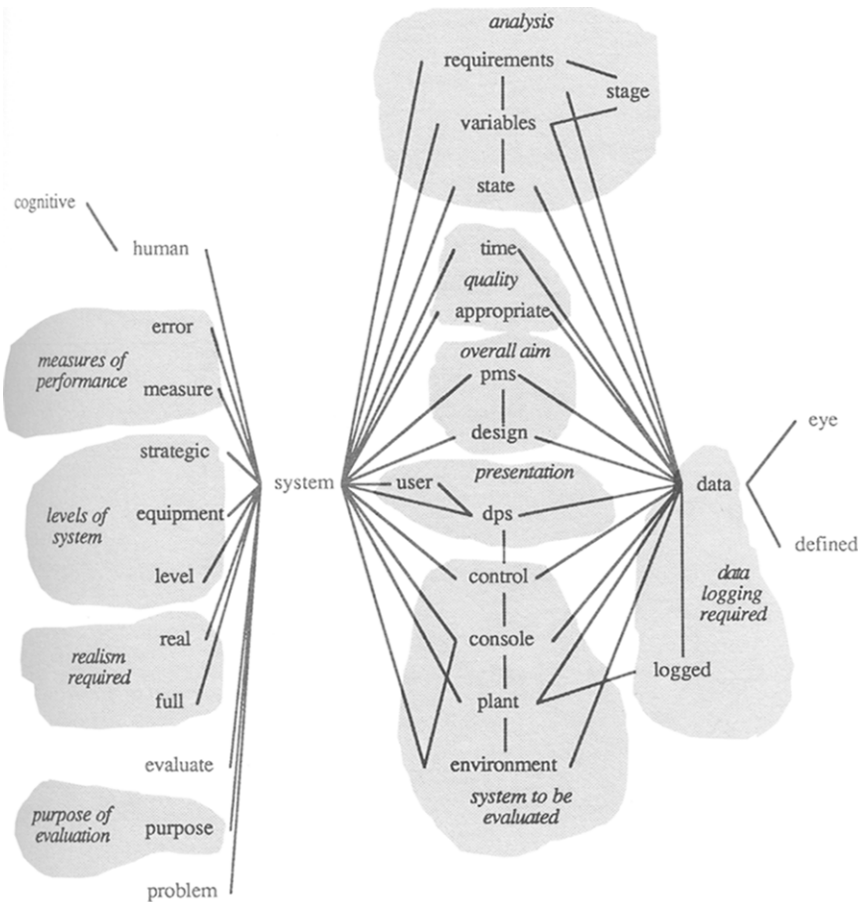

FIG. 29. TEXAN clustering of word associations from text with annotation showing knowledge islands.

Figure 29 shows a TEXAN clustering of an evaluation study of data logging, analysis and presentation methodologies for human performance evaluation in complex systems (Gaines & Moray, 1985). Figure 30 shows an independent mapping of the main knowledge islands for an expert system design based on the analyzed report (Gaines, 1986). The TEXAN analysis was done some time after the production of Fig. 30, and the shading of Fig. 29 shows the relationship of some of the groupings in the schema with the knowledge islands. There is not a one-to-one correspondence but this, and similar analyses, show that basic text analysis can focus attention on salient features of the domain and pump-prime the knowledge elicitation process.

In the long term more sophisticated text analysis techniques may be used to derive knowledge from text without human intervention. However, for many domains the knowledge is not yet that explicit and pump-priming of elicitation from experts will remain a significant requirement.

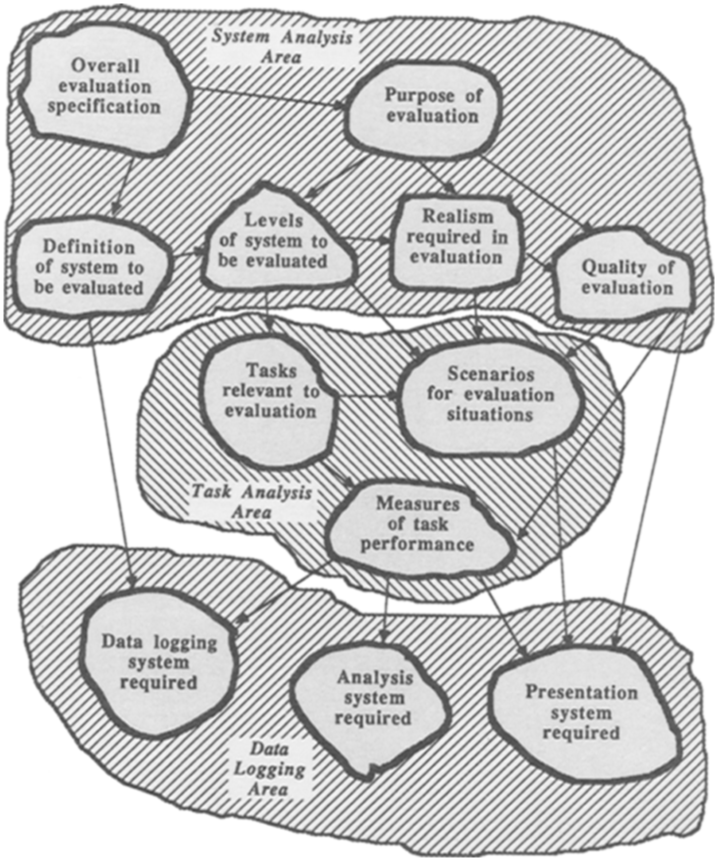

FIG. 30. Knowledge islands specified by the expert for a performance measuring system.

ANALOG: SCHEMA ANALYSIS

The groupings of Figs 29 and 30 when combined with the construct classifications of repertory grids as in Fig. 26 may be viewed as schema structuring a knowledge domain. ANALOG is a program that maps schema to schema based on their structure without regard to content. It is based on a theory of analogy that explicates analogies as pullbacks of faithful functors between categories (Gaines & Shaw, 1982) and generates maximal sub-graph isomorphisms between two classificatory data structures. It may be regarded as a generalization of the copy-edit process being used in the encoding of commonsense knowledge in CYC (Lenat, Prakash & Shepherd, 1986). ANALOG produces meaningful results on artificial examples and grids in related domains. It will also find meaningless analogies between unrelated domains which cannot be rejected by information-theoretic statistical procedures such as those used in ENTAIL and ATOM. It seems likely that effective application of ANALOG depends on the expert pump-priming the matching with known or hypothesized relations and the program extending these rather than attempting to generate them completely.

ATOM: BEHAVIOR ANALYSIS

Michalski and Chilausky (1980) have demonstrated that inductive modeling of an expert's behavior may produce effective rules when those elicited by interview techniques are clearly inadequate. ATOM is an algorithm for inducing the structure of a system from its behavior using a search over a model space ordered by complexity and goodness of fit. As in ENTAIL, models are evaluated in terms of the uncertainty reduction induced by the model in the distribution of the modeled behavior (Gaines, 1976; 1977; 1979). We have incorporated a version of ATOM in KITTEN that takes a set of sequences of arbitrary symbolic data and generates a set of production rules that will reconstruct it. These can be loaded into the expert system shell to give a simulator of the behavioral system. This has proved effective with inter-personal interaction data such as that analyzed by Mulhall (1977) and interactively elicited by Stevens (1985).

SOCIO: MUTUAL AGREEMENT AND UNDERSTANDING ANALYSIS

We have already emphasized the need for knowledge elicitation methodologies to cope with a group of experts as well as the individual. 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.

Conclusions: steps toward an integrated knowledge support system

This paper has presented the concept of knowledge support systems as interactive knowledge engineering tools, stated the design criteria for such systems, and outlined the structure and key components of KITTEN. KITTEN consists of a set of knowledge engineering tools, some of which already have track records of successful use in knowledge acquisition studies. In developing KITTEN we have preserved the integrity of each of these tools, enabling each to be utilized effectively in a stand-alone mode. However, we have also made the first steps towards an integrated knowledge support system by building the tools around a common database, providing access to the same data in each of its intermediate forms, and providing conversion utilities between different data forms.

The objective of integrating the tools has raised a number of new and significant questions. ENTAIL transforms a repertory grid to a set of production rules—is it possible, and useful, to convert production rules to a repertory grid? Technically the result is a possible world of grids that might have generated the rules, and the capability does prove useful, particularly given the other grid analysis tools available in KITTEN. Similar considerations apply to the transformations between other forms of knowledge representation. We see the next generation of knowledge support tools as being increasingly flexible in handling all aspects of knowledge acquisition, representation, processing and presentation. They will not be optimized with a particular knowledge representation, uncertainty calculation, inference mechanism, and so on, that are in some sense right. Rather they will provide a wide range of perspectives on the knowledge base, preserving source data and chains of derivative processes, so that users can freely explore the knowledge or follow a very specific path according to their choices and needs.

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. We are grateful to John Boose and Jeff Bradshaw of Boeing AI Center, for stimulating discussions relating to knowledge support systems.

References

ANTONELLI, D. (1983). The application of artificial intelligence to a maintenance and diagnostic information system (MDIS). *Proceedings of the Joint Services Workshop on Artificial Intelligence in Maintenance*, Boulder, CO.

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

BOOSE, J. H. (1985). A knowledge acquisition program for expert systems based on personal construct psychology. *International Journal of Man–Machine Studies*, **20**(1), 21–43 (January).

BOOSE, J. H. (1986). Rapid acquisition and combination of knowledge from multiple experts in the same domain. *Future Computing Systems*, **1**(2), 191–216.

BRADSHAW, J. M. & BOOSE, J. H. (1986). NeoETS. *Proceedings of North American Personal Construct Network Second Biennial Conference*. University of Calgary: Department of Computer Science (June), pp. 27–41.

BROWN, D. E. (1984). Expert systems for design problem-solving using design refinement with plan selection and redesign. Unpublished Ph.D. dissertation, Ohio State University, CIS Department, Columbus, Ohio, August, 1984.

CHEN P. P. (ed.) (1980). *Entity-relationship Approach to Systems Analysis and Design*. North-Holland, New York.

DAVIS, R. & LENAT, D. B. (1982). *Knowledge-Based Systems in Artificial Intelligence*. New York: McGraw-Hill.

ESHELMAN, L, EHRET, D., MCDERMOTT, J. & TAN, M. (1987) MOLE: A tenacious knowledge acquisition tool. In: special issue on the AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop. *International Journal of Man–Machine Studies* **26**(1), 41–54.

ESHELMAN, L. & MCDERMOTT, J. (1986). MOLE: a knowledge acquisition tool that uses its head. *Technical Report*. Carnegie-Mellon University: Department of Computer Science.

GAINES, B. R. (1976). Behaviour/structure transformations under uncertainty. *International Journal of Man–Machine Studies*, **8**(3), 337–365.

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

GAINES, B. R. (1979). Sequential fuzzy system identification. *International Journal of Fuzzy Sets and Systems*, **2**(1), 15–24.

GAINES, B. R. (1986). Development of performance measures for computer-based man–machine interfaces: Application to previous SHINMACS evaluation. *Technical Report DCIEM-PER-SUP: MAT 86.*

GAINES, B. R. & MORAY, N. (1985). Development of performance measures for computer-based man–machine interfaces. *Technical Report DCIEM-PER-FIN:JUL 85.*

GAINES, B. R. & SHAW, M. L. G. (1980). New directions in the analysis and interactive elicitation of personal construct systems. *International Journal of Man–Machine Studies*, **13**(1) 81–116.

GAINES, B. R. & SHAW, M. L. G. (1981*a*). A programme for the development of a systems methodology of knowledge and action. In RECKMEYER, W. J., Ed. *General Systems Research and Design: Precursors and Futures.* Society for General Systems Research, pp. 255–264.

GAINES, B. R. & SHAW, M. L. G. (1981*b*). New directions in the analysis and interactive elicitation of personal construct systems. In SHAW, M. L. G., Ed. *Recent Advances in Personal Construct Technology.* Academic Press, London, pp. 147-182.

GAINES, B. R. & SHAW, M. L. G. (1982). Analysing analogy. TRAPPL, R., RICCIARDI, L. & PASK, G., Eds. *Progress in Cybernetics and Systems Reserach, Vol. IX.* Washington: Hemisphere, pp. 379–386.

GAINES, B. R. & SHAW, M. L. G. (1986*a*). Induction of inference rules for expert systems. *Fuzzy Sets and Systems*, **8**(3), 315–328.

GAINES, B. R. & SHAW, M. L. G. (1986*b*). Knowledge engineering for an FMS advisory system. In LENZ, J. E., Ed. *Proceedings of Second International Conference on Simulation in Manufacturing: AMS'86.* Bedford, UK: IFS Conferences, pp. 51–61.

GAMMACK, J. G. & YOUNG, R. M. (1985). Psychological techniques for eliciting expert knowledge. Bramer, M., Ed. *Research and Development in Expert Systems.* Cambridge University Press, pp. 105–116.

KAHN, G. S., BREAUX, E. H., JOSEPH, R. L., & DEKLERK, P. (1987). An intelligent mixed-initiative workbench for knowledge acquisition. In: special issue on the AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop. *International Journal of Man–Machine Studies*, **27** (in press).

KAHN, G., NOWLAN, S. & MCDERMOTT, J. (1985). MORE: an intelligent knowledge acquisition tool. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence.* California: Morgan Kaufmann, pp. 581–584.

LANDFIELD, A. (1976). A personal construct approach to suicidal behavior. In SLATER, P. Ed. *Dimensions of Intrapersonal Spaces* Vol. 1, John Wiley, London, pp. 93–107.

LENAT, D., PRAKASH, M. & SHEPHERD, M. (1986). CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Magazine* **6**(4), 65–85.

MARCUS, S. (1987). Taking backtracking with a grain of SALT. In: special issue on the AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop. *International Journal of Man–Machine Studies*, **26**(3), 383–398.

MARCUS, S., and MCDERMOTT, J. (1987) SALT: a knowledge acquisition tool for propose-and-revise systems technical report, forthcoming, Carnegie–Mellon University Department of Computer Science.

MARCUS, S., MCDERMOTT, J. & WANG, T. (1985). Knowledge acquisition for constructive systems. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence.* California: Morgan Kaufmann, pp. 637–639.

MICHALSKI, R. S. & CHILAUSKY, R. L. (1980). Knowledge acquisition by encoding expert rules versus computer induction from examples—a case study involving soyabean pathology. *International Journal of Man–Machine Studies*, **12**, 63–87.

MULHALL, D. J. (1977). The representation of personal relationships: an automated system. *International Journal of Man–Machine Studies*, **9**(3), 315–335.

POPE, M. L. & SHAW, M. L. G. (1981). Personal construct psychology in education and learning. In SHAW, M. L. G., Ed. *Recent Advances in Personal Construct Technology.* Academic Press, London, pp. 105–114.

ROY, J. (1986). Expert systems in Nexpert. *MacTutor*, **2**(2), 48–51.

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

SHAW, M. L. G. (1981). *Recent Advances in Personal Construct Technology.* 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. (1986). PCS: a knowledge-based interactive system for group problem solving. *Proceedings of* 1986 *International Conference on Systems, Man and Cybernetics.* pp. 1353–1357.

SHAW, M. L. G. & CHANG, E. (1986). A participant construct system. *Proceedings of North American Personal Construct Network Second Biennial Conference*. University of Calgary: Department of Computer Science, pp. 131–140.

SHAW, M. L. G. & GAINES, B. R. (1980). Fuzzy semantics for personal construing. *Systems Science and Science*. Society for General Systems Research, Kentucky, pp. 59–68.

SHAW, M. L. G. & GAINES, B. R. (1983). A computer aid to knowledge engineering. *Proceedings of British Computer Society Conference on Expert Systems*. Cambridge, pp. 263–271.

SHAW, M. L. G. & GAINES, B. R. (1986*a*). A framework for knowledge-based systems unifying expert systems and simulation. LUKER, P. A. & ADELSBERGER, H. H., Eds. *Intelligent Simulation Environments*. La Jolla, California: Society for Computer Simulation, pp. 38–43.

SHAW, M. L. G. & GAINES, B. R. (1986*b*). Interactive elicitation of knowledge from experts. *Future Computing Systems*, **1**(2), 151–190.

SHAW, M. L. G. & GAINES, B. R. (1986*c*). An interactive knowledge elicitation technique using personal construct technology. Kidd, A., Ed., *Knowledge Elicitation for Expert Systems: A Practical Handbook*. Plenum Press (to appear).

SHAW, M. L. G. & MCKNIGHT, C. (1981). *Think Again*. Englewood Cliffs, NJ: Prentice-Hall.

SHEPHERD, E. & WATSON, J. P. (Eds.) (1982). *Personal Meanings*, John Wiley, London.

SMITH, R. A. (1976). Computer-based structural analysis in the development and administration of educational materials. *International Journal of Man–Machine Studies*, **8**(4), 439–463.

STEVENS, R. F. (1985). An on-line version of the personal relations index psychological test. *International Journal of Man–Machine Studies*, **23**(5), 563–585.

VAN DE BRUG, A. BACHANT, J. & MCDERMOTT, J. (1985). Doing R1 with style. *Proceedings of the Second Conference on Artificial Intelligence Applications*. IEEE 85CH2215-2. Washington: IEEE Computer Society Press, pp. 244–249.

WAHL, D. (1986). An application of declarative modeling to aircraft fault isolation and diagnosis. LUKER, P. A. & ADELSBERGER, H. H., Eds. *Intelligent Simulation Environments*. La Jolla, California: Society for Computer Simulation, pp. 25–28.