PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1992,
edited by ICOT. © ICOT, 1992

An Integrated Knowledge Support System

B R Gaines, M Linster† and M L G Shaw

Knowledge Science Institute, University of Calgary
Calgary, Alberta, Canada T2N 1N4
†Expert Systems Research Group, GMD
PO Box 1240, D-5205 St. Augustin 1, Germany

Abstract

The overall aim of the studies reported is to explore the possibility of creating highly integrated *knowledge support systems* through the loose coupling of software tools developed independently at unrelated sites for different purposes. Three tools were integrated to form a combined knowledge acquisition and performance system. A hypermedia tool was used as a general-purpose knowledge acquisition tool for unstructured material in the form of text and diagrams. A knowledge acquisition tool was used to elicit knowledge more formally and structure it as a computable knowledge base. A knowledge representation and deduction tool was used to represent the elicited knowledge and perform inferences with it to generate advice in a performance situation. Strong synergy was created between the tools in that the annotation and explanation captured in the hypermedium system were available as context-sensitive help to the user of the expert system, and the expert system was used to validate the knowledge base generated by the knowledge acquisition tool and feed back anomalous cases as additional data for induction.

1 Introduction

This paper reports the results of a collaborative program of research between the Canadian Knowledge Science Institute (KSI) and the German National Research Centre for Computer Science (GMD) on the integration of knowledge acquisition and performance systems. Researchers at the GMD have developed a knowledge-based systems shell, BABYLON, which combines object-oriented knowledge representation with a number of powerful inference paradigms (Christaller, di Primio & Voss, 1989). Researchers at the KSI have developed a knowledge acquisition system, KSS0, which combines object-oriented knowledge representation with a number of powerful elicitation, visualization and induction paradigms (Shaw & Gaines, 1987). The Canadian researchers have also developed inter-program communication protocols enabling Apple's HyperCard to be used as an extension to KSS0 for the textual and graphic annotation of knowledge structures (Gaines, Rappaport & Shaw, 1989). The outcome of the collaborative program is a system, Hyper-KSE, combining hypermedia, knowledge acquisition and performance tools, to provide an environment supporting knowledge-based system development from acquisition, through application to maintenance.

The approach we have adopted to system implementation 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, 1991).

2 The Principles Involved

When combining tools in an integrated system it is important to have a clear systems architecture in terms of required functionality and how it is to be allocated among the tools. It is necessary to exclude some facilities provided by some tools because it is duplicated in others or inappropriate in the overall system. If this is not done, and the combined system does not project a clear model of its intended use, then users can become confused by excessive features. A functional model for the integrated system has been built at a high level of abstraction in terms of four dimensions of logical validation of a knowledge base, corresponding to similar dimensions in evaluating the truth of scientific theories (Rescher, 1979):-

- Coherence—the coherence of internal relationships between knowledge structures
- Consistency—the lack of logical contradiction between knowledge structures
- Correctness—the correctness of deductions from the knowledge structures checked against external data
- Completeness—the adequate coverage of an intended scope for deductions from the knowledge structures.

Figure 1 shows how the three tools relate to the four dimensions of validation.

At the center of Figure 1, each of the tools provides means for *visualizing* the conceptual structures they support. For example, the knowledge acquisition tool provides various clustering algorithms for presenting case data graphically, the hypermedia tool is essentially visual, and the representation system provides its own knowledge-base grapher. From a formal validation point of view such visualization supports the expert and knowledge engineer in evaluating the *coherence* truth of the knowledge structures—the internal relations between different representations. Such evaluation provides feedback through each of the tools to correct errors, improve expressiveness, and so on, that are made manifest through incoherence.

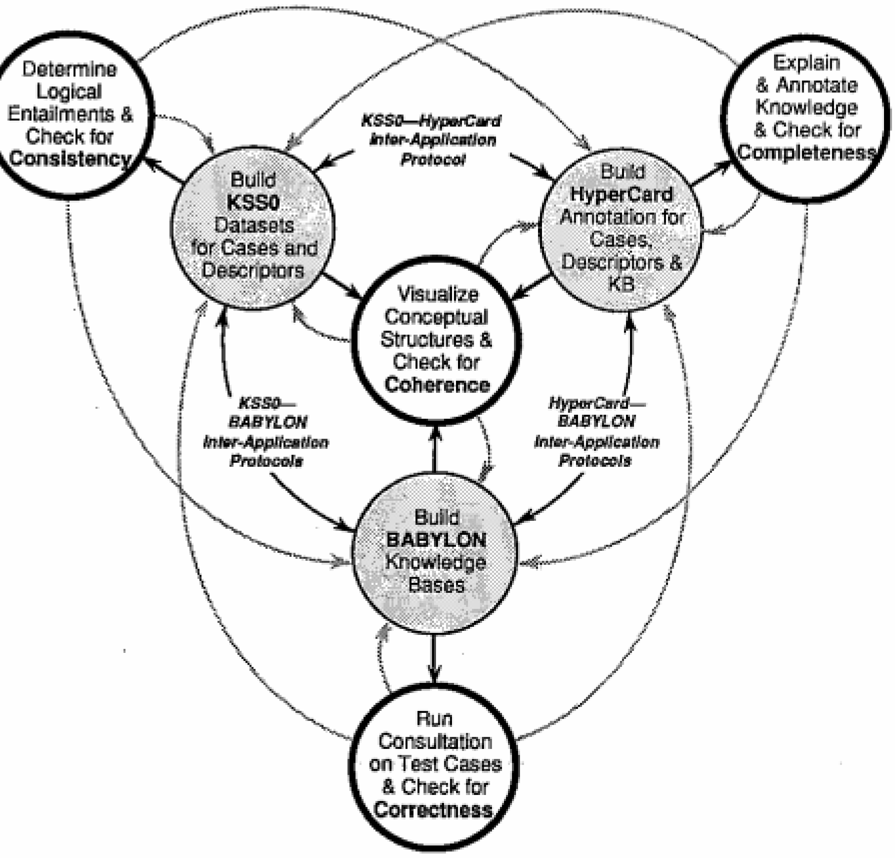

Fig. 1 Logical basis for modes of knowledge validation in the integrated system

At the top left of Figure 1, the induction module in the knowledge acquisition tool derives plausible constraints on the database of case data 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 the knowledge acquisition tool, amend rules in the representation tool, or make a note of the potential problem in the hypermedia tool.

At the bottom of Figure 1, the running of new test 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 case base and re-induction, direct editing of the knowledge structures, or commentary in hypertext.

Finally, at the top right of Figure 1, 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.

3 The Integrated System Architecture and Operation

The knowledge acquisition tool KSS0, 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. It 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 it has been extended 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 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.

KSS0 exports knowledge bases to a range of existing knowledge-based system shells, and a number of collaborative studies have been reported in which these shells have been integrated directly (Gaines, Rappaport & Shaw, 1989; Gaines & Linster, 1990). Figure 2 shows the distributed knowledge base and inter-application protocols linking the hypermedia, knowledge acquisition and knowledge-based system shell in Hyper-KSE.

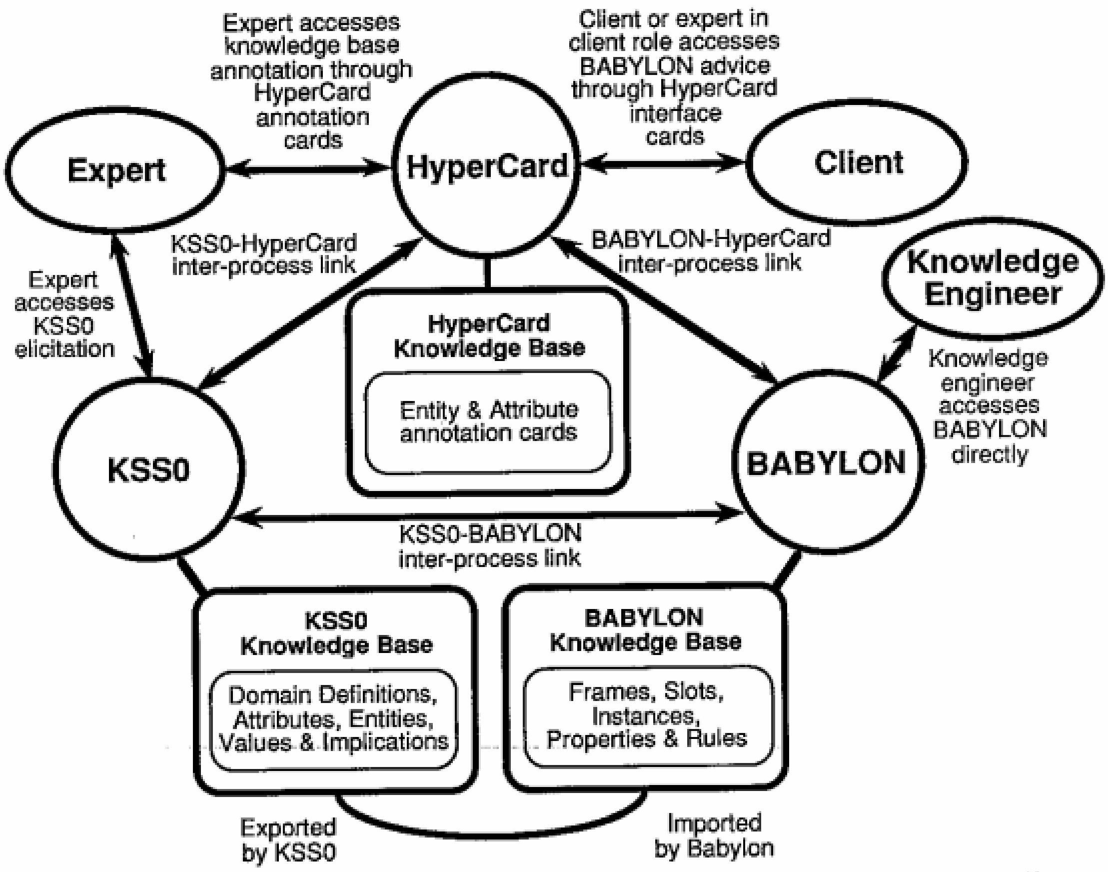

Fig. 2 Integration of hypermedia, knowledge acquisition and knowledge-based system shell

Figure 3 shows a typical sequence of activity in using the integrated system to develop a knowledge base. The acquisition commences with parallel creation by the knowledge engineer of a repertory grid in KSS0 and annotation stack in HyperCard. The elicitation tools are then used by the expert to enter critical cases characterizing the domain together with their relevant distinctive attributes and annotation about both cases and attributes. In particular, this annotation can include descriptions of the procedures by which the expert characterizes a case in terms of the attributes, and tutorial examples of such characterization. At any time during this process the expert or knowledge engineer can analyze the knowledge entered to date, visually clustering it, deriving underlying implications, and extending the knowledge structures and annotation in the light of such feedback.

When the expert feels that a reasonable amount of knowledge has been entered it is exported to BABYLON where it can be consulted on test cases, including access to the annotation in HyperCard. If a test case leads to an error then this case can be posted back to KSS0 with a corrected recommendation, re-analyzed and a revised knowledge base exported to BABYLON.

The following sequence of screen dumps is based on a simple tutorial example on mushroom toxicity provided for training with Hyper-KSE. Figure 4 shows the entity screen in KSS0 where a list of mushrooms has been entered, KSS0 is automatically evaluating entity and attribute matches and suggesting further elicitation, and the user has popped up a menu giving access to commands, annotation in HyperCard and consultation in BABYLON.

Figure 5 shows the way in which matching attributes are shown graphically in KSS0. The expert can change a rating just by dragging an entity along the scale. He or she may also respond to the prompt at the top to enter another, discriminating case. This graphic presentation has proved very effective in involving domain experts, and allowing them to enter knowledge directly into the computer without communicating it through a knowledge engineer.

Figure 6 shows a cluster analysis of the data entered in KSS0. This analysis is available interactively at any time, providing a different perspective on the cases which both motivates the expert and allows the coherence of the data in terms of meaningful relations between entities and between attributes to be evaluated.

Figure 7 shows the result of the user selecting HyperCard annotation on the popup menu over "Sweet scented Boletus" in Figure 5. The fields in the upper half are generated automatically from the case data, and those in the lower half are user-entered annotation, including a button to show further information.

Figure 8 shows the result of the user selecting the "Show" button in Figure 7. The additional annotation can be of any form, access to a videodisc, sound replay, database, simulation, and so on. This is not computational information, but it is an important part of the interface of the knowledge base to a user, for example, in explaining terminology.

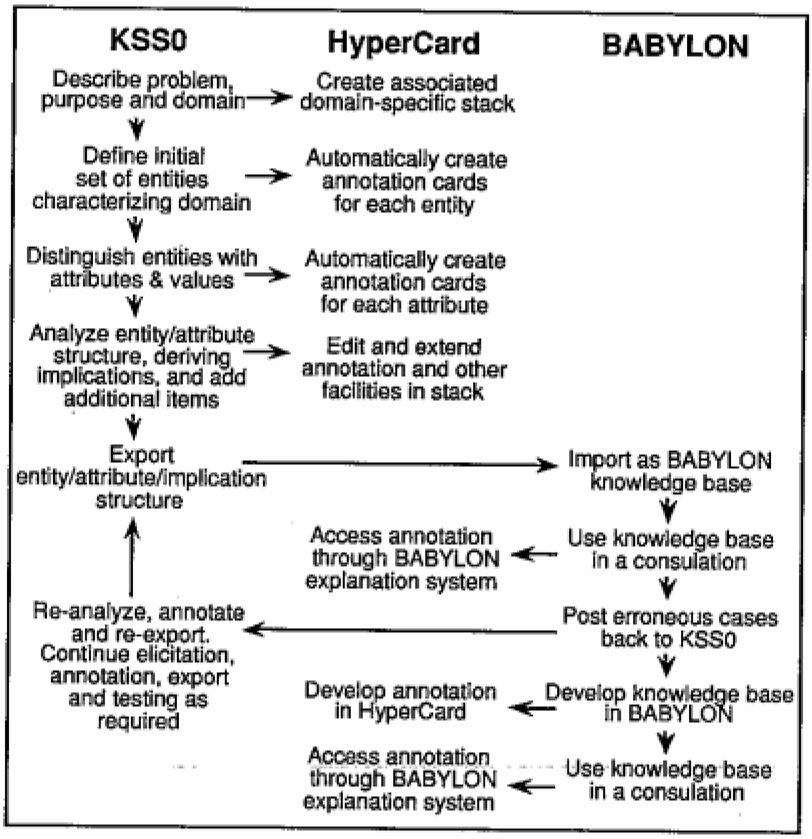

Fig. 3 Sequence of activities in using integrated system for knowledge base development

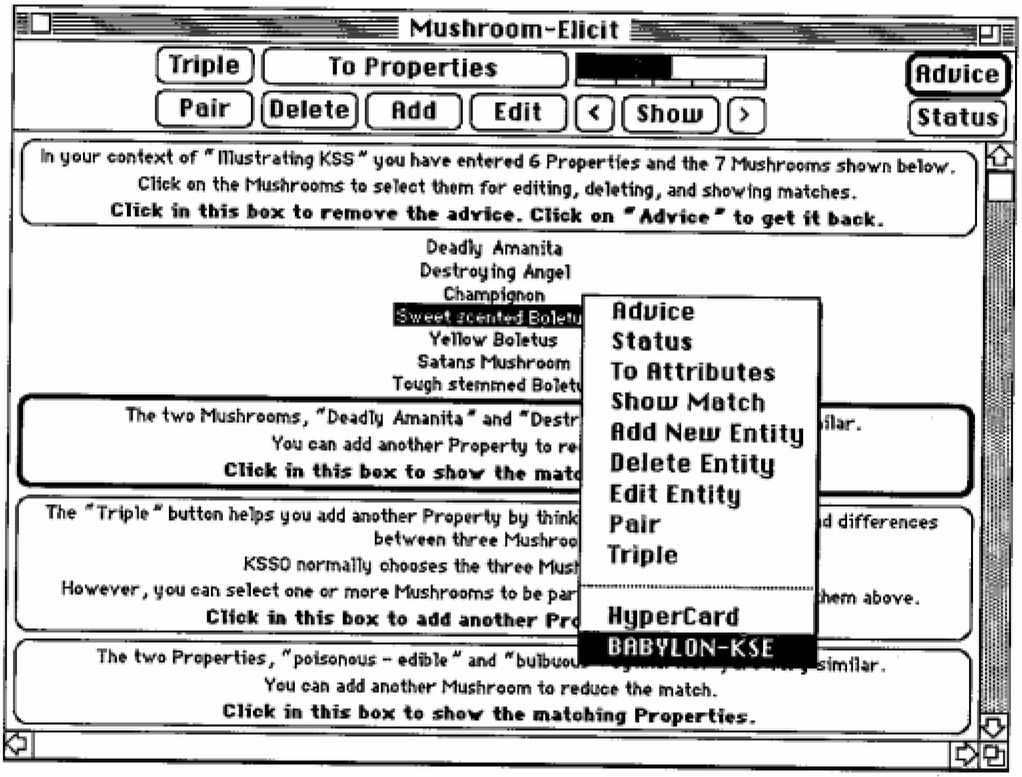

Fig. 4 Entity screen in KSS0 showing popup menu to access annotation and shell

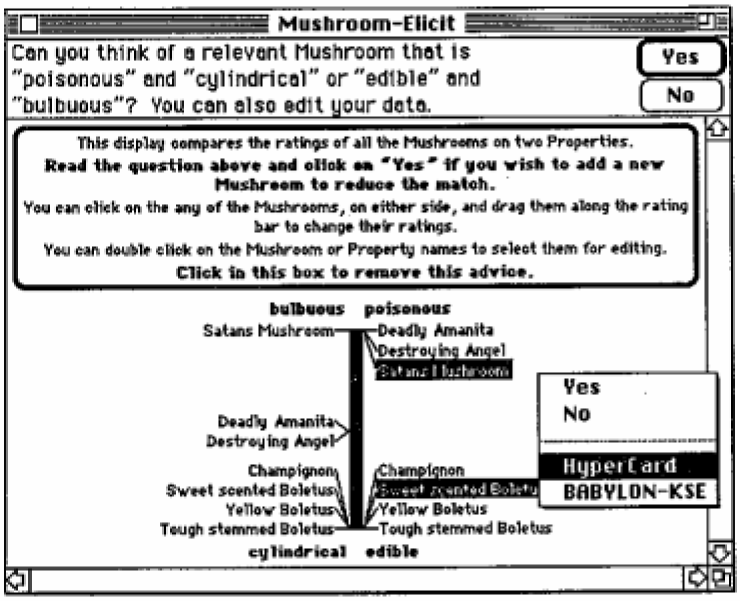

Fig. 5 Attribute match screen in KSS0 prompting entry of a new entity

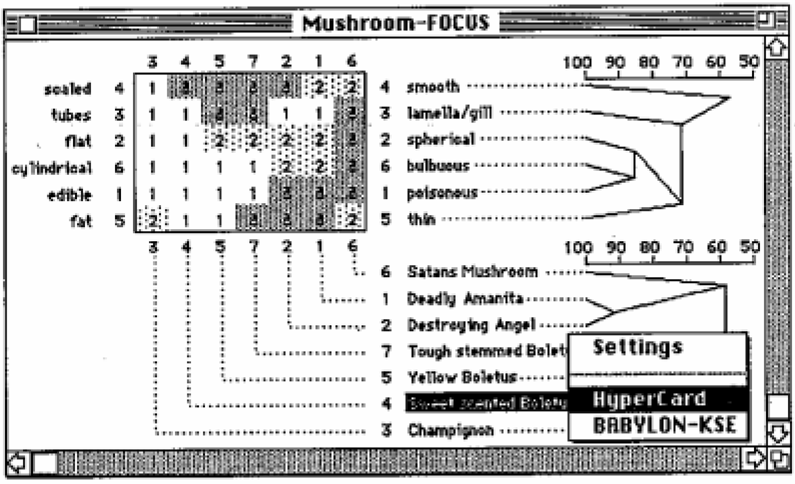

Fig. 6 FOCUS cluster analysis screen in KSS0

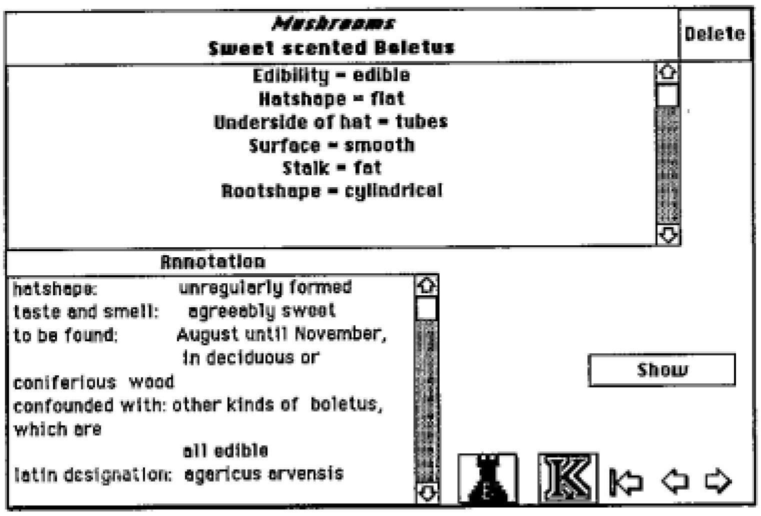

Fig. 7 Entity annotation card

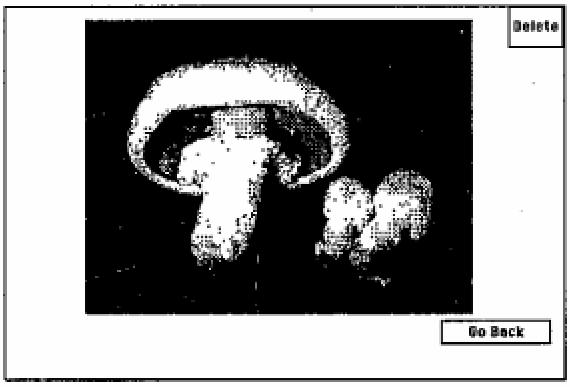

Fig. 8 Associated auxiliary annotation

Figure 9 shows a consultation running directly in BABYLON using the knowledge base entered in KSS0. The popup dialog box allows the user to select one of the possible values, or to go to the HyperCard annotation if, for example, he or she does not understand the question and needs some explanation of how this attribute should be evaluated.

Figure 10 shows the same consultation running in BABYLON with the requests for data being made through HyperCard using the attribute annotation cards for query purposes. This is the preferred mode for end users since HyperCard may be used to give a non-technical interface oriented to the particular domain with extensive help facilities. The availability of the direct mode as shown in Figure 9 is important to the knowledge engineer since it gives access to the extensive tracing and knowledge structure visualization facilities in BABYLON.

Figure 11 shows how the knowledge engineer can edit the results of a consultation on a test case in BABYLON and then post it back as a new entity to KSS0—similar facilities are available to the expert or client during the equivalent HyperCard-based consultation. These facilities put the application of the knowledge-based system into the knowledge acquisition process. They model what must be achieved throughout knowledge-based systems if we are to develop systems in which knowledge-base maintenance is an integral feature of ongoing knowledge acquisition.

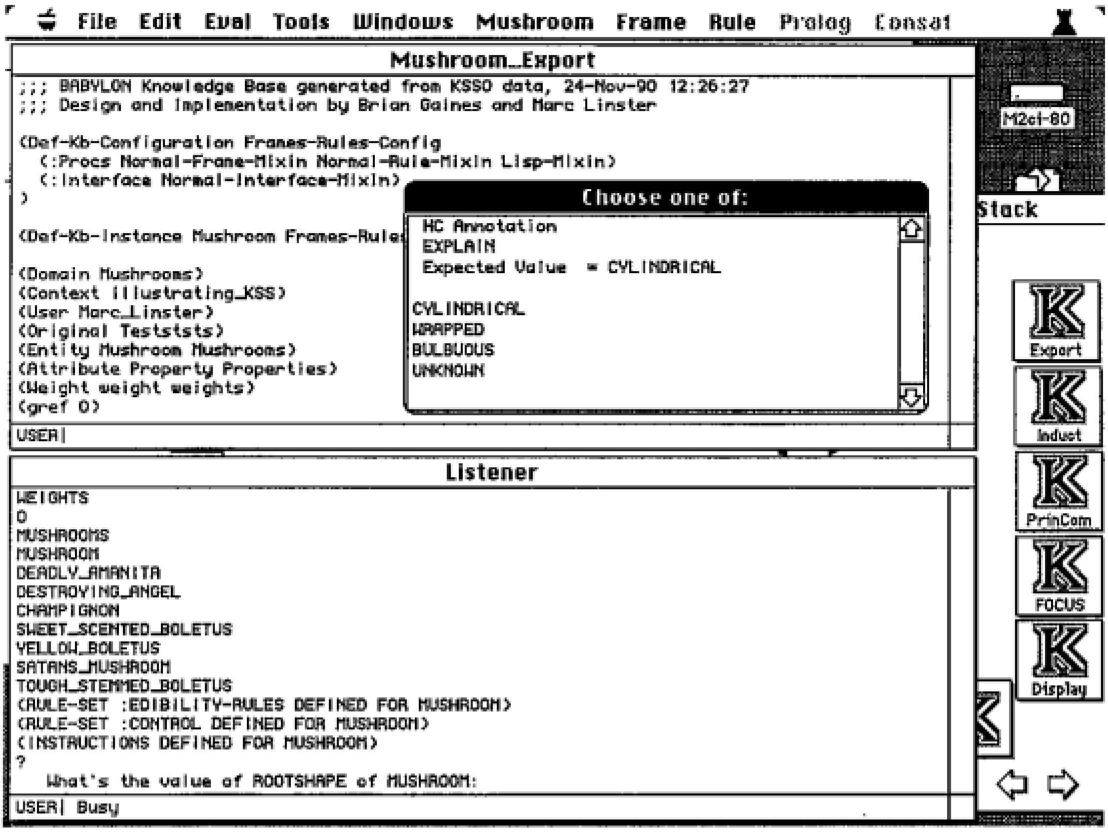

Fig. 9 Direct consultation with BABYLON

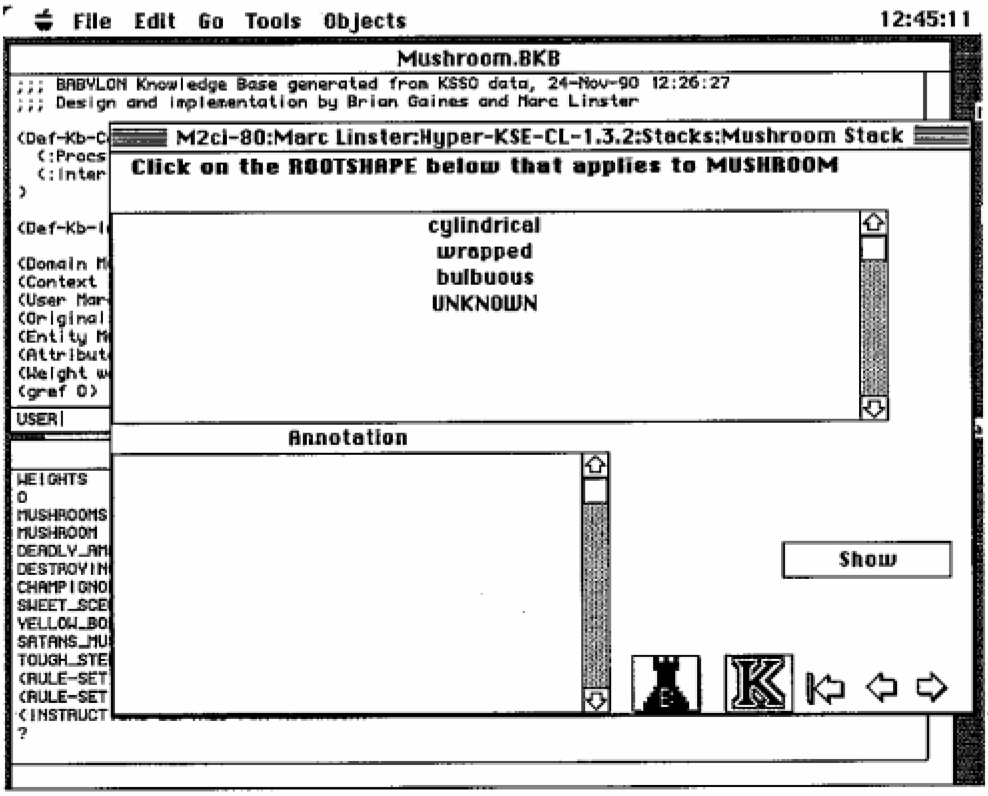

Fig. 10 Consultation with BABYLON using HyperCard as user interface

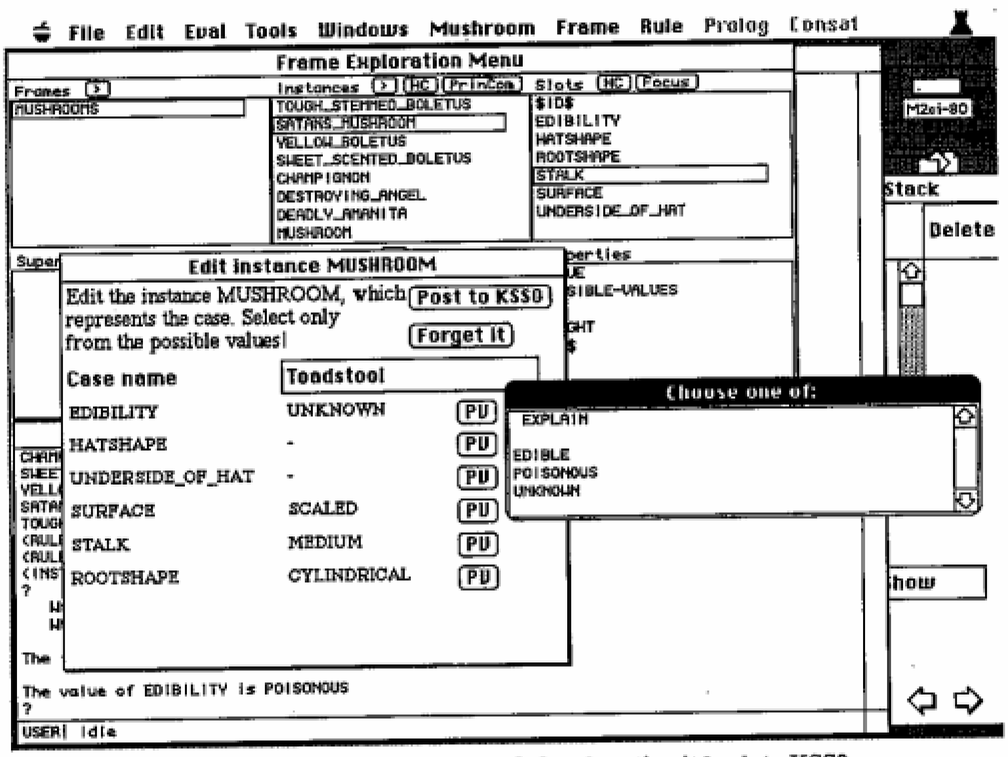

Fig. 11 Editing a case in BABYLON and posting it back to KSS0

4 Conclusions

This simple example serves to illustrate the main features of the integrated system. Each of the tools supports very large knowledge bases and a variety of applications has shown that the system scales effectively to significant applications. The graphic case elicitation tools in KSS0 operate effectively for some 20 to 50 cases which is enough to characterize the attributes of a coherent subdomain. The induction tool in KSS0 has been shown to be effective with databases exceeding 10,000 cases (Gaines, 1991d), inducing rules and evaluating edited knowledge bases rapidly enough for interactive use. HyperCard, with appropriate indexing tools, is capable of annotating databases of 10,000 cases without loss of interactivity. BABYLON has been used on a number of majorknowledge-based system developments, and has recently been reimplemented to support large scale industrial applications.

The main weakness of Hyper-KSE is in the difficulty of sustaining the functional integration of the knowledge acquisition tools in the development and application of complex applications. In a straightforward diagnostic application, soluble through heuristic classification, the major part of the knowledge base is a single, large but coherent, case base, and induced and manually entered rules. The representation and inference system has a simple task and its knowledge structures do not extend beyond those of the acquisition tools. In more complex system developments involving multiple subdomains, the acquisition tools may be used to characterize each subdomain, but the problem-solving, strategic knowledge that is involved in using the subdomain knowledge effectively has to be entered directly into the representation and inference tool. As the balance of the system changes such that this knowledge becomes increasingly important, the involvement of the acquisition tools in knowledge base maintenance is reduced.

This indicates the need for acquisition tools supporting problem-solving techniques, and knowledge-level integration based, for example, on generic problem-solving methodologies (Chandrasekaran, 1988). Some recent experiments have shown that multiple heuristic classification of subdomains may be used to solve complex procedural problems, such as sequential decision making in room allocation (Gaines, 1991e). As a wider range of structures for generic problem solving methodologies are developed it is becoming feasible to extend the type of system described here to include more meta-knowledge designed to manage the acquisition, validation, and maintenance phases systematically. Extensions to KSS0 to support such methodologies have been reported recently (Gaines, 1991a,b,c), and many researchers are working to develop the problem solving methodologies and test them in applications.

5 References

Chandrasekaran, B (1988) Generic tasks as building blocks for knowledge-based systems: the diagnosis and routine design examples. **The Knowledge Engineering Review, 3**(3), 183-211.

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

Gaines, B.R. (1990). 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. (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) and **SIGART Bulletin 2**(3), 45-56 (June).

Gaines, B.R. (1991b). Integrating rules in term subsumption knowledge representation servers. **AAAI'91: Proceedings of the Ninth National Conference on Artificial Intelligence**. pp.458-463. Menlo Park, California: AAAI Press.

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. (1991d) Induction and visualization of rules with exceptions. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Sixth AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop.** pp. 9-1-9-25. Banff (October).

Gaines, B.R. (1991e). Organizational modeling and problem solving using an object-oriented knowledge representation server and visual language. **COCS'91: Proceedings of Conference on Organizational Computing Systems**. ACM Press (November).

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).

Rescher, N. (1979) **Cognitive Systematization**. Oxford: Basil Blackwell.

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.