GROUP DECISION SUPPORT FOR THE VERIFICATION AND VALIDATION PROCESS

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 any realistic system is essentially a group process involving multiple requirements specifiers, knowledge engineers, domain experts, system implementation specialists, evaluation specialists and knowledgeable clients. Validation and verification is not a distinct and separate process but rather an integral part of the socio-technical interactions between all these differing participants. Much attention has been paid to the development of computer-based interactive tools to support individual technical activities in knowledge-based system development, for knowledge acquisition, expertise transfer, validation, and so on. However, 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 our 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. In particular, our design philosophy has been that validation and verification issues, while being specifically managed by those with V&V accountability, should also be open to question and investigation by all group members at any stage of the development process. This entails automatic support of exploratory testing at all levels of the system, with a high level of existing knowledge engineering functionality, yet an openness enabling issues to be investigated which were not part of the original support system conception.

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; 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 on individuals, tools to interview the expert, tools to support the knowledge engineer, and has neglected the essential social aspects of large-scale system development. As we become more and 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 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 all blur into a single, 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 KSS0 (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 support 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, 1989), 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 our existing tools within a networked environment designed to support the group as a whole as well as individuals playing roles within it.

EXISTING TOOLS WITHIN THE GDSS CONTEXT

The architecture we have adopted for our GDSS developments is the heterogeneous integration of existing tools, involving substantial redevelopment but not the design and implementation of a 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).

Our original knowledge acquisition tool, KSS0 shown in Figure 1, providing repertory grid, 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. 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.

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.

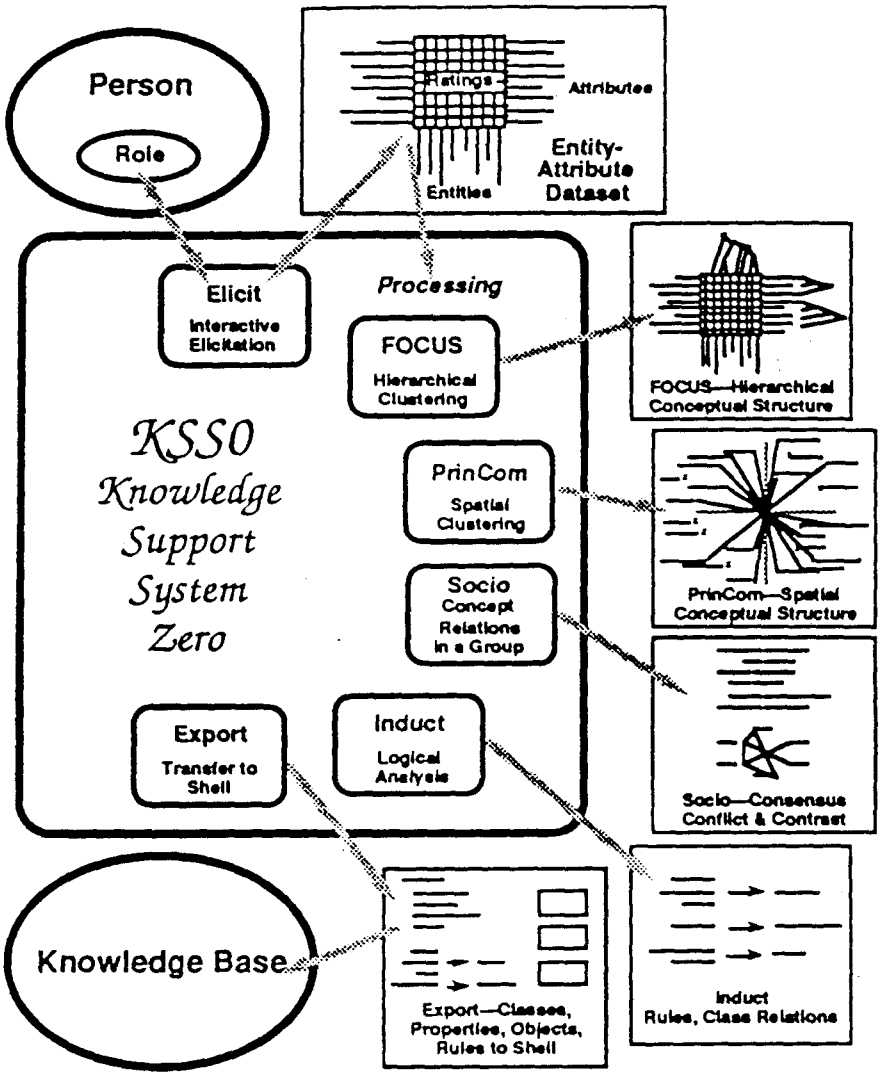

Figure 1 Architecture and facilities of KSS0, a knowledge acquisition workbench

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 2 illustrates the system architecture and classifies the forms of validation supported.

At the center of Figure 2, 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 correct errors, improve expressiveness, and so on, that are made manifest through incoherence.

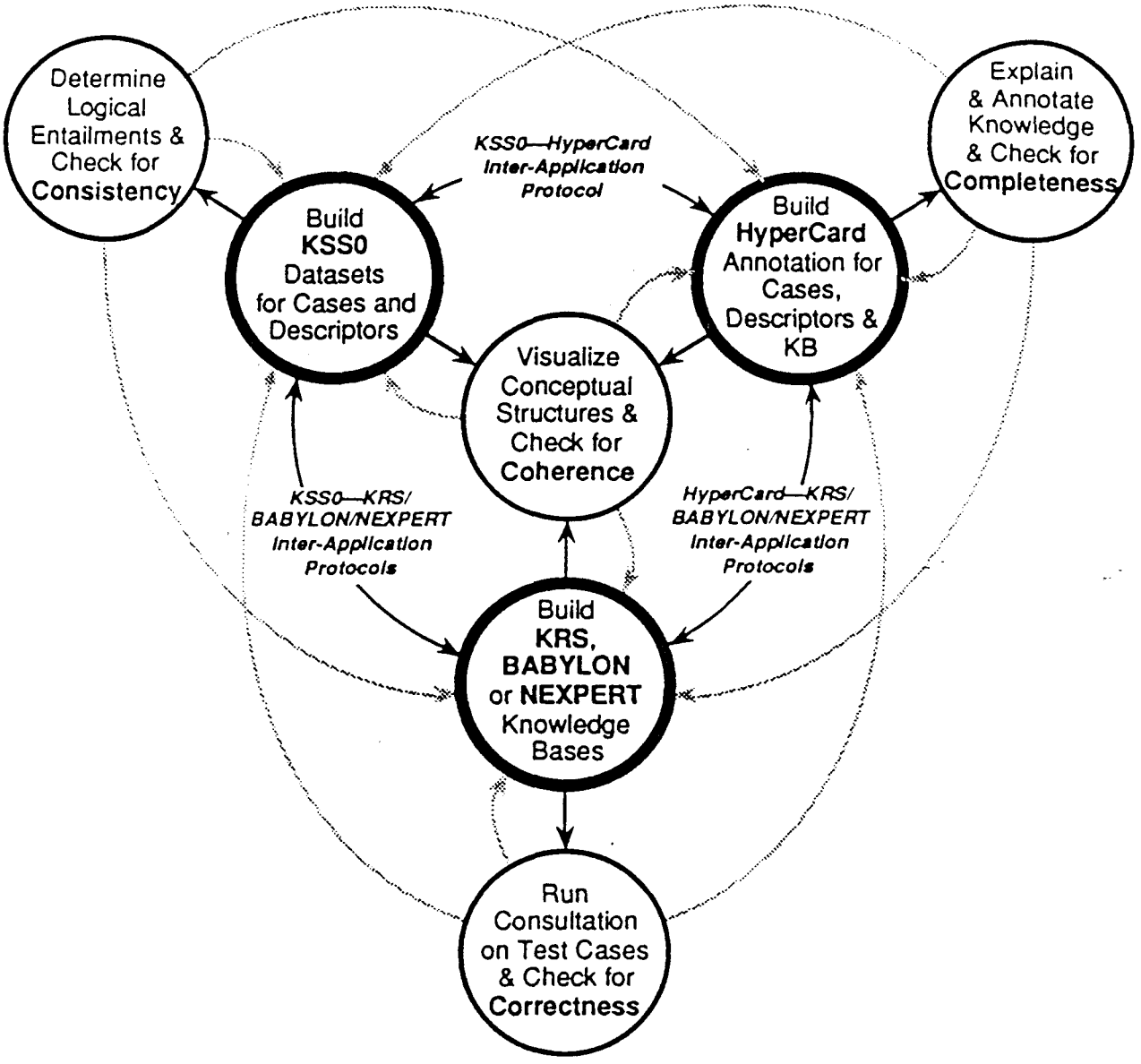

Figure 2 Heterogeneous integration of knowledge acquisition, hypermedia and knowledge-based system shells

At the top left of Figure 2, 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 2, 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 2, 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 2. 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 completely different context we have already developed a 'groupware' system based on the integration of the repertory grid components of KSSO (Shaw, 1980; Shaw & Gaines, 1983) with an electronic mail sub-system to facilitation the formation and management of 'special interest groups' (Shaw & Gaines, 1991). The architecture of this system, *RepGrid-Net*, is shown in Figure 3. RepGrid-Net utilizes the Socio analysis tools within KSSO to analyze and display graphically comparisons between individual conceptual structures, and a socionet of a complete group showing the extent to which one individual has the concepts to understand another.

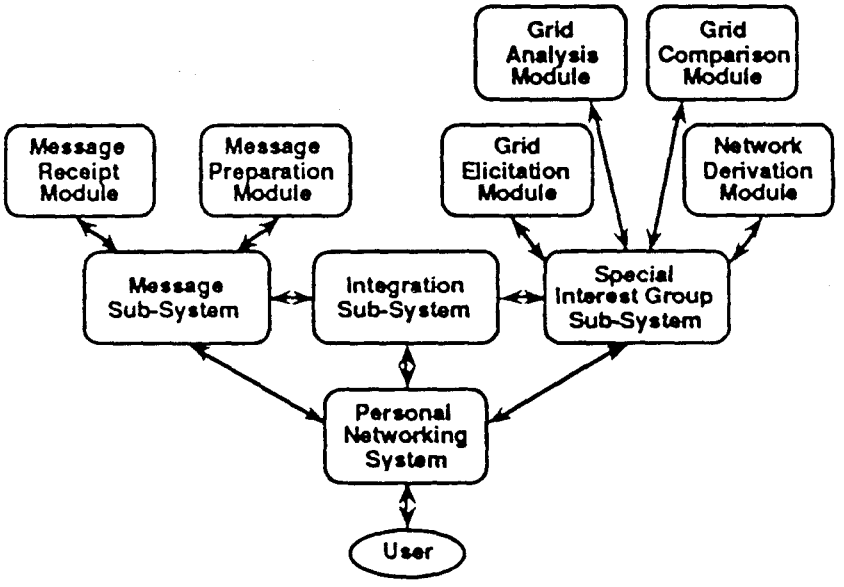

Figure 3 RepGrid-Net integration of repertory grid elicitation and analysis with electronic mail to facilitate the formation and management of special-interest groups

The development and application of RepGrid-Net brought to our attention 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 far more readily add such information if there is a specific target audience as there is in the human social situation.

RepGrid-Net integrates the mail subsystem with the repertory grid tools in a seamless fashion and provides the model for our group decision support architecture for knowledge-based system development as shown in Figure 4. We have identified a number of roles within the system development group, each of which requires some specialist support facilities. However, the system interface is uniform over the whole group—the same person may play different roles at different times and individuals commonly change roles during the course of development, and it is important that they see the minimum additional functionality, uniformly consistent with that which they have already, rather than discontinuous jumps to different styles of interface and operation.

The central GDS module acts as switchboard routing users to appropriate functionality, and also provides new functionality of its own, including electronic mail, conferencing, annotation extended to all system databases and functionality rather than just knowledge-base items, and a documentation facility supporting typographic text, diagrams and hypertext-style linkage. We have used commercially available tools for much of this additional functionality allowing us to focus development effort on the integration issues.

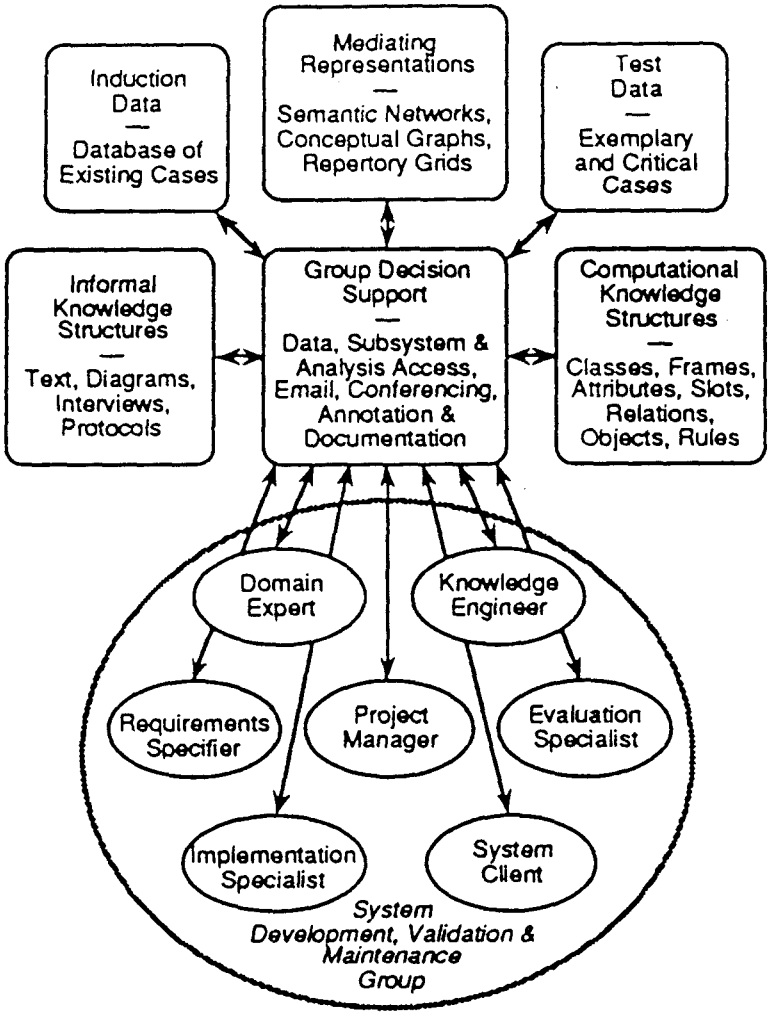

Figure 4 Group decision support system for knowledge-based systems development

All of the components of the system of Figure 4 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/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 resea groups and commercial vendors may be incorporated with the minimum of effort. This already proven effective in our work with Neuron Data to incorporate NEXPERT, with the GM in Germany to incorporate BABYLON, and current work on the integration of DataPak's wo processing engine and Faircom's SQL database server.

It is also apparent that there is no box in Figure 4 specifically labelled 'validation a verification'. We have adopted the viewpoint from the start of our work on knowled acquisition tools that V&V issues occur at every stage of the development process and requ specific support at every stage in every tool. We have begun to adopt the same approach maintenance issues, for example, that no tool should be designed for 'one-off' use but sho always be seen as playing a continuing role in the iterative cycle of develop. This has a sev technical impact in requiring support for the integration of reprocessed data with previou processed data that may have been substantially edited. It is not any easy problem and needs be tackled as part of tool design—it is virtually impossible to provide as a later enhanceme We have also incorporated in later tools, such as KRS, knowledge representation schema such Compton and Jansen's (1990) 'ripple-down' rule organization that are designed to substantia reduce problems of validation and maintenance. This seems an important direction to follow— design out problems rather than develop florid support systems to 'fix' them unnecessari However, it is not clearly how widely and uniformly this approach can be adopted. There see to be an essential anarchy in human affairs—particularly where knowledge is concerned!

CONCLUSIONS

This paper has reported ongoing research on group decision system support architectures knowledge based system development. It has been argued that knowledge engineering for a realistic system is essentially a group process involving multiple requirements specifie knowledge engineers, domain experts, system implementation specialists, evaluation speciali and knowledgeable clients. It has been argued that validation and verification is not a disti and separate process but rather an integral part of the socio-technical interactions between these differing participants. This paper has reported the extension of our existing knowled support tools, which already provide offline analysis of group-related issues such as conflicts concepts and terminology, to provide online support of the group decision processes in both social and technical aspects. In particular we have integrated electronic mail and conferenc facilities with the knowledge acquisition tools, hypermedia systems and knowledge-ba systems shells that currently comprise our knowledge support system.

ACKNOWLEDGEMENTS

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

REFERENCES

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

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

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

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

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. This has already proven effective in our work with Neuron Data to incorporate NEXPERT, with the GMD in Germany to incorporate BABYLON, and current work on the integration of DataPak's word processing engine and Faircom's SQL database server.

It is also apparent that there is no box in Figure 4 specifically labelled 'validation and verification'. We have adopted the viewpoint from the start of our work on knowledge acquisition tools that V&V issues occur at every stage of the development process and require specific support at every stage in every tool. We have begun to adopt the same approach to maintenance issues, for example, that no tool should be designed for 'one-off' use but should always be seen as playing a continuing role in the iterative cycle of develop. This has a severe technical impact in requiring support for the integration of reprocessed data with previously processed data that may have been substantially edited. It is not any easy problem and needs to be tackled as part of tool design—it is virtually impossible to provide as a later enhancement. We have also incorporated in later tools, such as KRS, knowledge representation schema such as Compton and Jansen's (1990) 'ripple-down' rule organization that are designed to substantially reduce problems of validation and maintenance. This seems an important direction to follow—to design out problems rather than develop florid support systems to 'fix' them unnecessarily. However, it is not clearly how widely and uniformly this approach can be adopted. There seems to be an essential anarchy in human affairs—particularly where knowledge is concerned!

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. It has been argued that validation and verification is not a distinct and separate process but rather an integral part of the socio-technical interactions between all these differing participants. This paper has reported the extension of our 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. In particular we have integrated electronic mail and conferencing facilities with the knowledge acquisition tools, hypermedia systems and knowledge-based systems shells that currently comprise our knowledge support system.

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 Man-Machine 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.

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

Compton, P. & Jansen, R. (1990b) A philosophical basis for knowledge acquisition. **Knowledge Acquisition 2**(3).241-258.

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

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. & 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. (1980). New directions in the analysis and interactive elicitation of personal construct systems. **International Journal of Man-Machine Studies, 13**, 81-116.

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, Califomia: Morgan Kaufmann (August).

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).

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

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. (1991). Supporting personal networking through computer networking **Proceedings of CHI'91: Human Factors in Computing Systems.** New York: ACM Publications.

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