GROUP DECISION SUPPORT SYSTEMS FOR KNOWLEDGE BASED SYSTEMS DEVELOPMENT

M L G Shaw & B R Gaines

Knowledge Science Institute, University of Calgary
Calgary, Alberta, Canada T2N 1N4
(mildred@cpsc.ucalgary.ca, (403) 220-5114,
gaines@cpsc.ucalgary.ca, (403) 220-5901)

Abstract: Knowledge engineering for large scale systems is essentially a *group process* involving multiple requirements specifiers, knowledge engineers, domain experts, system implementation specialists, evaluation specialists and knowledgeable clients. Past research on knowledge acquisition tools has focused on the development of computer-based interactive systems to support *individual* technical activities in knowledge-based system development. Only recently has effort begun to be applied to support of the decision processes of the overall group participating in system development, particularly the interactions between group members. This paper reports work in progress on the extension of existing knowledge support tools, which already provide offline analysis of group-related issues such as conflicts in concepts and terminology, to provide online support of the group decision processes in both its social and technical aspects.

INTRODUCTION

Ever since Feigenbaum (1980) identified the knowledge engineering bottleneck impeding the effective application of knowledge-based systems, knowledge acquisition has been a topic attracting major research effort and resulting in the development of a wide range of methodologies, techniques and tools (Boose & Gaines, 1988, 1990; Gaines & Boose, 1988; Boose, 1989). In terms of the direct knowledge engineering process we now have a rich repertoire of approaches, support systems and experience on which to base further research and practical system development. However, the technological focus has been the support of individuals, such as tools to interview the expert or to support the knowledge engineer, and has neglected the essential social aspects of large-scale system development. As we become more and more concerned with issues of system requirements, integration with existing systems, validation and verification, documentation, enhancement and maintenance, it is becoming clear that the technological focus has to change from the individual to the group.

The next generation of knowledge support systems for large-scale knowledge-based system development has to be conceptualized within a group decision support systems (GDSS) framework. In particular, the separation between the decision support appropriate to system performance applications, and that appropriate to ongoing system development, appears increasingly inappropriate for knowledge-based systems. It is the nature of knowledge to develop through applications and to change continually in the light of reflection on experience. Development, validation and maintenance become a unitary process in knowledge-based systems.

The move towards group decision support has already begun in the knowledge acquisition community. Generic toolboxes encompassing and integrating a range of methodologies such as KSSO (Shaw & Gaines, 1987; Gaines, 1988) and AQUINAS (Boose & Bradshaw, 1987), what have come to be called second-generation knowledge acquisition tools (Gaines & Linster, 1990), already make provision for users with widely varying roles, requirements and capabilities. Liou, Weber and Nunamaker (1990) have

reported research on a methodology for knowledge acquisition in a group decision support system environment. Methodologies have also been developed that manage the elicitation of knowledge from groups of experts and are supported by tools that can analyze relations between the conceptual frameworks, and conflicts in the use of terminologies, for groups comprising multiple experts and clients (Gaines & Shaw, 1989; Shaw & Gaines, 1989).

We have previously emphasized the need to view knowledge acquisition as part of the socio-technical processes involved in knowledge flows in society (Shaw & Gaines, 1989), have developed socio-cognitive models of the way in which various forms of knowledge arise in individuals and society (Gaines, 1989a), and have extended our original concept of knowledge support systems (Shaw & Gaines, 1987) to encompass the wide variety of technologies supporting knowledge flows (Gaines, 1990c). This paper reports on our current research program for developing a group decision support infrastructure that integrates existing tools within a networked environment designed to support the group as a whole as well as individuals playing roles within it.

THE SOCIAL CONTEXT OF KNOWLEDGE ACQUISITION

The split between knowledge acquisition and performance systems is an invidious one. It can be taken to imply that expertise is a static collection of skills and knowledge, capable of being elicited and transferred to an expert system that will then be able to reproduce the expertise. This static model is a dubious one in systems design, leading to systems that date rapidly, interfere with the growth of expertise, have poor reliability and high maintenance costs. The following model explains how these conditions arise, and leads to approaches for avoiding them.

Hawkins (1983) has abstracted from industrial experience in developing mineral exploration expert systems and proposed a model of human expertise relevant to expert systems. Figure 1 presents a summary of the essential features of this model:

- The expert first elicits data about the problem from the client;
- He or she develops a minimal model that accounts for the data provided;
- He or she generates advice based on the model and feeds this back to the client;
- The client may accept the advice, or query it and, possibly, the model;
- The queries lead to further data elicitation, and repeat of the elicitation-modeling-advice-query cycle.

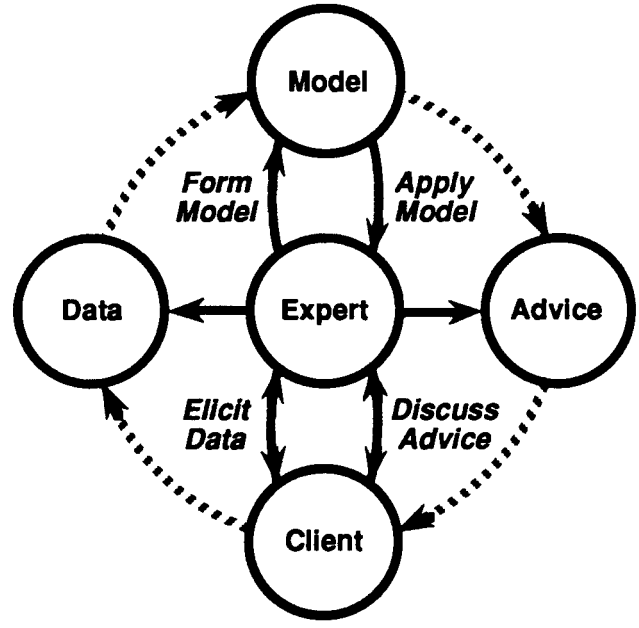

Figure 1 The negotiation cycle in expert-client interaction

Thus, in Hawkins' model, the client plays an active role in further developing the model by providing more data until he or she is satisfied with the model and consequent advice. Expert advice giving and taking is part of a cycle of negotiation around a process of model formation.

The particular importance of this model of expert performance is the emphasis it places on the embedded knowledge acquisition process. The expert is managing a data collection, modeling and decision cycle within a consultation. As Figure 1 shows, this implies that the expert is able to enhance his or her collection of case histories, models and advisory strategies through the consultation. Skilled performance is not the static utilization of expert knowledge and skills in Hawkins' model—it is instead a microcosm, and an essential part of, the wider processes of knowledge acquisition which are intrinsic to the role of an expert.

Access to the learning experience of attempting to solve the problems of a client community is one major source of knowledge in the development of the expertise of an individual expert. The client community provides access to practical experience, a variety of problems, particularly novel ones that go beyond existing expertise, and it also manages the growth of expertise through systems of reward, criticism and access to resources. However, learning from experience is a slow and error-prone process, and socially significant areas of expertise become associated with professional communities that attempt to expedite learning and reduce errors through the sharing of experience. Figure 2 extends the model of Figure 1 to include these other processes, such as apprenticeship, instruction, training, education, workshops, conferences, journals and books. Professional communities usually also play a major role in refining and directing the client community's reward, criticism and resource allocation systems.

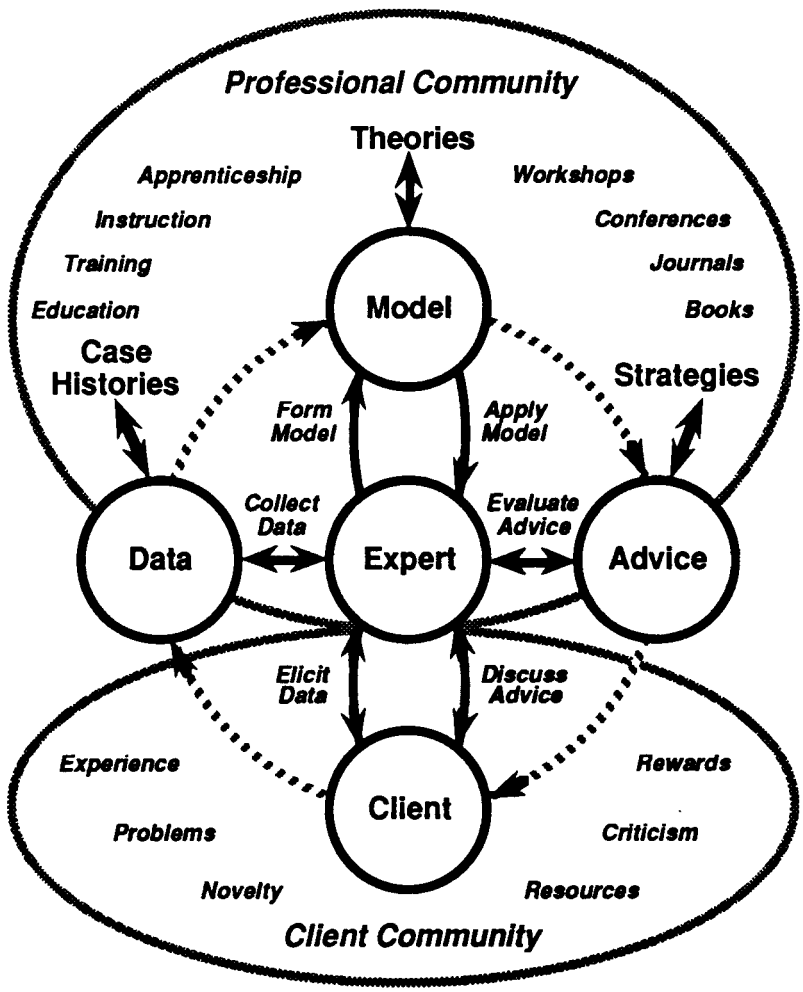

Figure 2 Extended knowledge acquisition process in expert-community interaction

The role of the professional community can be seen as an indirect enhancement of the basic processes of knowledge acquisition in the negotiation cycle of expert performance. The individual can acquire the data of others through access to reported case histories. He or she can enhance his or her repertoire of models through the published theories developed by others to account for their data and the data of many others. He or she can enhance his or her repertoire of advice through the published advisory strategies developed by others and evaluated by them against past case histories, theoretical foundations, and experience in use.

This model of individual expertise as intrinsically a knowledge acquisition process at multiple levels intrinsically embedded in, part of, and supported by essential social processes, is consistent with Stich and Nesbitt's (1984) view of the social role of experts being managers of the *reflective equilibrium* of the inductive process of knowledge generation in society. It is also consistent with the emphasis in professional development books concerned with good practice on the importance of balancing experts' authoritarian roles as knowledge repositories and appliers with their entrepreneurial roles as knowledge acquirers (Schön 1983).

In terms of design considerations for a next generation of knowledge acquisition tools, what the extended model of expertise in Figure 2 makes clear is that we should be making our objective the support of the knowledge processes of the *community* in which expertise plays a role. However, this does not necessarily mean that we have to discard the methodologies, techniques and tools that have been developed for use with individual experts. With the rapid pace of innovation in information technology, and the need for effective knowledge-based system development *now*, it is more realistic to design group support tools around the *integration* of existing techniques. In the longer term, the experience with such integration may lead to new designs that commence with a socially oriented paradigm, but we need much more practical experience before such designs can be proposed on a credible basis.

GROUP DECISION SUPPORT FOR KNOWLEDGE ACQUISITION

Figure 3 show an architecture for integrating existing knowledge acquisition techniques and tools to form a group decision support system for the knowledge processes of a community involved in knowledge-based system development. The roles shown in the lower part of the diagram are intended to be indicative rather than definitive. Users have to be supported in all these roles, and in hybrids and variants of them. Particular individuals may change between roles, either instantly as they operate the system or as part of changing capability and career migration, and hence it is important that the interface is as uniform as possible across roles.

The functionality shown in the boxes at the top of the diagram is also only illustrative, based on experience with existing individually oriented tools. It is reasonable to expect that needs for new functionality will become apparent as we gain experience of the group decision support approach. The center box already contains functionality such as email and conferencing that is not normally integrated with a knowledge acquisition system and yet is clearly part of the support of group processes. To what extent is it possible to use existing email and conferencing, and how much modification is required to achieve effective integration?

It is important to note that integration is itself a structured concept. System integration at the level of inter-application communication is easy to achieve. System integration at the level of appropriate mutual control of functionality and data exchange is also largely an implementation issue. However, knowledge level integration such as that necessary to support the combination and editing of knowledge structures in several different tools with different perspectives is very much more difficult.

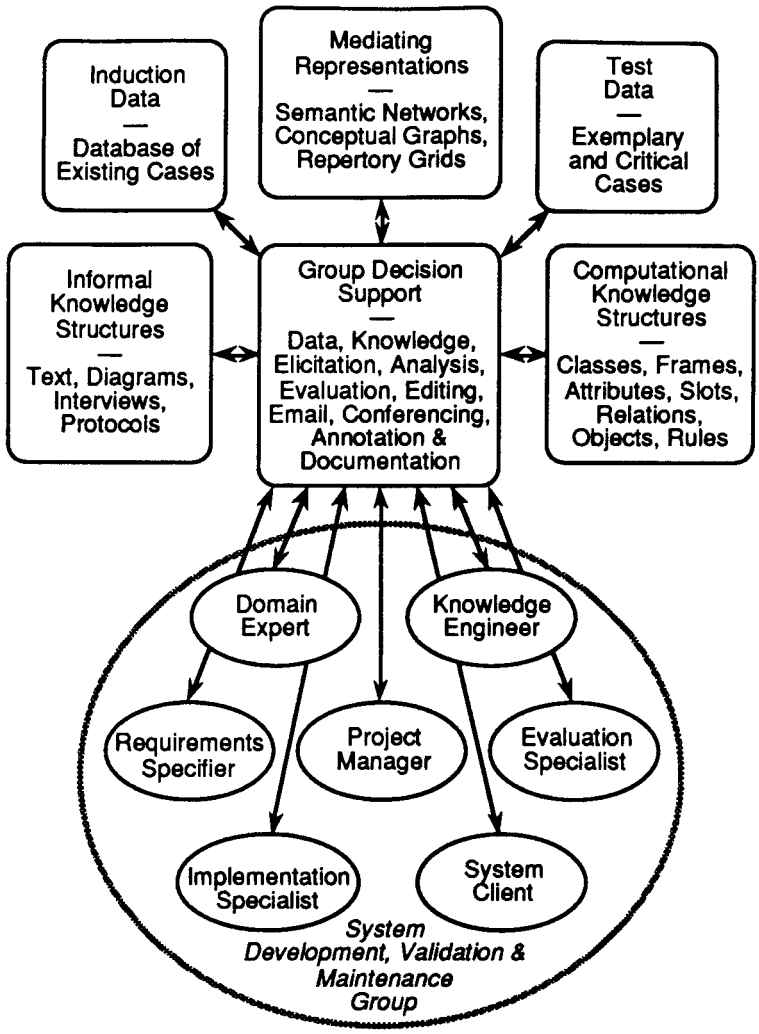

Figure 3 Group decision support system for knowledge-based systems development

Thus, Figure 3 presents an intermediate stage in system development, not an ultimate architecture. Over the next few years experience in the design, implementation and application of systems such as that shown will begin to generate the examples and principles necessary to the well-founded design of group decision support systems for the development of large scale knowledge based systems. Meanwhile, the considerations discussed above provide guidelines for research and development.

EXISTING TOOLS WITHIN THE GDSS CONTEXT

The architecture we have adopted for GDSS developments is the heterogeneous integration of existing tools, involving substantial redevelopment but not the design and implementation of a new monolithic system. This is an important issue in its own right and our logic for heterogeneous integration is manyfold (Gaines, 1990a); notably: the rapid pace of change of all the base technologies making systems obsolescent as they are implemented; the need to incorporate subsystems optimized for particular roles, probably by different organizations; and the overarching requirement for continuous upgrading and enhancement without a total system rebuild. All our current system development is based on highly modular systems communicating through servers and implemented as class libraries in object-oriented languages (Gaines, 1991b).

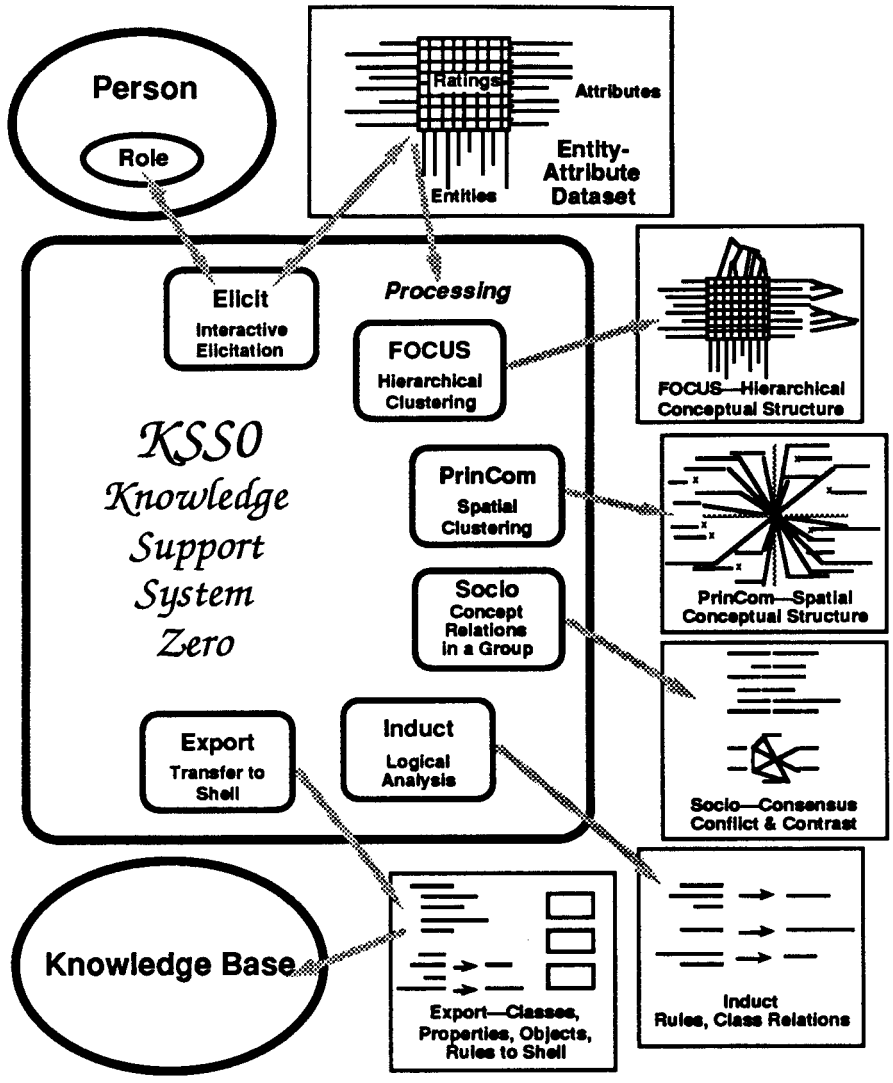

Figure 4 Architecture and facilities of KSS0, a knowledge acquisition workbench

Our original knowledge acquisition tool, KSS0 (Shaw & Gaines, 1987; Gaines, 1988) shown in Figure 4, providing repertory grid elicitation, conceptual clustering, conceptual comparison, empirical induction, and knowledge base creation tools, is already configured as a set of modules around a specialist database. KSS0 focuses on mediating representations supporting the expert's modeling processes in moving from skilled performance to overt knowledge structures supporting its emulation in the computer. In recent years we have extended KSS0 to support the informal representations of knowledge that are prior to those within the tool, such as text, pictures, diagrams, semi-structured interviews, protocols, and so on (Gaines, Rappaport & Shaw, 1989; Gaines & Linster, 1990). This has been done by providing an interapplication protocol allowing KSS0 to interact with Apple's HyperCard to provide the appearance of a seamless single application to users. KSS0-specific functionality in HyperCard is supported by scripts that allow conceptual structures on the KSS0 side to be linked to informal sources and annotation on the HyperCard side. However, the system is designed to provide an open architecture so that domain-specific tools may be developed in HyperCard and existing knowledge acquisition tools in HyperCard may be integrated, for example Woodward's (1990) Cognosys for the analysis of protocol data in textual form.

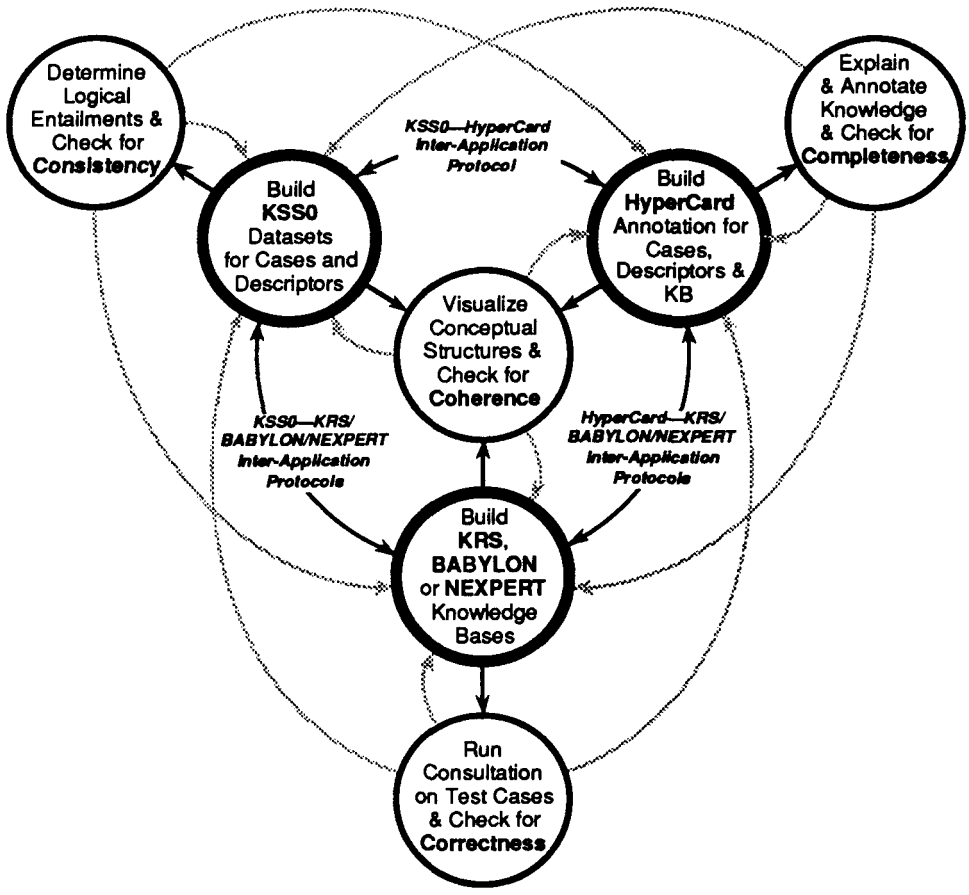

Figure 5 Heterogeneous integration of knowledge acquisition, hypermedia and knowledge-based system shells

KSS0 exports knowledge bases to a range of existing knowledge-based system shells, and we have reported a number of collaborative studies in which we have integrated these shells directly with KSS0 and HyperCard through inter-application protocols (Gaines, Rappaport & Shaw, 1989; Gaines & Linster, 1990). This leads to new modes of knowledge acquisition in which, for example, a test case that leads to an incorrect decision within the performance engine is posted directly back to KSS0 where the induction tool is automatically rerun to update the knowledge base taking into account the new test case.

The heterogeneous integration of knowledge acquisition, hypermedia and performance tools leads to a variety of options for validation and verification at each stage of the system development. Figure 5 illustrates the system architecture and classifies the forms of validation supported.

At the center of Figure 5, it is shown that each of the tools provides means for *visualizing* the conceptual structures they support. For example, KSS0 provides various clustering algorithms for presenting repertory grid data graphically, HyperCard is essentially visual, and KRS (Gaines, 1990b, 1991a), BABYLON (Christaller, di Primio & Voß, 1989) and NEXPERT (Rappaport, 1987) each provide their own knowledge-base graphers. From a formal validation point of view such visualization supports the expert and knowledge engineer in evaluating the *coherence* truth of the knowledge structures—their internal relations and consistency. Such evaluation provides feedback through each of the tools to improve expressiveness and correct errors that are made manifest through incoherence.

At the top left of Figure 5, the induction tool in KSS0 derives plausible constraints on the database of case data (essentially empirical rather than normative normalization of the database) in the form of entailments that eventually become expressed as rules. It supports the evaluation of these entailments for mutual *consistency* (can two rules arrive at different conclusions on the same case) and for consistency with the case data. Such evaluation again provides feedback through all the tools—for example, one may amend cases in KSS0, amend rules in KRS, or make a note of the potential problem in HyperCard.

At the bottom of Figure 5, the running of further cases, perhaps in routine system use, provides evaluation of the *correctness* of the knowledge base as a decision support tool. Again such evaluation provides feedback through all the tools, amendment of the KSS0 case base and re-induction, direct editing of the knowledge structures in KRS, or commentary in HyperCard.

Finally, at the top right of Figure 5, it is shown that the hypermedia tool provides a far more significant logical validation role that its annotation duties may suggest. Systems intrinsically cannot validate themselves for *completeness*, and clearly there can be no guarantees of completeness in an open universe. However, in terms of validating the formal knowledge base completeness, the informal knowledge base held in hypermedia form plays a very significant role. Anything mentioned informally that has no referent formally leads to a suspicion of incompleteness requiring further investigation.

INTEGRATING THE TOOLS WITHIN A GDSS INFRASTRUCTURE

Most of the technical functionality required for group decision support at the knowledge base level is already present in the existing systems of Figure 5. The tools each support various modes of user interaction for different user roles, and provide communication and data transformation protocols supporting mutual integration. What is missing, however, is explicit recognition within the existing technologies of the group as an entity, and technical support for the management of group processes.

In a different context we have developed a special-purpose 'groupware' system based on the integration of the repertory grid components of KSS0 (Shaw, 1980; Shaw & Gaines, 1983) with an electronic mail sub-system to facilitate the formation and management of 'special interest groups' (Shaw & Gaines, 1991a). The architecture of this system, *RepGrid-Net*, is shown in Figure 6. RepGrid-Net uses the Socio analysis tools within KSS0 to analyze and display graphically the comparisons between individual conceptual structures, and socionets of a complete group showing the extent to which one individual has the concepts to understand another.

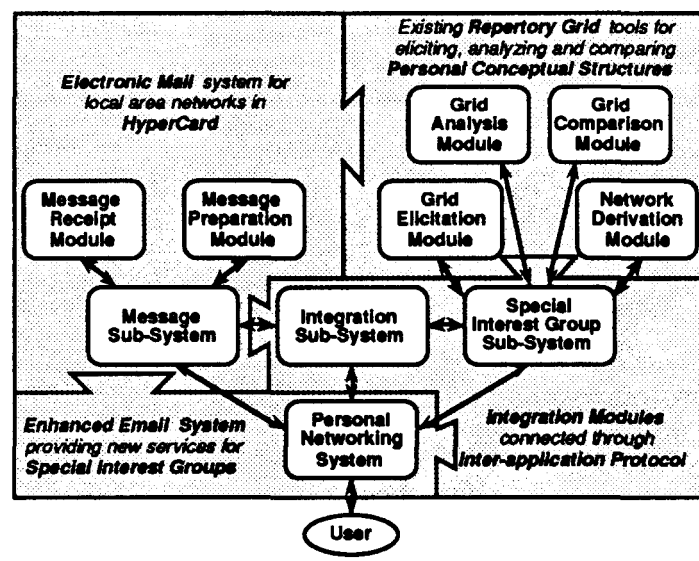

Figure 6 RepGrid-Net integration of repertory grid elicitation and analysis with electronic mail to facilitate the formation and management of special-interest groups

The development and application of RepGrid-Net showed the major system impact that the addition of facilities for electronic mail may have on an interactive system. Activities such as annotation and documentation are notoriously difficult to encourage in a development team. Part of the problem is that the underlying human process is one of discourse—what is being produced is best regarded as messages to other people not as an abstract text database. Our preliminary studies suggest that users may more readily add such information if there is a specific audience as there is in the human social situation.

A second parallel development of a technology that, at first sight, appears to have little to do with knowledge acquisition has been our development of *GroupWriter*, a word processor to support collaborative authoring of scientific papers (Malcolm & Gaines, 1991). GroupWriter appears to its users very much like Microsoft Word but gives access to alternative versions of a document, and text and speech annotation, through a narrow side bar in which paragraphs with alternates or annotation are bracketed. Moving the cursor into this side bar gives access to popup menus allowing alternates to be shown and merged, and annotation to be shown or played.

Since GroupWriter provides a hypertext-style linkage structure to document components it is readily extended to provide hypertext links to modules within a knowledge acquisition system. For example, source documents and transcripts of interviews and protocols may be linked to the knowledge structures that have been developed based on them. This can be done without undermining the visual appearance of the original document. Documents generated as part of the knowledge engineering process can also be treated in this way—they are normal documents with full typographic formatting, embedded pictures, and so on, but they are also tightly embedded in the knowledge structures being developed.

Most of RepGrid-Net is written in HyperCard which has been the basis of our integration of KSS0 with NEXPERT and BABYLON as shown in Figure 5, and GroupWriter is written as modules in the same class library in C++ as our own knowledge-based system shell, KRS. Hence, it has been technically simple to link them both into our existing knowledge acquisition tools. Figure 7 shows the current state of development of KSSn, the general knowledge support system that is being developed as a successor to KSS0.

At the center is the group decision support system seen by the user community, provided by integrating the sub-systems in the outer ring through an inter-application communication protocol. Going around the diagram clockwise from the bottom left:

- The current *electronic mail* system is the HyperCard/AppleTalk system developed for RepGrid-Net. This has been useful for laboratory trials and demonstrations but, for significant applications, is being extended to interface to the full Internet mail system.
- The *group writing* system has been discussed above. This is also being extended currently to operate across Internet, and facilities are being added to import and export documents from other text processing systems supporting RTF or SGML.
- The current *hypermedia* system is the HyperCard/SuperCard system developed for informal knowledge base development and knowledge base annotation as part of the integration with NEXPERT and BABYLON. The script languages of the hypertext systems give this module far greater significance than is apparent in the figure. It is the most open architecture and customizable component of the system and, hence, is the natural vehicle for further integration with other systems.
- The *text analysis* tools are currently the Texan lexical and clustering modules from KSS0. They have been rewritten in C++ to export text clusters as knowledge structures in the visual language of KRS. The current text analysis is useful as a vocabulary acquisition tool for the other elicitation techniques. In the longer term the text analysis system needs to be extended with the more powerful parsing and semantic analysis techniques now being developed for knowledge bases (Gomez & Segami, 1990).

- The *knowledge elicitation* tools consist of the modules of KSS0 shown in Figure 4. The strength of this system is the ease of use of its graphic elicitation and clustering tools by end users such as domain experts. Its weakness is the limited representational power of the entity-attribute database structure of basic repertory grids. These were overcome in AQUINAS through extensions to hierarchical structures, and we have adopted a KL-ONE-like knowledge representation schema in KRS to provide similar extensions for KSSn. However, developing user interfaces which maintain the usability of KSS0 is problematic with the new data structures and, whilst these are being developed and evaluated, the original KSS0 system remains an important resource.
- The *empirical induction* tools are based on Induct (Gaines, 1989b), a system extending Cendrowska's (1987) Prism algorithm to develop rules directly from noisy datasets.
- The *visual language* tools are based on an interactive graphical structure edit for concept maps and semantic nets (Gaines, 1991c). They support the automatic layout of existing KL-ONE-like knowledge structures, and the interactive entry and editing of such knowledge structures. They compile them to the intermediate languages of a number of knowledge-based system shells.
- The *knowledge-based system* shells currently supported are KRS, NEXPERT and BABYLON. These are sufficiently diverse to force us to maintain an open architecture at this interface which should be extensible to other knowledge-based system shells.

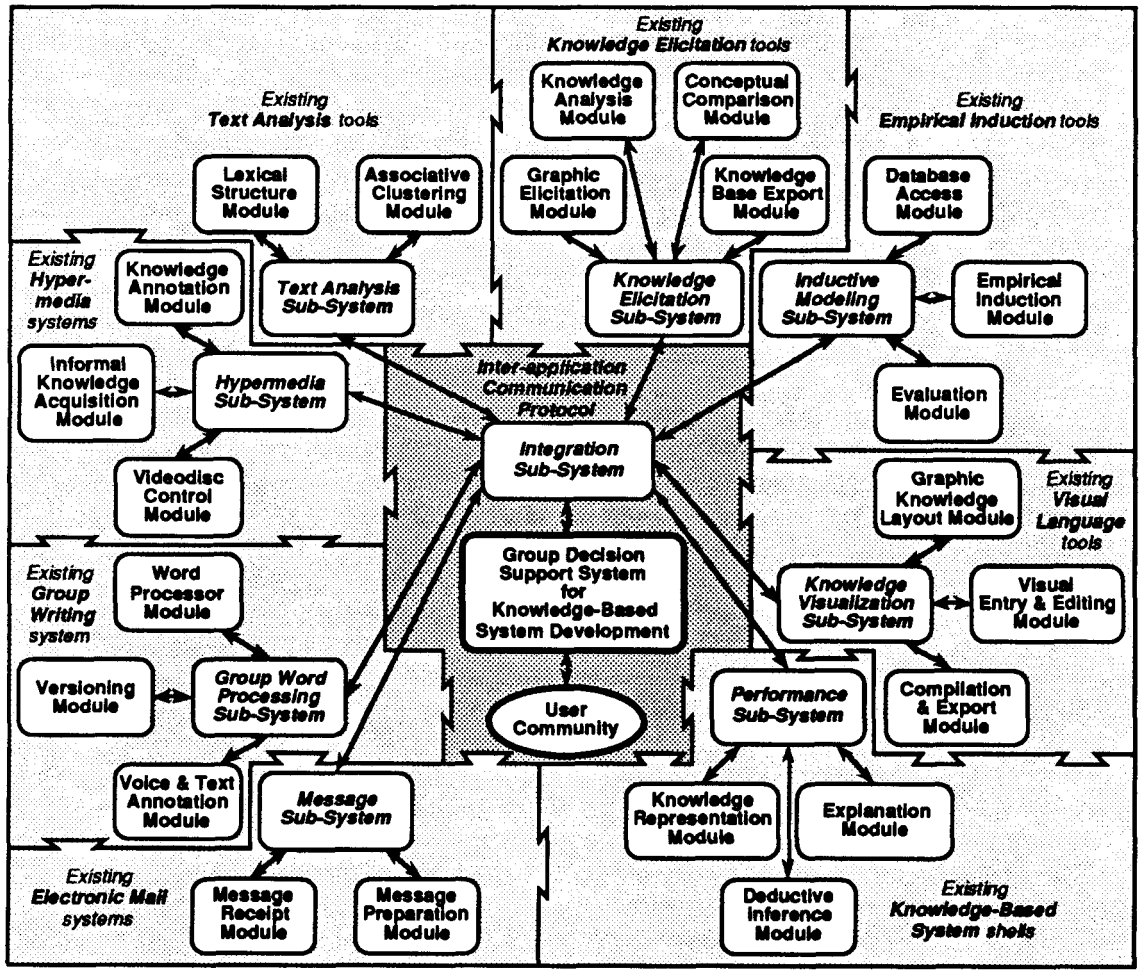

Figure 7 Group decision support system for knowledge-based systems development

All of the components of the system of Figure 7 are now in operation in a variety of different configurations. It is not clear whether the system as shown will ever, or need ever, exist as an ostensible monolithic piece of software or a particular workstation and network configuration. This is an important feature of the architecture, that it is open and reconfigurable for particular studies, developments and applications. In particular, it is designed such that tools from other research groups and commercial vendors may be incorporated with the minimum of effort.

The complexity and diversity of technologies involved in Figure 7 makes clear the need, in research at least, for the heterogeneous integration approach discussed earlier. The various sub-systems have to be developed as modular, reliable and usable systems in their own right. For example, the electronic mail and group writing sub-systems need to be proven as viable mailing and writing systems in routine use before they are incorporated in the overall system. Little can be learnt from studies of integrated systems if critical sub-systems are not effective when used alone. The complexity can also become a major impediment to development if the effort of modifying the system becomes too great. It is important to be able to develop different types of system with such little effort that there is no resistance to discarding them when they have served their experimental function.

It is also worth noting the sub-systems that are *not* included in Figure 7:

- *Conferencing* has been omitted because the problem of managing real-time interaction between participants in a conferencing situation is still a major research problem. We believe that much can be achieved through non-real-time interaction through email, documents and knowledge structures, but conferencing is an obvious technology to eventually include in experiments with knowledge acquisition in a group decision support framework.
- *Tutoring/training* has been omitted as a specific module because we see the knowledge support system of Figure 7 as providing an adequate framework for investigating applications to education without any specific tutoring tools. Some preliminary experiments on the HyperCard/KSSO/KRS combination in supporting graduate student's creativity have been very encouraging and suggest that knowledge acquisition toolkits have important educational roles (Shaw & Gaines, 1991b).
- *Analogical* and *case-based* reasoning modules have been omitted because we have not had time to develop significant sub-systems yet. However, the direct comparison of knowledge structures, abstraction from them, and inference using case data are clearly important extensions to the reasoning mechanisms offered by the shells already included.

Others will find additional components missing. This seems inevitable. It seems unlikely that any knowledge support system will ever be complete. One can only attempt to provide an extensible architecture of sufficient generality to allow enhancement, and even the architecture must eventually become obsolete as our knowledge and requirements develop.

CONCLUSIONS

This paper has reported ongoing research on group decision system support architectures for knowledge based system development. It has been argued that knowledge engineering for any realistic system is essentially a group process involving multiple requirements specifiers, knowledge engineers, domain experts, system implementation specialists, evaluation specialists and knowledgeable clients. This paper has reported the extension of our existing knowledge support tools to provide online support of the group decision processes in both its social and technical aspects. In particular we have integrated electronic mail and conferencing facilities with the existing knowledge acquisition tools, hypermedia systems and knowledge-based systems shells.

ACKNOWLEDGEMENTS

This work was funded in part by the Natural Sciences and Engineering Research Council of Canada. We are grateful to many colleagues who have played significant roles in exchanging ideas and research data with us that have been very important to the developments reported here, notably, John Boose, Jeff Bradshaw, Bill Clancey, Paul Compton, Bob Jansen, Marc Linster, Ross Quinlan, Alain Rappaport, and Brian Woodward, plus many stimulating and influential discussions at the Knowledge Acquisition Workshops worldwide.

REFERENCES

Boose, J H. (1989). A survey of knowledge acquisition techniques and tools. **Knowledge Acquisition 1** (1), 39-58 (March).

Boose, J.H. & Bradshaw, J.M. (1987). Expertise transfer and complex problems: Using AQUINAS as a knowledge acquisition workbench for knowledge-based systems. **International Journal of ManMachine Studies, 26,** 328.

Boose, J.H. & Gaines, B.R., Eds. (1988). **Knowledge Acquisition Tools for Expert Systems**. London, Academic Press.

Boose, J.H. & Gaines, B.R., Eds. (1990). **The Foundations of Knowledge Acquisition**. London, Academic Press.

Cendrowska, J. (1987) An algorithm for inducing modular rules. **International Journal of Man-Machine Studies 27** (4), 349-370 (October).

Christaller, T., di Primio, F. and Voß, A. (1989) **Die KI-Werkbank BABYLON**. Bonn: Addison Wesley.

Feigenbaum, E.A. (1980). Knowledge Engineering: the Applied Side of Artificial Intelligence. **Report STAN-CS-80-812**. Department of Computer Science, Stanford University.

Gaines, B.R. (1988). Knowledge acquisition systems for rapid prototyping of expert systems. **INFOR, 26**(4), 256-285 (November).

Gaines, B.R. (1989a) Social and cognitive processes in knowledge acquisition. **Knowledge Acquisition 1**(1), 251-280 (March).

Gaines, B.R. (1989b) An Ounce of Knowledge is Worth a Ton of Data: Quantitative Studies of the Trade-Off between Expertise and Data based on Statistically Well-Founded Empirical Induction. **Proceedings of 6th International Workshop on Machine Learning**, pp.156-159. San Mateo, California: Morgan Kaufmann (June).

Gaines, B.R. (1990a). Organizational integration: modeling technology and management. Zunde, P. & Hocking, D. (Eds.) **Empirical Foundations of Information and Software Sciences II.** pp. 51-64. New York: Plenum Press.

Gaines, B.R. (1990b) Knowledge representation servers: a generic technology for knowledge acquisition and knowledge-based systems. **Proceedings of the First Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop: JKAW'90.** pp. 413-430. Tokyo: Ohmsha (October).

Gaines, B.R. (1990c). Knowledge support systems. **Knowledge-Based Systems** 3(4) 192-203 (December).

Gaines, B.R. (1991a). Empirical investigation of knowledge representation servers: design issues and applications experience with KRS. **AAAI Spring Symposium: Implemented Knowledge Representation and Reasoning Systems**. pp. 87-101. Stanford (March).

Gaines, B.R. (1991b). Object-oriented implementation of object-oriented knowledge representation. **Proceedings of the Third AAAI Workshop on Object-Oriented Programming in AI**. Anaheim, Califonia (July).

Gaines, B.R. (1991c). An interactive visual language for term subsumption visual languages. **IJCAI'91: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence**. San Mateo, California: Morgan Kaufmann.

Gaines, B.R. & Boose, J.H., Eds. (1988). **Knowledge Acquisition for Knowledge-Based Systems**. London: Academic Press.

Gaines, B.R. & Linster, M. (1990). Development of second generation knowledge acquisition systems. Wielinga, B., Boose, J.H. & Gaines, B.R., Schreiber, G., van Someren, M., Eds. **Current Trends in Knowledge Acquisition.** pp. 143-160. Amsterdam, IOS.

Gaines, B.R., Rappaport, A. & Shaw, M.L.G. (1989). A heterogeneous knowledge support system. Boose, J.H. & Gaines, B.R., Eds. **Proceedings of the Fourth AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop.** pp.13-1-13-20. Banff (October).

Gaines, B.R. & Shaw, M.L.G. (1989). Comparing the conceptual systems of experts. **Proceedings of the Eleventh International Joint Conference on Artificial Intelligence**. pp. 633-638. San Mateo, California: Morgan Kaufmann (August).

Gomez, F. & Segami, C. (1990). Knowledge acquisition from natural language for expert systems based on classification problem-solving methods. **Knowledge Acquisition, 2**(2), 107-128 (June).

Hawkins, D. (1983). An analysis of expert thinking. **International Journal of Man-Machine Studies, 18**(1), 1-47 (January).

Liou, Y.I., Weber, E.S. & Nunamaker, J.F. (1990). A methodology for knowledge acquisition in a group decision support environment. **Knowledge Acquisition 2** (2), 129-144 (June).

Malcolm, N. & Gaines, B.R. (1991). A word processor supporting group-writing. **Technical Report No.91/420/04**. Department of Computer Science, University of Calgary (February).

Rappaport, A. (1987) Multiple-problem subspaces in the knowledge design process. **International Journal of Man-Machine Studies 26**(4), 435-452 (April).

Schön, D.A. (1983). **The Reflective Practitioner**. New York: Basic Books.

Shaw, M.L.G. (1980). **On Becoming A Personal Scientist: Interactive Computer Elicitation of Personal Models Of The World**. London: Academic Press.

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.

Shaw, M.L.G. & Gaines, B.R. (1987). KITTEN: Knowledge Initiation & Transfer Tools for Experts & Novices. **International Journal of Man-Machine Studies, 27**, 251-280.

Shaw, M.L.G. & Gaines, B.R. (1989). A methodology for recognizing conflict, correspondence, consensus and contrast in a knowledge acquisition system. **Knowledge Acquisition 1**(4), 341-363 (December).

Shaw, M.L.G. & Gaines, B.R. (1991a). Supporting personal networking through computer networking **Proceedings of CHI'91: Human Factors in Computing Systems**. New York: ACM Publications.

Shaw, M.L.G. & Gaines, B.R. (1991b). Mapping creativity with knowledge support tools. **Proceedings of the AAAI Workshop on Creativity: Models, Methods and Tools.** Anaheim, California (July).

Stich, S.P. & Nisbett, R.E. (1984). Expertise, justification and the psychology of inductive reasoning. Haskell, T.L., Ed. **The Authority of Experts**. pp. 226-241. Bloomington, Indiana: Indiana University Press.

Woodward, B. (1990) Knowledge engineering at the front-end: defining the domain. **Knowledge Acquisition, 2**(1) 73-94 (March).