KNOWLEDGE SUPPORT SYSTEMS

Brian R Gaines & Mildred L G Shaw

Knowledge Science Institute, University of Calgary
Calgary, Alberta, Canada T2N 1N4

ABSTRACT

This paper introduces the concept of knowledge support systems—integrated interactive knowledge acquisition systems and expert system shells—that can provide a specialist user community with tools supporting a wide range of knowledge processes. These system extrapolate the trend from human knowledge engineering, through automated interviewing of the expert, to continuing on-line access to both knowledge acquisition and application processes. In knowledge support systems the distinctions between expert, knowledge engineer and user roles are deliberately blurred, and a diversity of knowledge processes and changing roles are supported within an entire interacting community. A prototype knowledge support system is described with examples of some of the knowledge acquisition and application tools provided. It is suggested that such systems provide a major knowledge-based technology with commercial implications and applications going beyond those currently envisioned for expert systems.

INTRODUCTION

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

Knowledge acquisition and transfer processes in human society are complex and poorly understood. Many disciplines are involved in their study but none provides a comprehensive framework let alone overall theory. Neurology, psychology, linguistics, education, sociology, anthropology, philosophy and systems theory all have significant contributions to make. However, integrating them, disambiguating jargon, and combining different objectives and perspectives are major intellectual problems. The task is attracting increasing effort but this is unlikely to have significant short-term impact on knowledge-based system development.

However, the fundamental problems of understanding knowledge processes should not blind us to the commercial opportunities offered by the development of knowledge-based systems. These go far beyond the expert system shells and knowledge bases currently being marketed or developed. The integration of knowledge acquisition and transfer technologies with expert system shells is currently being undertaken to aid the rapid prototyping of expert systems. The initial objective of systems such as KITTEN [13], KSSO [8] and AQUINAS [5] is to expedite the process of knowledge acquisition and transfer to expert systems. However, the incorporation of group interaction by multiple experts with these systems, and the availability of both acquisition and application tools on line makes available a new type of system that supports a variety of knowledge processes.

This paper develops the concept of knowledge support systems, giving design objectives, some detailed design considerations, and examples of practical systems that have been implemented to date.

AUTOMATING KNOWLEDGE ENGINEERING

Problems of knowledge engineering have been recognized since the early days of expert systems. The initial approach to the development of expert systems has been that shown in Figure 1 where an expert is interviewed by a knowledge engineer who encodes the elicited expertise in a knowledge base which is then used to provide expert advice to users through an expert system shell.

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 be 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 lead 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 labor 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 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 interactive knowledge acquisition systems to automate knowledge engineering as a process of direct interaction between experts and the computer. In 1980 we proposed that personal construct psychology [17,26] could provide foundations for expert systems, and gave examples of algorithms and programs that extracted inference rules from repertory grid data [10]. 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 [30]. This work led to industrial studies of the methodology applied to the development of expert systems: Boeing Computer Services [2,3,4] and Lockheed Software Technology Center [34] have reported success in applications; and validation has been reported in a statistics domain [14].

Figure 2 shows the application of knowledge acquisition systems such as PLANET [31], ETS [3], MORE [16], SALT [20] and PCS [28,29].

- The experts now interact directly with a knowledge acquisition system that aids them to develop a conceptual structure for their knowledge and express it directly in a computational representation.
- Knowledge engineers now act as managers helping the expert in the initial interaction, having overall control of the knowledge acquisition system and editing the resultant knowledge base. They now have a far lower workload and do not have the responsibilities of an intermediary.
- Multiple experts and multiple knowledge engineers are shown because the use of diverse sources of expertise has proved valuable in practice, and the automation of the knowledge acquisition function makes it possible to manage the complexities of multiple sources and their comparison.

The system shown in Figure 2 has a distinct break between acquisition and application. However, in practice the validation of the knowledge base usually depends on the expert and knowledge engineer becoming users and checking the advice given through the expert system shell against well defined case histories. It is useful for the role switching to be as simple as possible, and this has led to the increasing integration of the knowledge acquisition system with the expert system shell as noted in the previous section. It then becomes less and less meaningful to distinguish between 'experts', 'knowledge engineers' and 'users'.

Figure 3 shows the integration of the system with a central core of knowledge acquisition, knowledge base and expert system shell interacting with a community of users in diverse, mixed and varying roles. It is this that we have termed a *knowledge support system* because it potentially supports such a wide variety of knowledge processes. In particular, it allows the community to share knowledge in an explicit form, keep it updated from a variety of sources and use it flexibly, comparing and contrasting different perspectives.

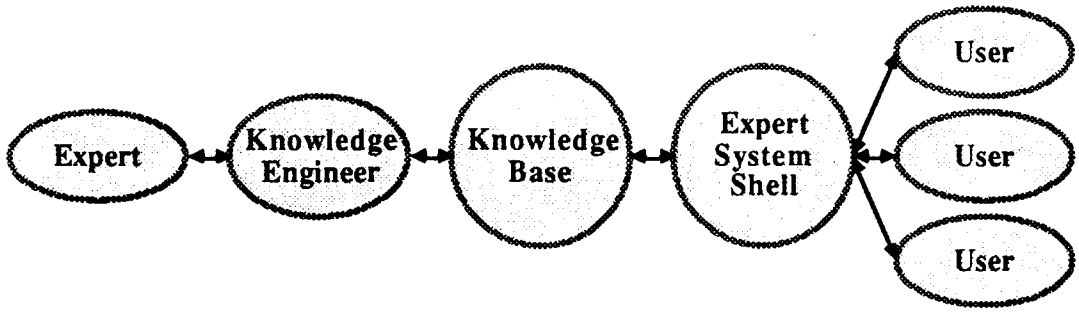

Figure 1 Manual development of an expert system

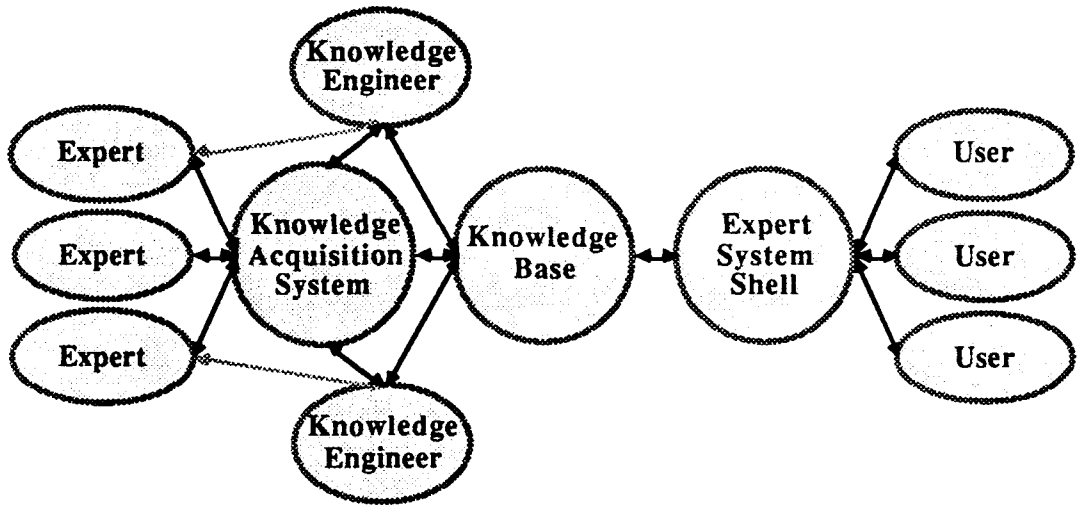

Figure 2 Automated knowledge acquisition for an expert system

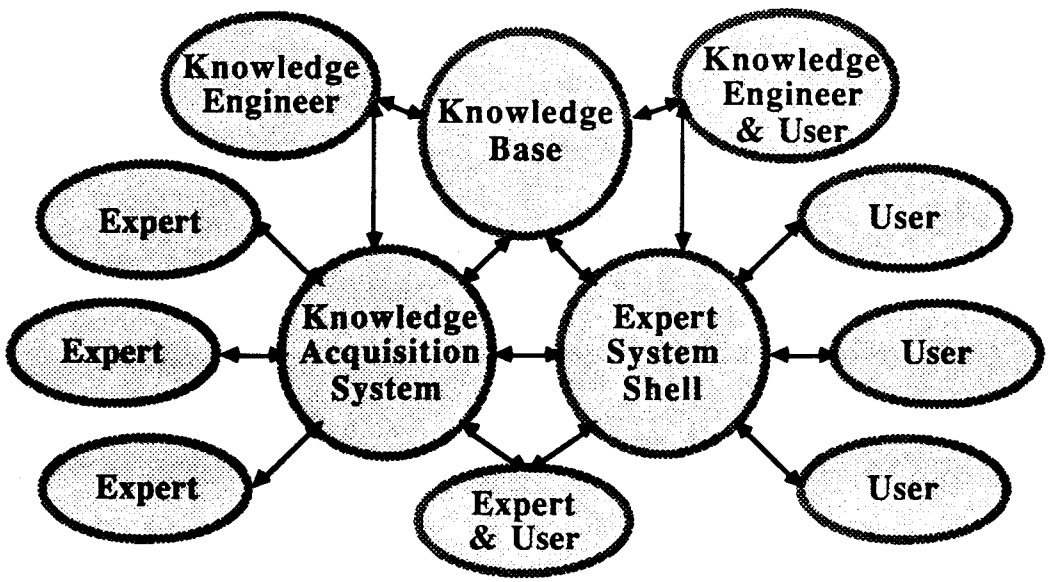

Figure 3 Integration of knowledge acquisition and expert system shell

The shell provides a number of facilities that interact with acquisition and these are shown explicitly in Figure 4 as part of the basic tools of a complete knowledge support system:-

- the *application system* at the bottom tests the user's hypotheses or seeks to satisfy his goals by deriving the consequences of facts about a particular situation reported;
- the *explanation system* to the lower left answers queries about the way in which facts have been derived in terms of what information and inference rules have been used;
- the *acquisition system* to the upper left provides tools for interviewing the expert to obtain his conceptual framework, and general facts and inference procedures in terms of it;
- the *display system* at the top provides tools for presenting the knowledge base in an understandable form, and the relations between the facts and inference procedures;
- the *edit system* to the upper right provides tools for editing the knowledge base while maintaining its integrity in terms of the vocabulary, variables and operations used;
- the *validation system* to the lower right provides tools for checking the knowledge base against specific case histories with known consequences.

Most current expert system shells:

- offer a limited knowledge base represented in terms of production rules and frames;
- have weak application capabilities limited to deriving consequences and testing assertions;
- give simple explanations in terms of the facts and rules used;
- have no integrated acquisition capabilities;
- have limited capabilities to display the knowledge base, sometimes in graphical form;
- have a user interface to the knowledge base editor which requires understanding of internal data structures;
- have limited validation capabilities and no integrated database of case histories.

However, rapid improvements are taking place in the knowledge representation, inferencing, explanation, display and editing capabilities of expert systems [15,22,23,25,33]. These coupled with developments in interactive knowledge acquisition systems provide the technical foundations for a range of products based on knowledge support systems.

The remainder of this paper outlines our experience in developing the integrated knowledge support systems, KITTEN (Knowledge Initiation & Transfer Tools for Experts and Novices) and KSSO.

KNOWLEDGE SUPPORT SYSTEM DESIGN

The general requirements for a knowledge support system are:

- the tools should be domain independent;
- the tools should be directly applicable by experts without intermediaries;
- the tools should be able to access a diversity of knowledge sources including text, interviews with experts, and observations of expert behavior;
- the system should be able to encompass a diversity of perspectives including partial or contradictory input from different experts;
- the system should be able to encompass a diversity of forms of knowledge and relationships between knowledge;
- the system should be able to present knowledge from a diversity of sources with clarity as to its derivation, consequences and structural relations;
- users should be able to apply the knowledge in a variety of familiar domains and freely experiment with its implications;
- the system should make provision for validation studies;
- as much of the operation of the 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.

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

Figure 5 shows the structure of KITTEN, an integrated knowledge support system consisting 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 [13]. 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 [12].

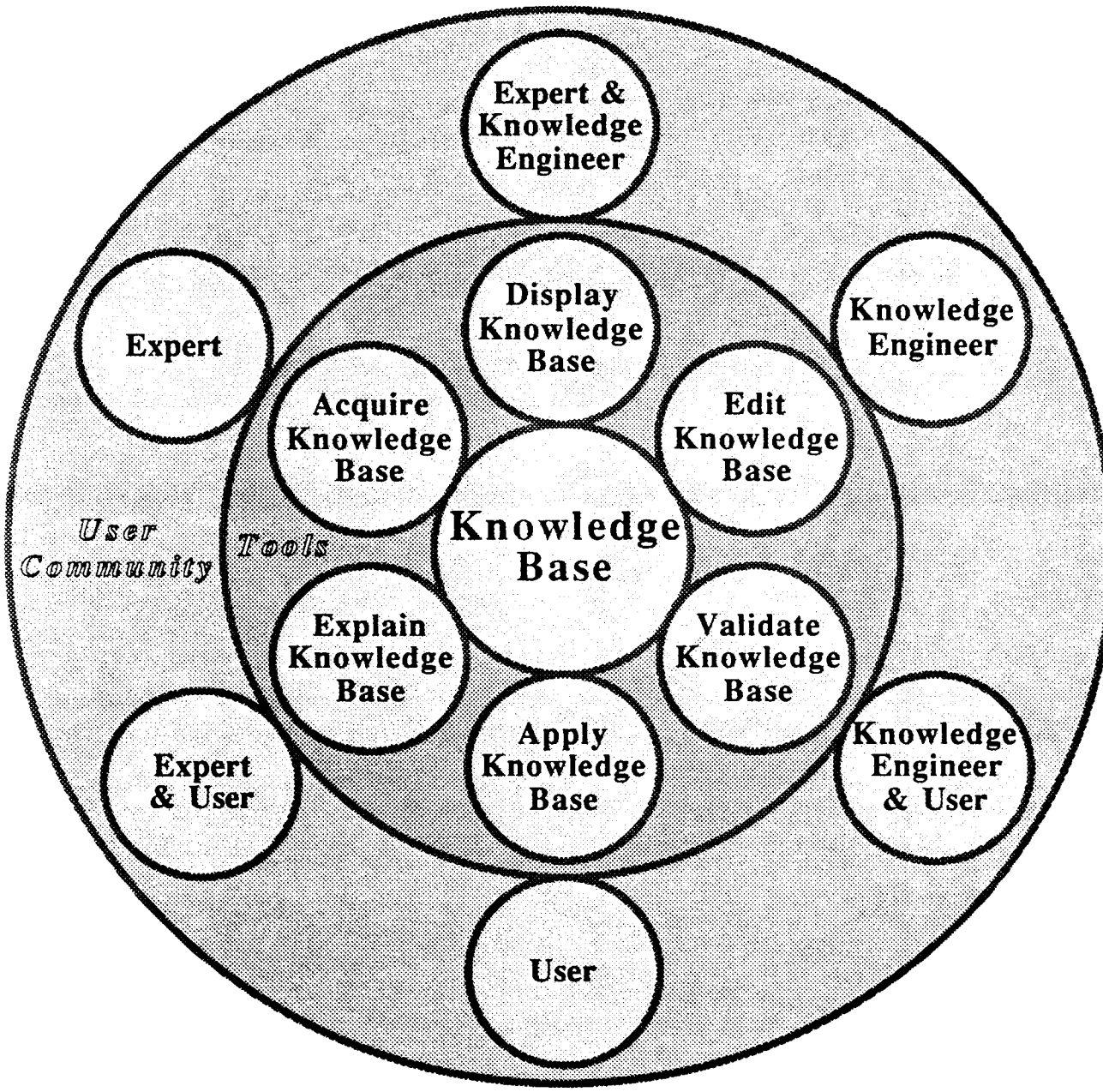

Figure 4 Structure of a full knowledge support system

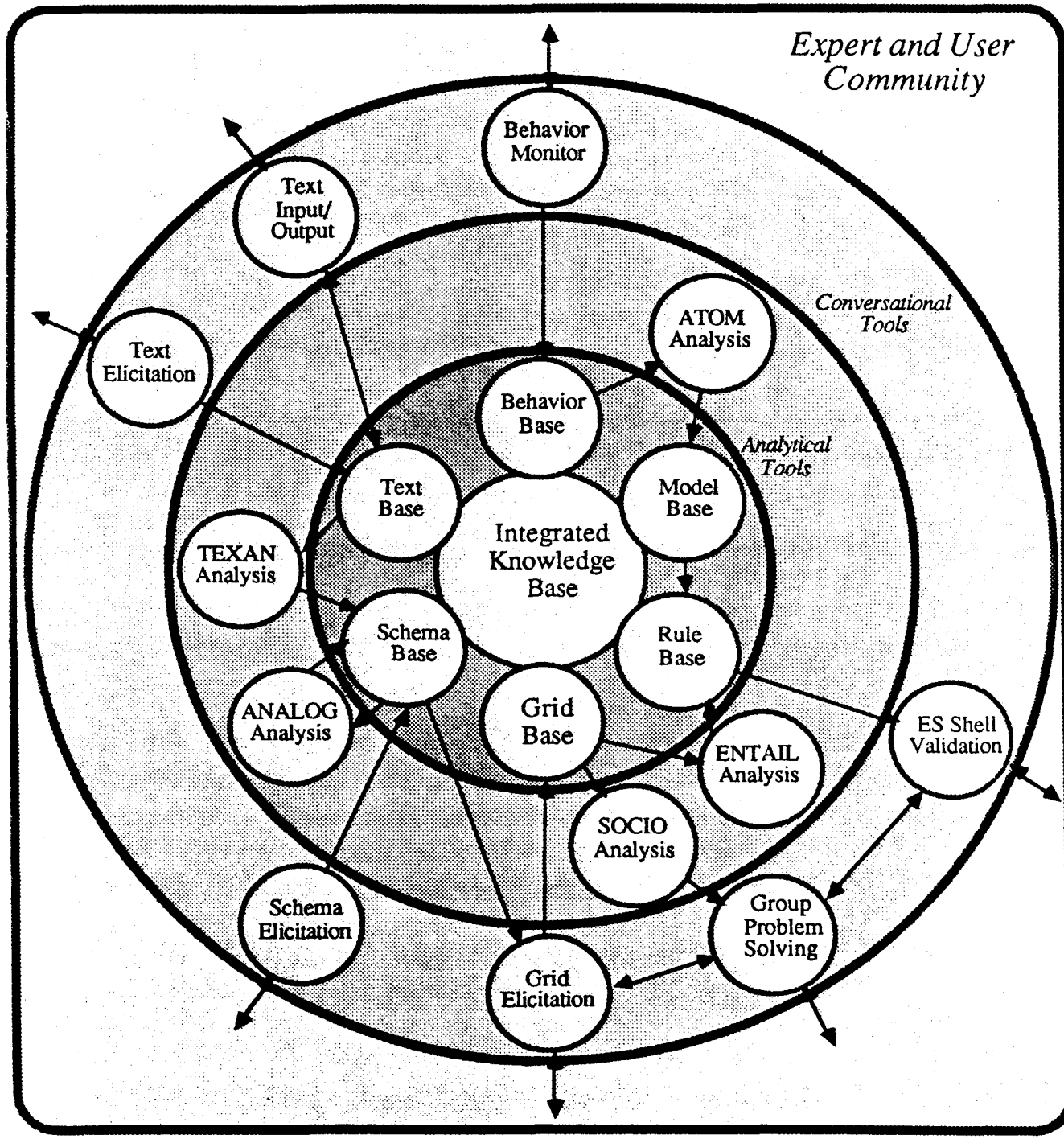

Figure 5 KITTEN—Knowledge Initiation and Transfer Tools for Experts and Novices

- 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 [6].
- 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 [24,25] 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 knowledge processes. 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 [26,27].

The role of these different knowledge acquisition paths in KITTEN is made clear in Figure 6 which contrasts the classical system development paradigm on the left with the expert system development paradigm on the right.

The classical approach to developing a decision or control system is to instrument the system about which decisions are to be made, collect data about its behavior, form a model of the system from this data, and then design a decision or control system based on this model. This has proved very effective in science and engineering.

However, there are many systems where such an approach is not possible, yet human operators are able to undertake decision-making or control. The expert systems approach to developing a decision or control system is to model the human operator rather than the system. The obvious approach is *interview analysis*, and this has been used in many studies. However, often the operator is unable to give a verbal description of his skill, or suggests an incorrect basis for it, and *behavior analysis* through computer modeling of his input-output behavior gives better results [21]. *Text analysis* of the operator's handbook is another significant source of information about his intended behavior.

This is the basis of the interview, behavior and text analysis tools included in KITTEN. Each of the tools included 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. KITTEN is being implemented on an Apollo workstation network as an initial knowledge support system prototype offering a workbench with minimal integration of the knowledge base. KSS0 is an implementation on a Macintosh personal computer network which is already in use at a number of test sites. The following sections describes and illustrate some of the major knowledge elicitation tools in the KSS0 and KITTEN knowledge support systems.

REPERTORY GRID ELICITATION

The elicitation of knowledge from experts through general, domain-independent techniques depends on the availability of a framework within which to represent knowledge. This needs to be well-founded in terms of both cognitive psychology and computer science. Kelly's *Personal Construct Psychology* provides a systemic model of the cognitive and behavioral processes of the individual [17].

A person is modeled as *construing* his experience by dividing it up into chunks (elements) and classifying these using a system of bipolar constructs. Constructs are themselves construed by other constructs and hence form a hierarchical system. Construing is used to enable a person to anticipate the future, and constructs are changed as a result of errors in anticipation. A person will be more willing to change some constructs rather than other *core constructs*, giving a further order relation on the construct system. Learning is seen as the development of the construct system, social processes are seen as the development of shared constructs, and problem-solving is seen as the active anticipation of the future.

The theory provides complete foundations for human psychological and social processes. Hence it is possible to build technologies based upon it which are very general in concept and application. An expert system encodes the relevant construction system of an expert in an active form. In computational terms the elements in personal construct psychology may be seen as entities, the constructs as attributes and the relation between entities and attributes as values in an entity-attribute database. Kelly terms such a database a *repertory grid.*

In the past the elicitation of repertory grids has been a task requiring a skilled psychologist who can draw out a person's constructs whilst not imposing his own ideas or personality. In KSS0 and KITTEN constructs are elicited through direct interaction with the computer using the natural click-and-drag techniques of the window/mouse interaction. The following section exemplifies these techniques in action.

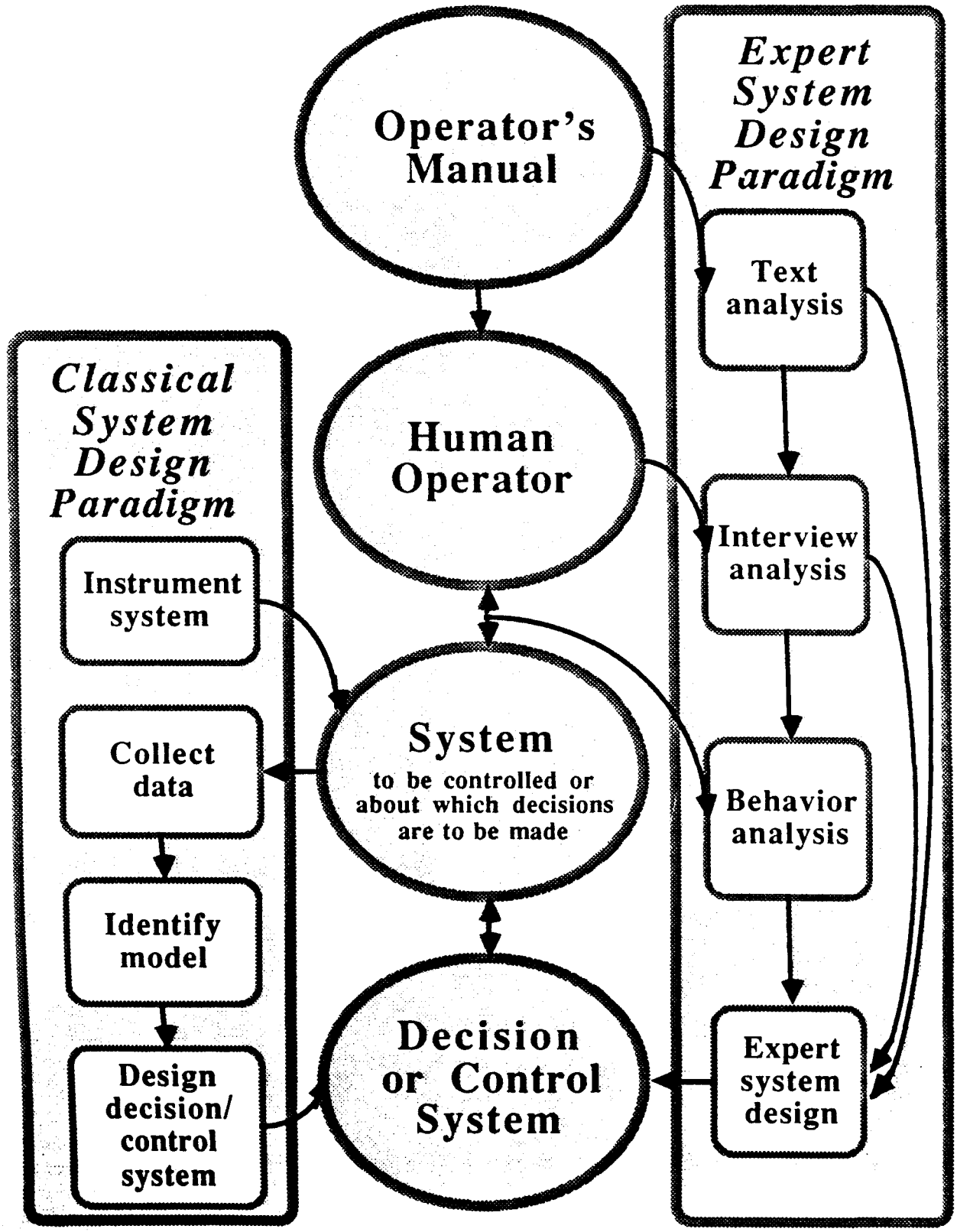

Figure 6 Contrasting classical and expert systems approaches to system development

KSS0 IN OPERATION

Figure 7 is the startup screen for KSS0. The system is designed to be very clean and simple to use with a substantial entertainment value through the use of digital speech and animation. The objective is to make interaction with system attractive and fun while retaining a professional working environment. Spoken and visual help are available at all times and context sensitive manual sections are stored on the bottom of each screen so that they are accessible through scrolling. The expert should never be at a loss, need help he cannot get, or have to access a manual.

The default action at all times is to click on the button at the top right of the lower control panel—usually "OK", "Yes", "More" or the most likely action. During elicitation, when the user has a choice of actions, this default action will select one for him by biassed random choice from a reasonable set generated by rules which can be edited by the knowledge engineer. Thus, experts can begin to make effective use of the system very quickly with no need for computer experience and very little hand holding.

Figure 8 is an entity screen from KSS0 in a knowledge elicitation sequence concerned with factory layout planning [1]. KSS0 allows its vocabulary to be tailored to the problem to maximize expert involvement, and the expert and knowledge engineer have agreed on a basic terminology. The expert has entered various "factors" which affect layout—these are the entities, and we will see the system help him to elicit relevant "qualities"—these are the attributes.

The system keeps track of matches between entities and between attributes which indicate structures in the knowledge, and the expert has decided to investigate one of these. The indicator at the lower left of Figure 8 shows the value of the highest match between entities. The expert has clicked on it to highlight the matching entities. He can use the "-" and "+" buttons to step backwards and forwards through the matches.

He next clicks on the "Break" button to see whether he can input an attribute that will reduce the match—that is, a counter-example to the inferred structure. KSS0 takes him to the screen shown in Figure 9 which makes explicit the question he has to answer and shows him how the match has come about. The horizontal bars show the ratings of the entities on the attributes, with those of the upper entity ("location of services") being shown by the upward markers, and those of the lower entity ("area of department") being shown by the downward markers.

The expert can edit these ratings by dragging the markers along the scales, or modify the entity names by clicking on them to open up a text edit window. At no time in KSS0 is he constrained to follow a specific course of action since the system is designed to facilitate free exploration and development of knowledge structures. In this case the expert decides to complete his action by answering "OK" to the question asked. KSS0 then takes him to the screen shown in Figure 10 where he is prompted to define a new attribute and then rate all the entities on it by clicking and dragging them to the vertical rating bar in the center.

Figure 11 shows the start of the complementary sequence in which the expert starts from the attribute screen and clicks on the indicator to see the highest matched attributes. He then clicks on the "Break" button again and moves to the screen shown in Figure 12 where the basis of the match is shown and again a very specific question is asked about the type of entity required to break the match. He can again edit the ratings by clicking and dragging the entities if he wishes. Answering "OK" to this question takes him to screen 13 where he can enter the name of the new entity and then rate it on each of the existing attributes.

Figure 14 shows a different form of attribute elicitation initiated when the expert clicks on the "Triple" button in Figure 8. KSS0 generates a triple of three entities and asks him in what way two are alike and differ from the third. He clicks on "worker amenities" and KSS0 takes him to the rating screen shown in Figure 15 with the triple pre-rated, so that he may enter the attribute and the ratings of the other entities. The triples are generated on a random basis biassed to promote variety, and this mode of elicitation tends to trigger recollections of significant attributes otherwise forgotten.

At any time during the elicitation the expert can ask for the structure of his entity-attribute conceptual system to be analyzed. Figure 16 shows a hierarchical clustering of his entities and attributes based on the matches between them. Figure 17 shows an alternative mapping into two-dimensions based on a principal components analysis. Figure 18 shows a third form of analysis based the logical *entailments* in his data [10,12]. This third form is used to generate rules with which to prime an expert system shell.

Figure 19 shows two of the clustering analyses being used in the middle of an elicitation to see the way in which a conceptual structure is developing. Figure 20 shows the logical entailments being used within the expert system shell, *Nexpert*, in a prototype expert system based on the conceptual structures whose elicitation has

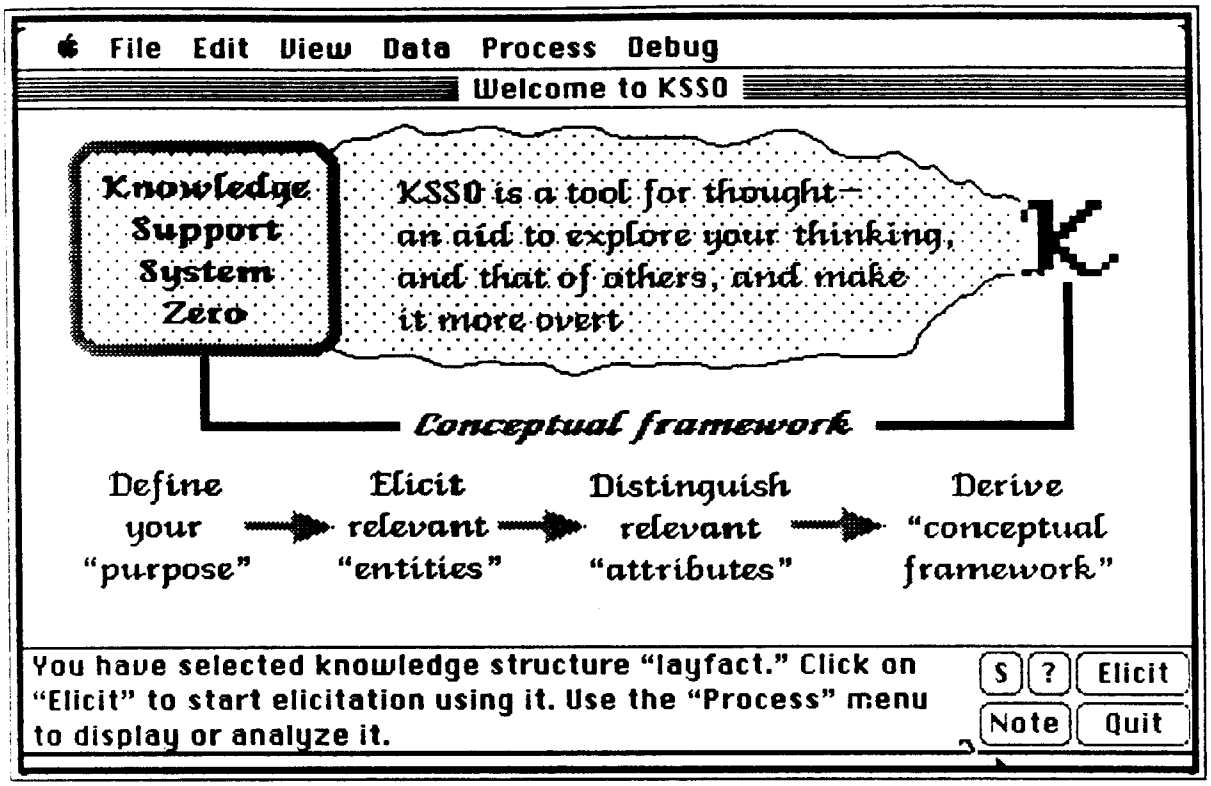

Figure 7 KSS0 startup screen showing process of elicitation

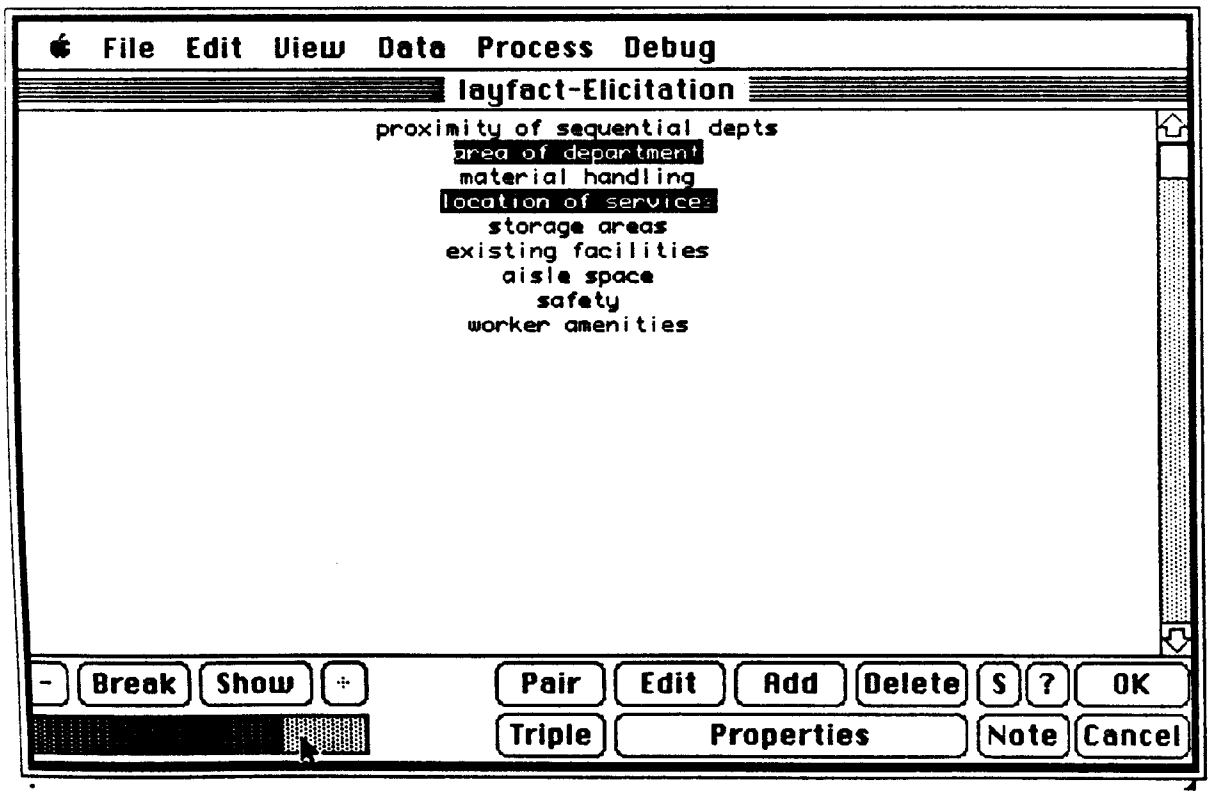

Figure 8 KSS0 entity screen showing matching entities

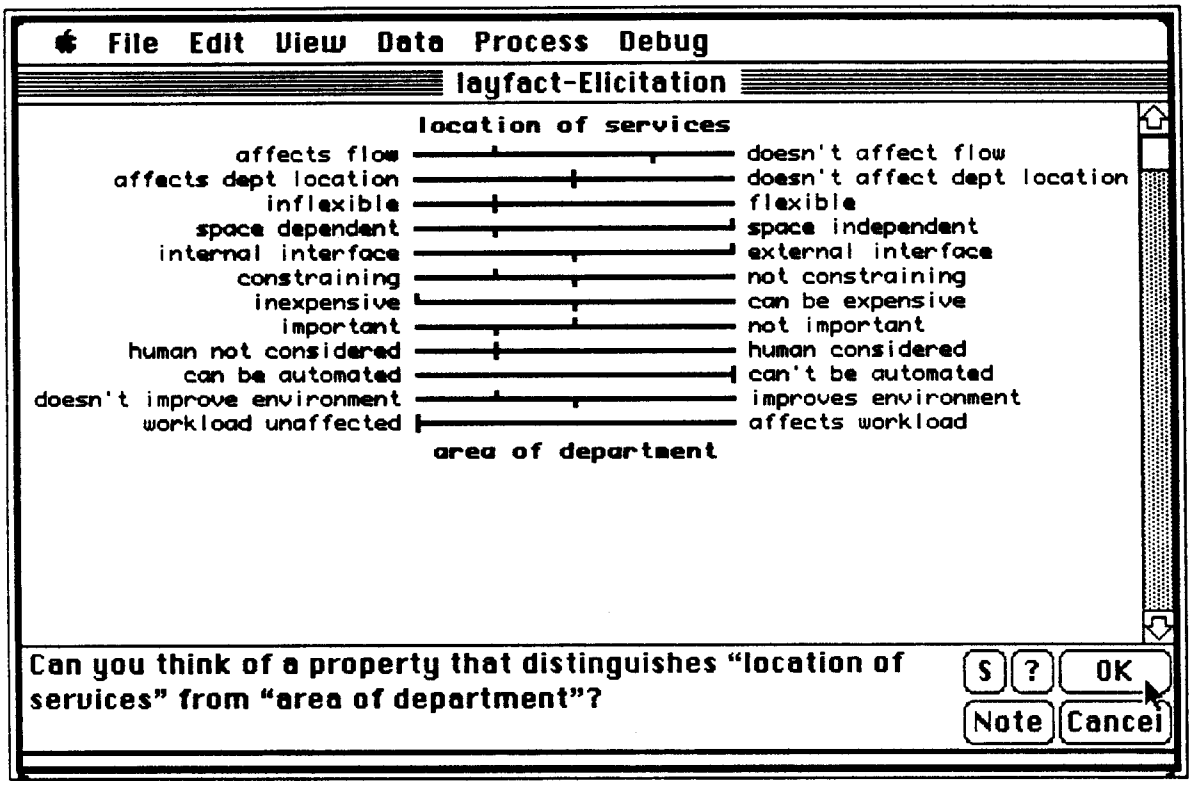

Figure 9 KSS0 entity match screen prompting new attribute

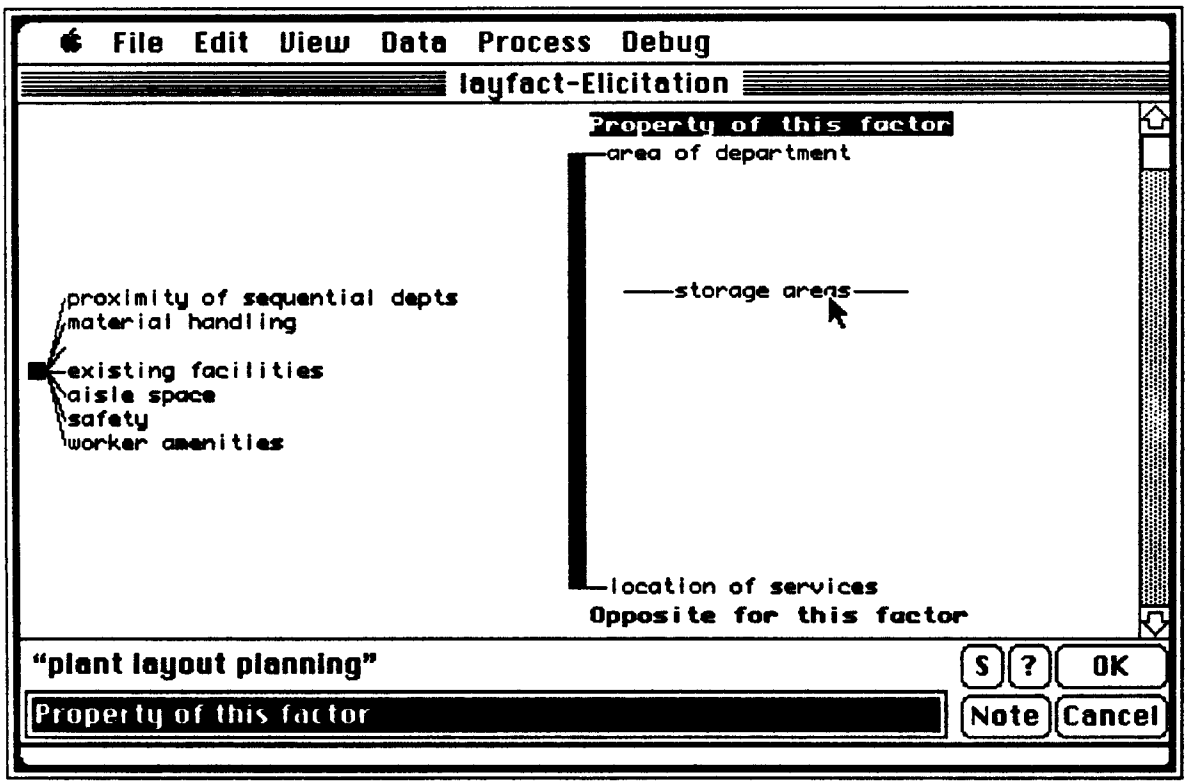

Figure 10 KSS0 attribute entry and rating screen

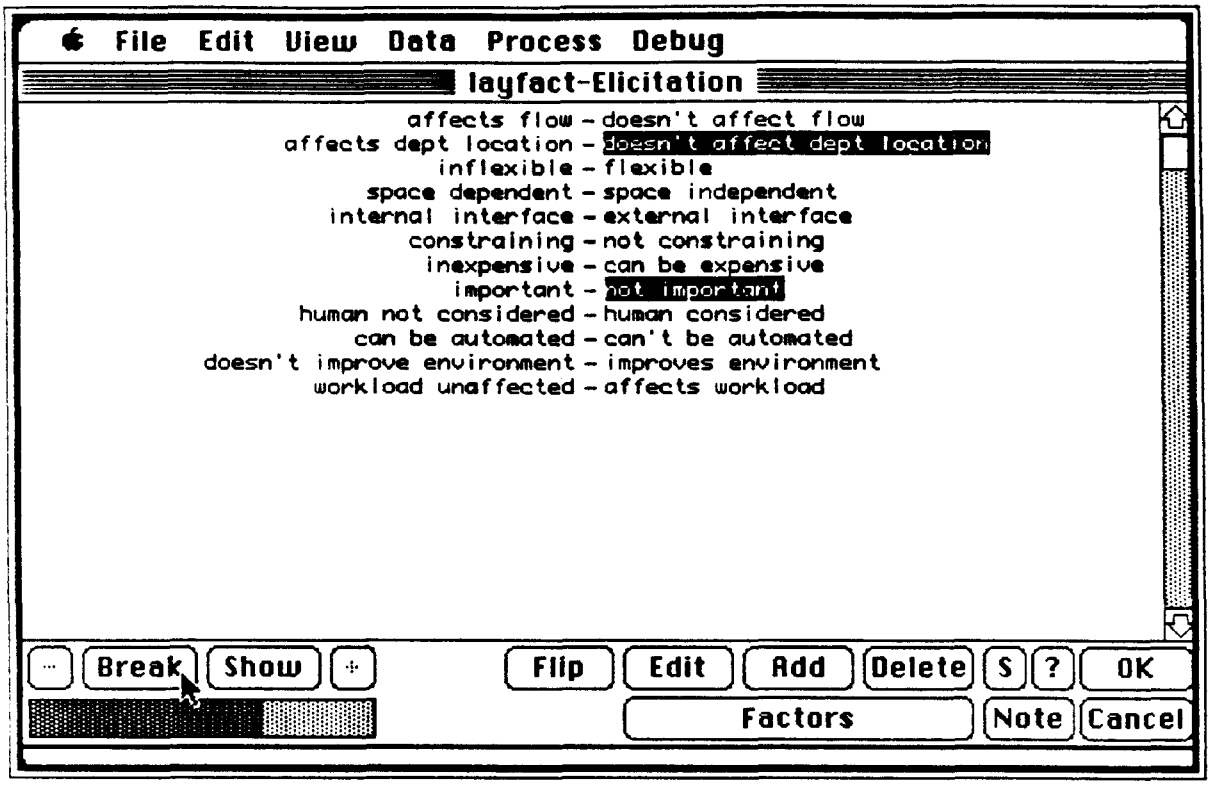

Figure 11 KSS0 attribute screen showing matching attributes

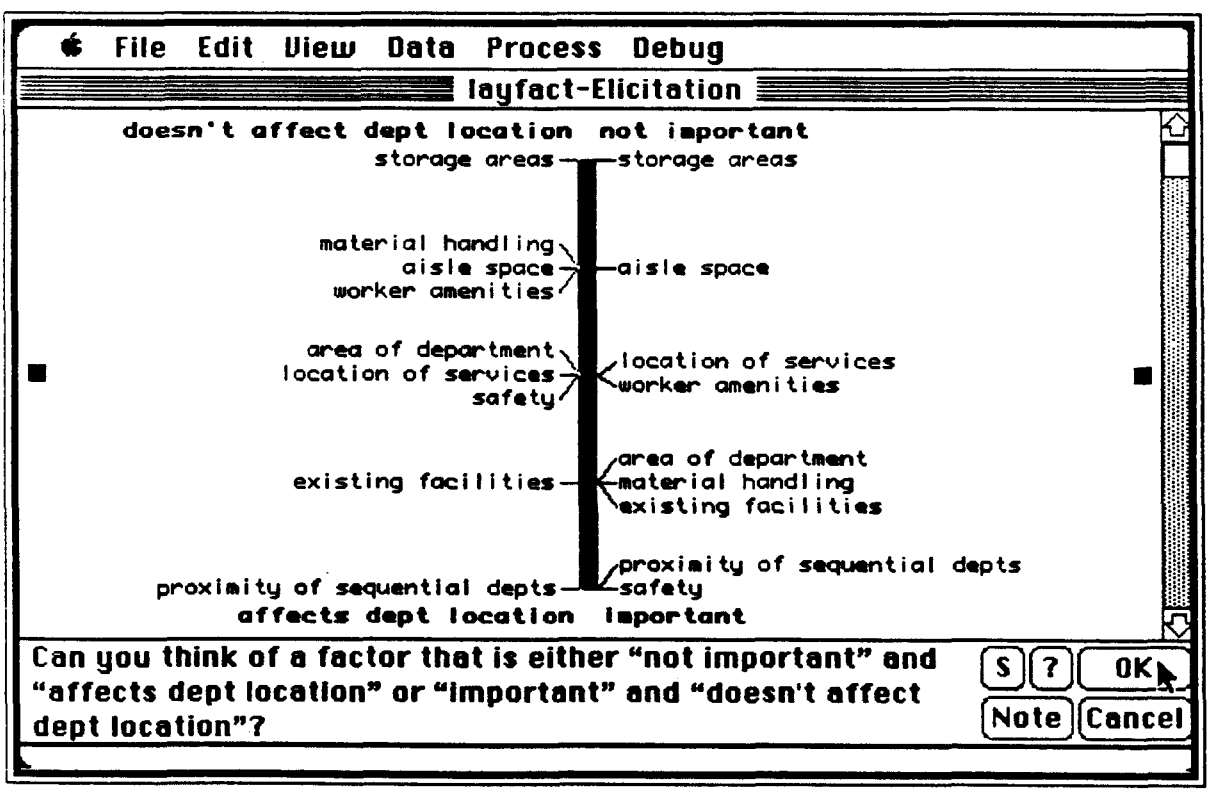

Figure 12 KSS0 attribute match screen prompting new entity

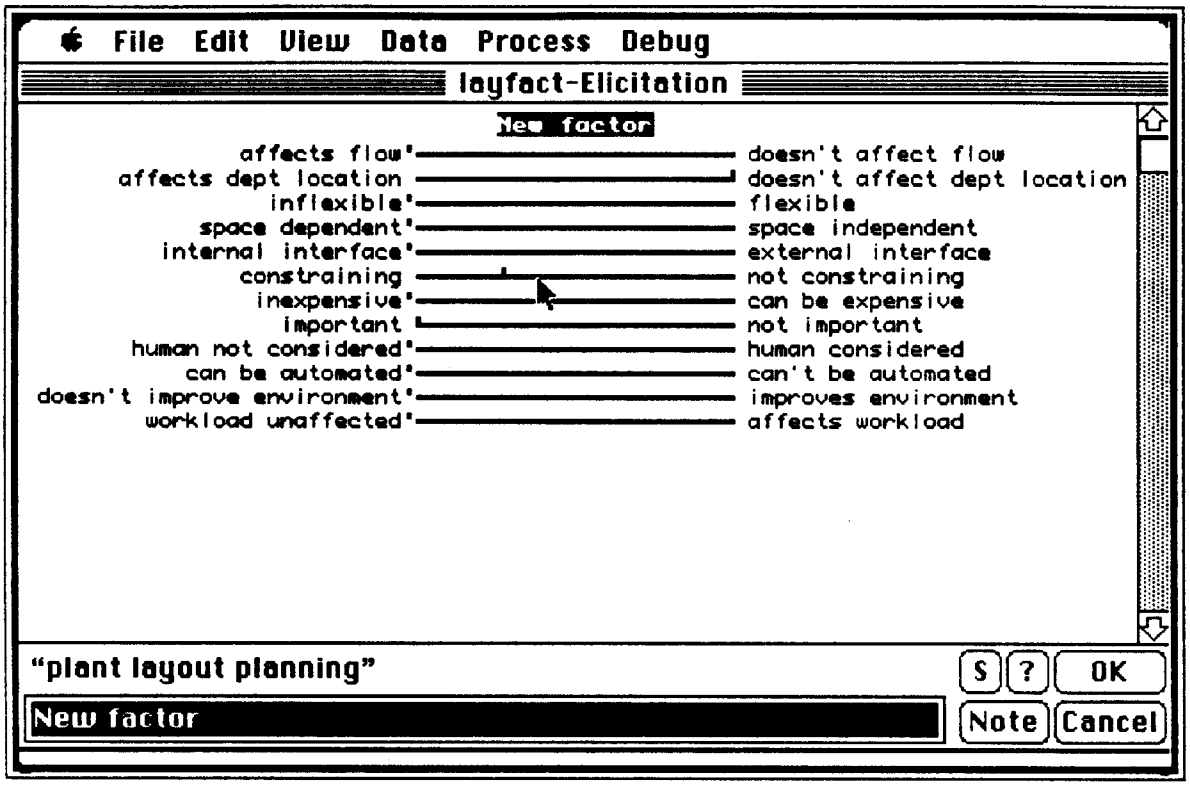

Figure 13 KSS0 entity entry and rating screen

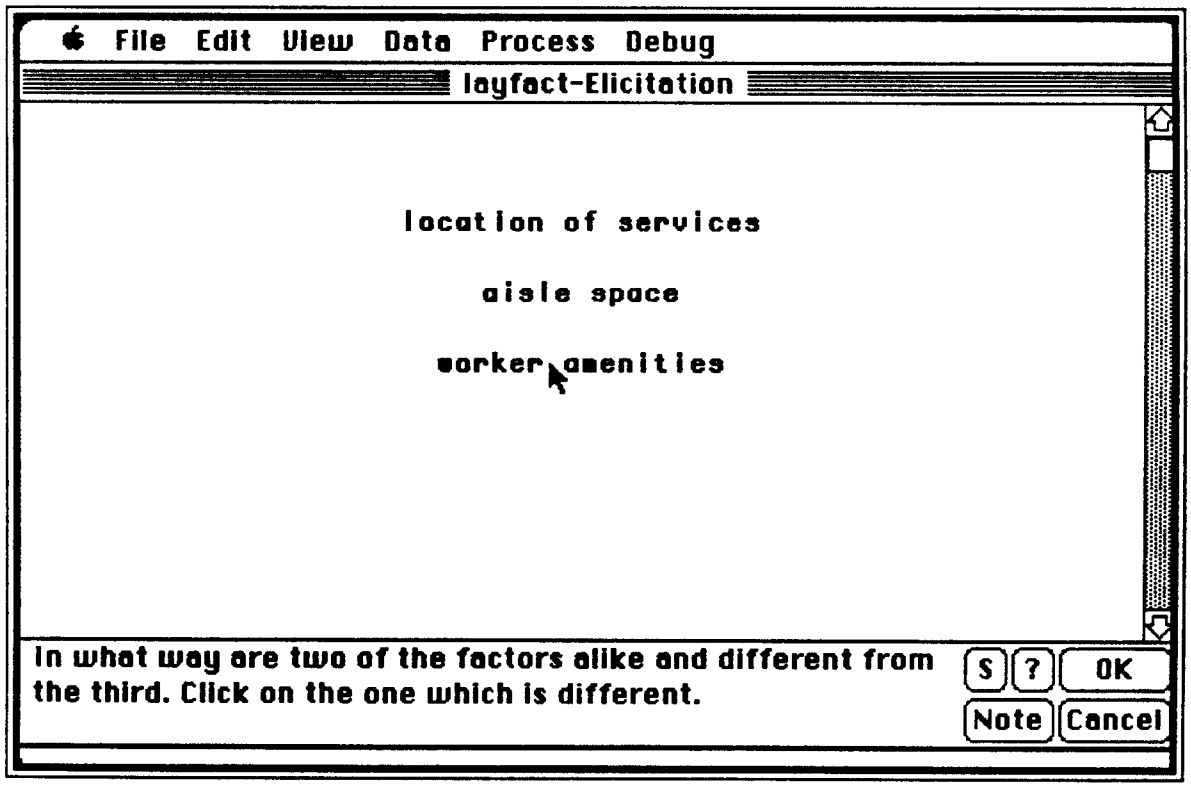

Figure 14 KSS0 attribute elicitation from a triple of entities

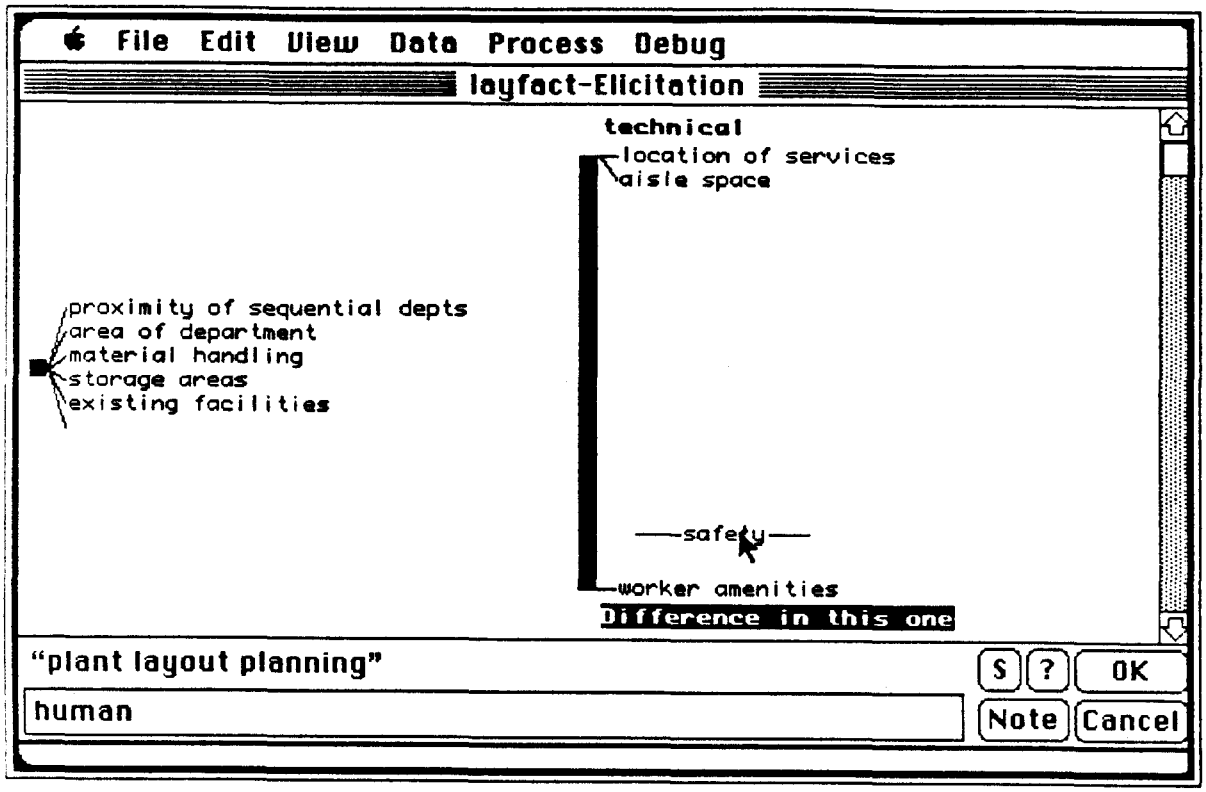

Figure 15 KSS0 attribute entry and rating screen after triple

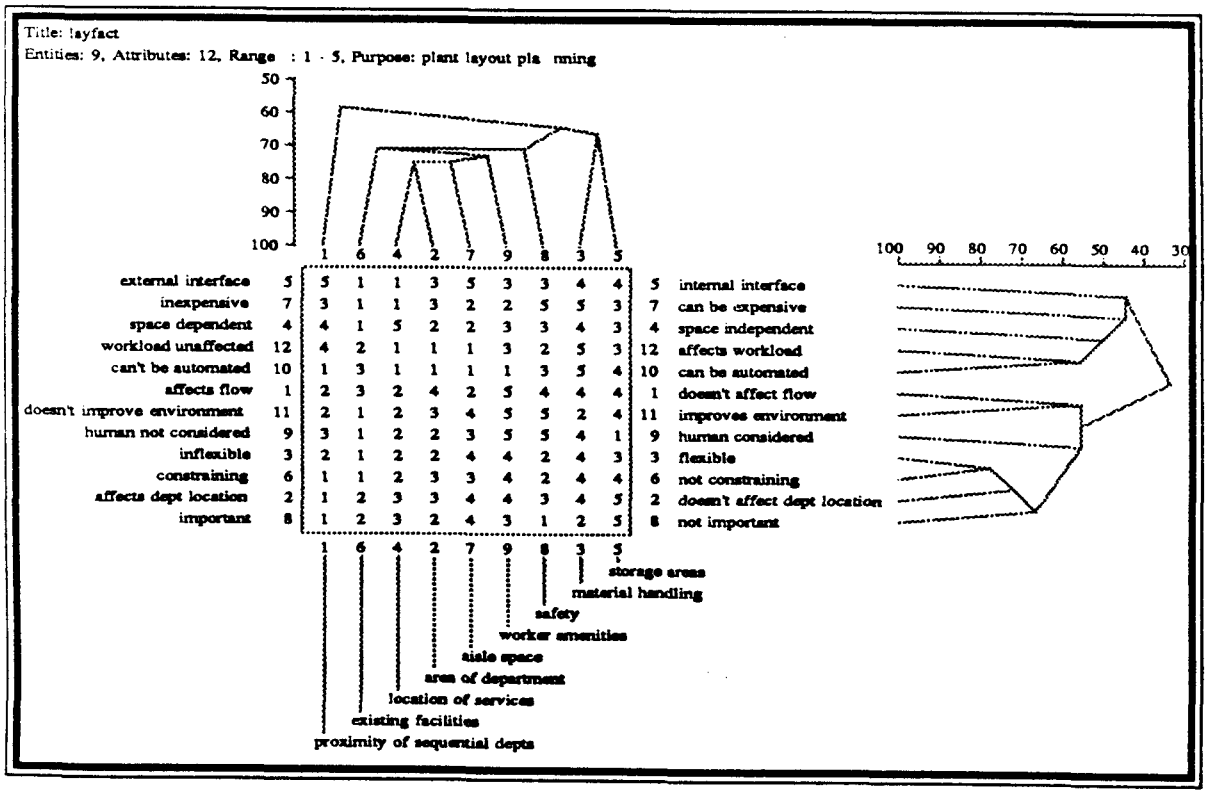

Figure 16 KSS0 hierarchical clustering of entities and attributes

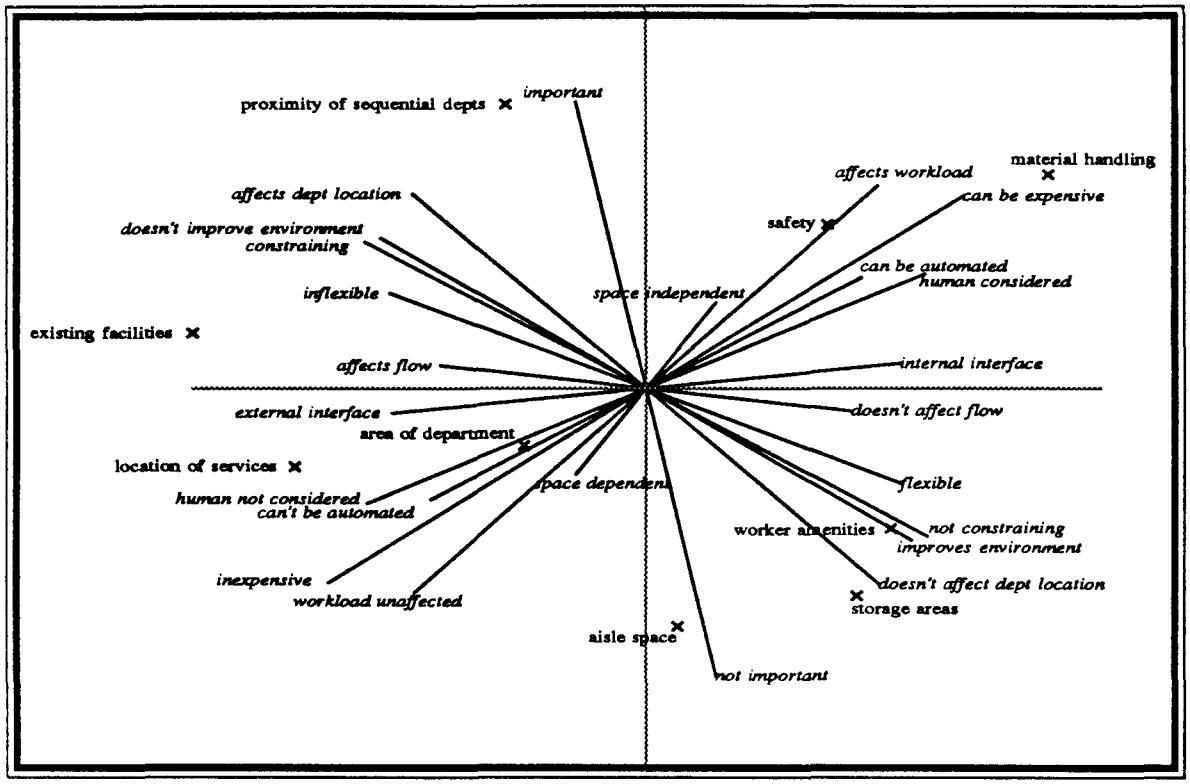

Figure 17 KSS0 principal components map of entities and attributes

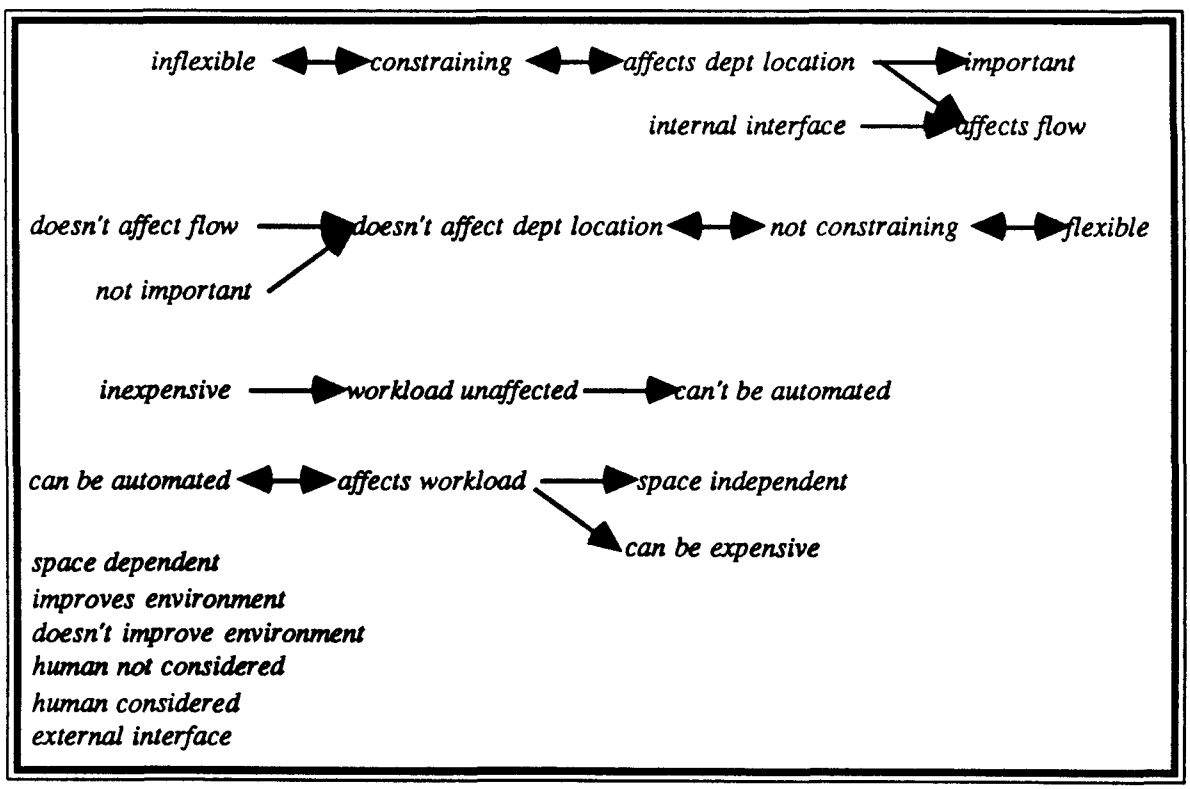

Figure 18 KSS0 logical analysis of entailments

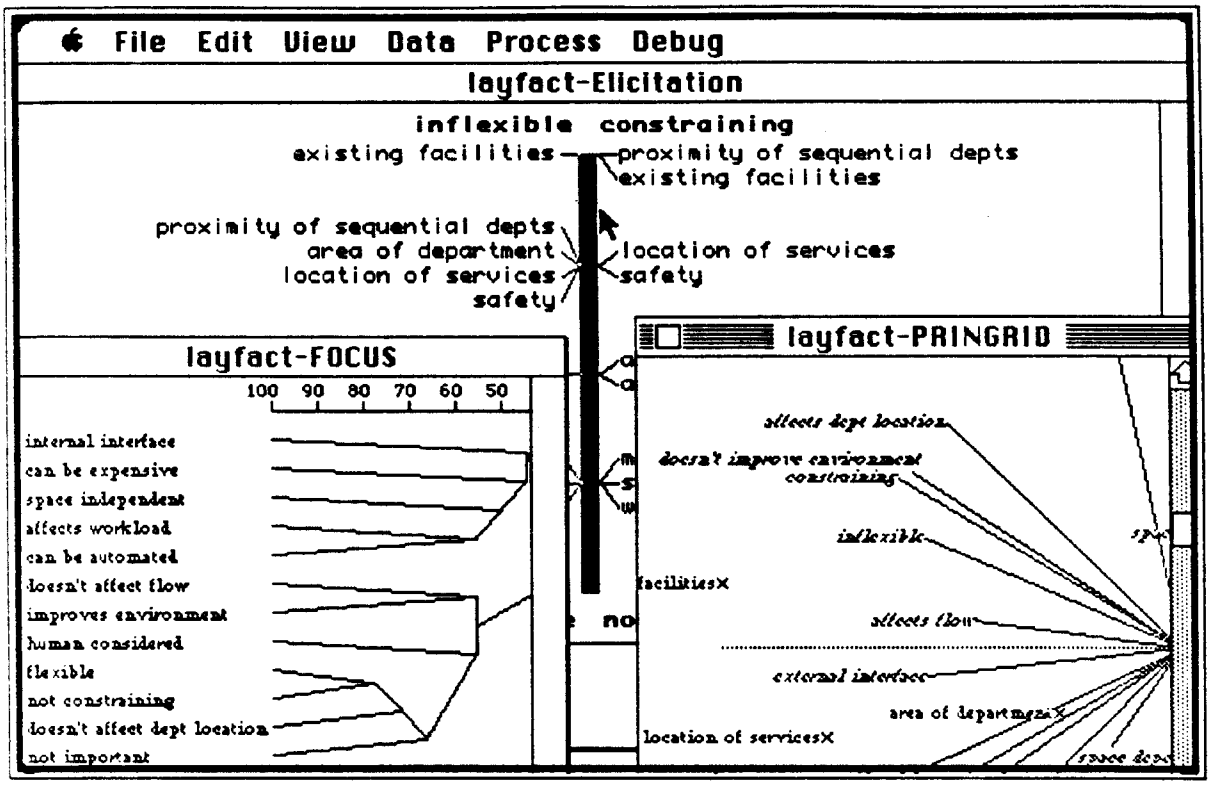

Figure 19 KSS0 access to clustering during elicitation

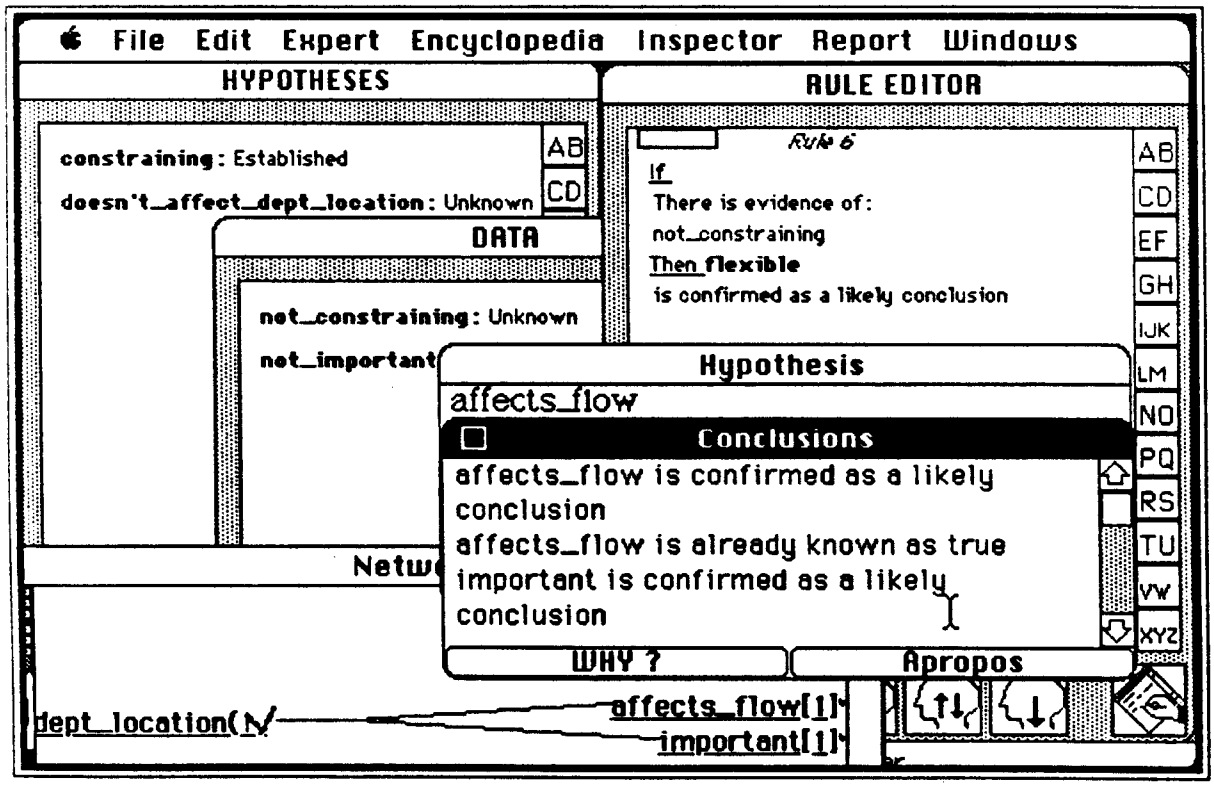

Figure 20 Nexpert operating with rules generated by KSS0

been shown together with rules relating these concepts to layout decisions.

The technical details of the processes going on in Figures 8 through 19 are less important than the psychological impact on the expert. Users of KSS0 generally find that it gives fascinating insights into their own thought processes which both makes it attractive and puts them into an introspective frame of mind conducive to knowledge elicitation. Most of the dialog they see on the screen is generated from their own inputs and hence appears natural in terms of their professional terminology. Most importantly, this translates immediately into the rules generated for use in the expert system shell—they use the expert's jargon and their derivation from the data entered has already been shown to the expert and changes negotiated.

Thus, the initial prototype system, such as that shown in Figure 20, is comprehensible to the expert in its derivation and terminology. Also, by that stage he has become accustomed the use of the computer—windows, menus and mice are familiar tools—and will often move over to editing the rules in the shell without really noticing that the environment has changed.

A number of other programs that have been developed for stand-alone knowledge elicitation and processing are being incorporated in KSS0 and KITTEN. These are briefly described in the following sections.

HIERARCHIES OF ELICITATION

In most problems environments there are multiple knowledge islands, each requiring a separate elicitation process specific to the knowledge structures within the island. These tend to form a hierarchy, for example "safety" is a general factor in the elicitation of Figure 8 but has its own detailed conceptual structure in terms of a variety of hazards, a variety of forms of accident, a variety of legal implications, and so on. The methodology shown above can be extended to deal with such hierarchies [5].

SOCIO: GROUP ELICITATION

It is important that knowledge elicitation methodologies 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 [5,18]. The SOCIO programs support 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. Such techniques allow the similarities and differences in expert perspectives to be rapidly identified.

ATOM: BEHAVIOR ANALYSIS

Michalski and Chilausky have demonstrated that inductive modeling of an expert's behavior may produce effective rules when those elicited by interview techniques are clearly inadequate [21]. 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 [8]. We have incorporating a version of ATOM 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.

TEXAN: TEXT ANALYSIS

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 designed to map subject matter concepts against student concepts in computer-managed instruction systems [32]. 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 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.

Figure 21 shows a TEXAN clustering of an evaluation study of data logging, analysis and presentation methodologies for human performance evaluation in complex systems [9]. 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.

ANALOG: SCHEMA ANALYSIS

The groupings of Figure 21 when combined with the construct classifications of repertory grids may be viewed as schema structuring a knowledge domain.

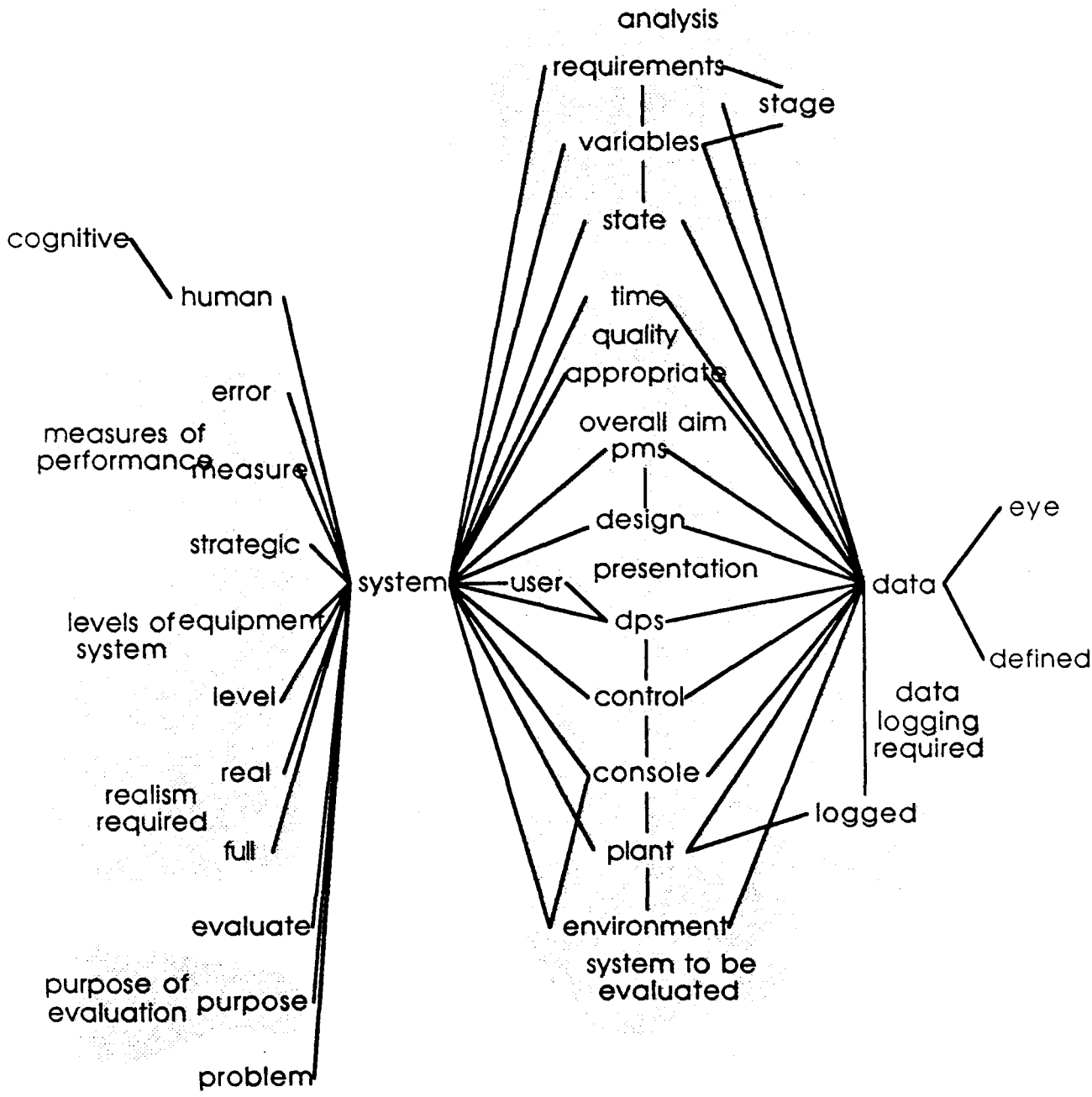

Figure 21 TEXAN clustering of word associations from text with annotation showing knowledge islands

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 as pullbacks of faithful functors between categories [11] 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 [19]. 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.

CONCLUSIONS

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 some preliminary implementations. These consist of sets of knowledge engineering tools, some of which already have track records of successful use in knowledge acquisition studies. In developing the knowledge support systems 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. 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.

ACKNOWLEDGEMENTS

Financial assistance for research on knowledge support systems has been made available by the National Sciences and Engineering Research Council of Canada. KITTEN is being developed in the Knowledge Science Institute at the University of Calgary. KSSO is being developed by the Centre for Person-Computer Studies, Calgary. PCS is being developed at the Alberta Research Council. We are grateful to our colleagues Alain Rappaport at Neuron Data, Jeff Bradshaw at Boeing Computer Services, Ernie Chang at Alberta Research Council, and John Boose at Boeing Computer Services, for discussions relating to knowledge support systems and for access to their own research in this area.

REFERENCES

[1] Adler, J. & Matheson, J. (1986). Expert knowledge elicitation using personal construct theory: obtaining plant layout expertise with 'PLANET'. **BASc Thesis.** Department of Industrial Engineering, University of Toronto (April).

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

[3] 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).

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

[5] Boose, J.H. & Bradshaw, J.M. (1986). Expertise transfer and complex problems using AQUINAS as a knowledge acquisition workbench for expert systems. **Proceedings of AAAI Workshop on Knowledge Acquisition for Knowledge-Based Systems.** 5-0-14-16. Banff (November).

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

[7] Gaines, B.R. (1986). An overview of knowledge acquisition and transfer. **Proceedings of AAAI Workshop on Knowledge Acquisition for Knowledge-Based Systems.** 14-0-14-18. Banff (November).

[8] Gaines, B.R. (1987). Rapid prototyping for expert systems. Oliff, M., Ed. **Proceedings of International Conference on Expert Systems and the Leading Edge in Production Planning**

and Control. pp.213-241. University of South Carolina.

[9] Gaines, B.R. & Moray, N. (1985). Development of performance measures for computer-based man-machine interfaces. **Technical Report** DCIEM-PER-FIN:JUL85.

[10] 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 (July).

[11] Gaines, B.R. & Shaw, M.L.G. (1982). Analysing analogy. Trappl, R., Ricciardi, L. & Pask, G., Eds. **Progress in Cybernetics and Systems Research**. Vol. IX, pp. 379-386. Washington: Hemisphere.

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

[13] Gaines, B.R. & Shaw, M.L.G. (1986b). Knowledge engineering techniques. **Proceedings of AUTOFACT'86**. 8-79-8-96. Detroit: Society of Manufacturing Engineers (November).

[14] Gammack, J.G. & Young, R.M. (1985). Psychological techniques for eliciting expert knowledge. Bramer, M., Ed. **Research and Development in Expert Systems**, pp. 105-116. Cambridge University Press.

[15] Hayes-Roth, B. (1985). A blackboard architecture for control. **Artificial Intelligence**, 26(3), 251-321 (July).

[16] Kahn, G., Nowlan, S. & McDermott, J. (1985). MORE: an intelligent knowledge acquisition tool. **Proceedings of the Ninth International Joint Conference on Artificial Intelligence**. pp. 581-584. California: Morgan Kaufmann.

[17] Kelly, G.A. (1955). **The Psychology of Personal Constructs**. New York: Norton.

[18] Kernaghan, J.A. & Cooke, R.A. (1986). The contribution of the group process to successful project planning in R&D settings. **IEEE Transactions on Engineering Management, EM33**(3), 134-140 (August).

[19] 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.

[20] Marcus, S., McDermott, J. & Wang, T. (1985). Knowledge acquisition for constructive systems. **Proceedings of the Ninth International Joint Conference on Artificial Intelligence**. pp. 637-639. California: Morgan Kaufmann.

[21] 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.

[22] Michalski, R.S. & Winston, P.H. (1985). Variable precision logic. **Artificial Intelligence**, 29(2), 121-146 (August).

[23] Pearl, J. (1986). Fusion, propagation and structuring in belief networks. **Artificial Intelligence, 29**(3), 241-288 (September).

[24] Rappaport (1986a). Task analysis in intelligent systems design. **WESTEX-86, IEEE 86CH2332-5.** 61-67. Anaheim (June).

[25] Rappaport (1986b). Multiple problem spaces in the knowledge design process. **Proceedings of AAAI Workshop on Knowledge Acquisition for Knowledge-Based Systems.** 37-0-37-15. Banff (November).

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

[27] 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.

[28] 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, IEEE 86CH2364-8**. pp.1353-1357. Atlanta (October).

[29] Shaw, M.L.G. & Chang, E. (1986). A participant construct system. **Proceedings of North American Personal Construct Network Second Biennial Conference**. pp. 131-140. University of Calgary: Department of Computer Science (June).

[30] Shaw, M.L.G. & Gaines, B.R. (1983). A computer aid to knowledge engineering. **Proceedings of British Computer Society Conference on Expert Systems**, 263-271 (December). Cambridge.

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

[32] 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 (July).

[33] Swartout, W. R. (1986). Knowledge needed for expert system explanation. **Future Computing Systems, 1**(2), 99-114.

[34] Wahl, D. (1986). An application of declarative modeling to aircraft fault isolation and diagnosis. Luker, P.A. & Adelsberger, H.H., Eds. **Intelligent Simulation Environments**, pp. 25-28 (January). La Jolla, California: Society for Computer Simulation.