MANAGING QUALITY ASSURANCE IN INTEGRATED KNOWLEDGE ACQUISITION AND PERFORMANCE SYSTEMS

B R Gaines, M L G Shaw, M Linster†

Knowledge Science Institute
University of Calgary
Calgary, Alberta, Canada T2N 1N4
gaines@cpsc.ucalgary.ca, mildred@cpsc.ucalgary.ca

†Expert Systems Research Group
GMD
PO Box 1240, D-5205 St. Augustin 1, Germany
linster@gmdzi.gmd.de

Abstract: Quality assurance in knowledge acquisition not only has to be supported in every acquisition activity at every stage of development and application, but is also multidimensional and cannot be subsumed into a monolithic validation and verification methodology. For example, much of the knowledge captured is non-computational and not subject to logical analysis, and yet it plays a major role in the use of the system, particularly in the interpretation of queries and results. Much of the informal knowledge captured is never formally represented in the knowledge base, and yet it provides a major check on the completeness of that knowledge base. This paper presents some design principles for integrated knowledge acquisition and performance systems in terms of the dimensions of quality control, and illustrates them in terms of the operation of a complete system.

INTRODUCTION

This paper is based on 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. This paper focuses on the quality assurance principles motivating the design of the integrated system.

THE QUALITY ASSURANCE 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. There are many bases for analyzing the functionality but we have found a quality assurance model to be the most powerful, in that the major principles underlying other issues, such as user interfaces and forms of analysis, can be derived from it. 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.

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.

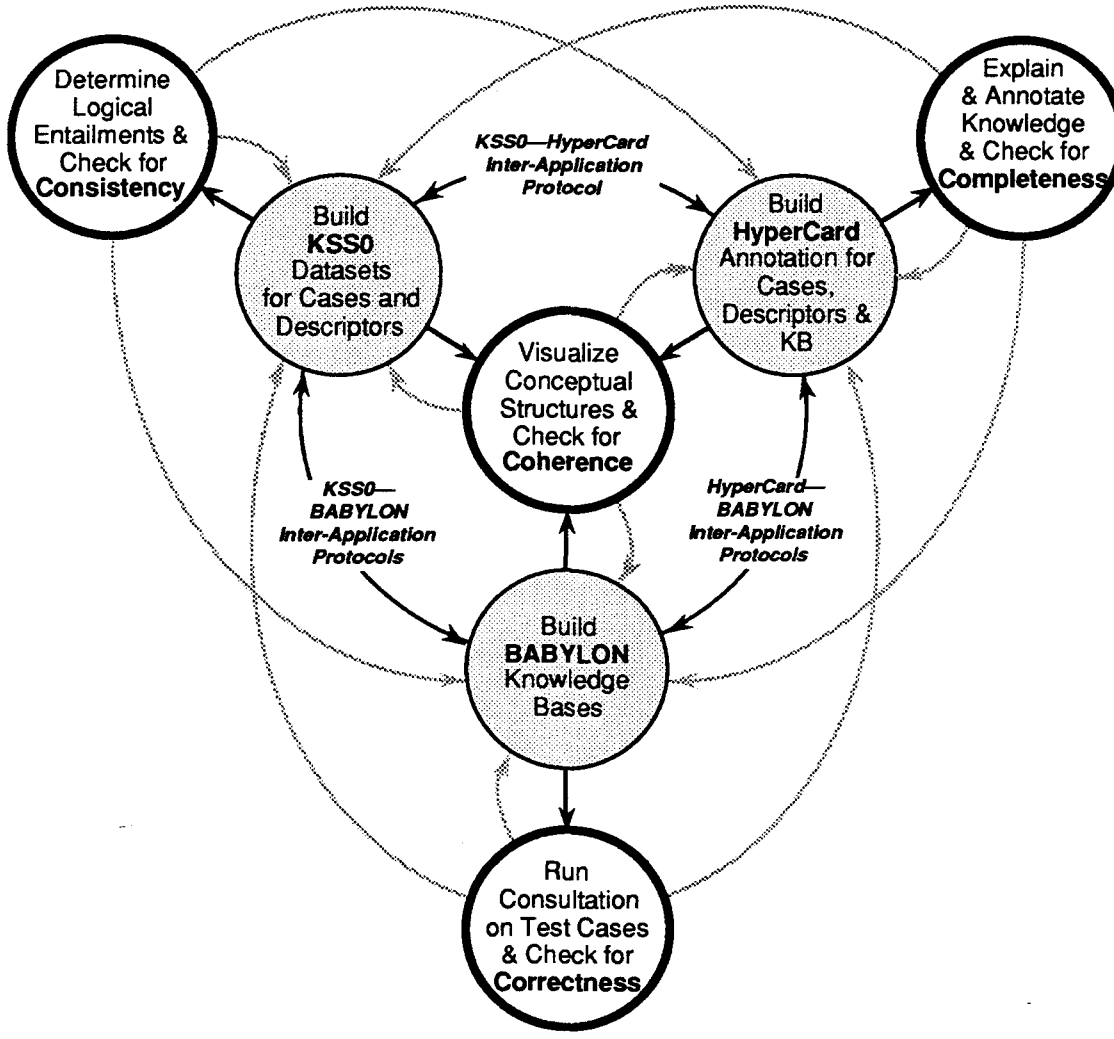

Fig. 1 Logical basis for modes of knowledge validation in the integrated system

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.

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.

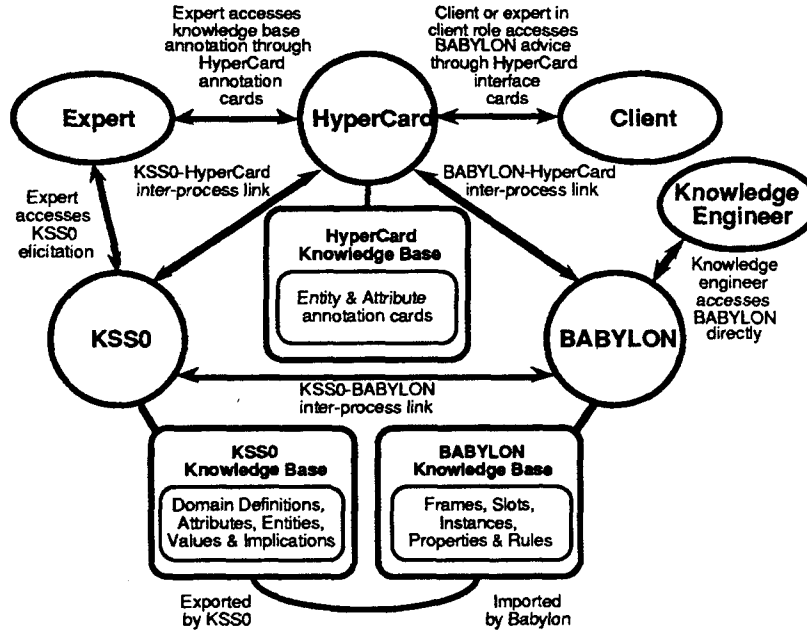

Fig. 2 Integration of hypermedia, knowledge acquisition and knowledge-based system shell

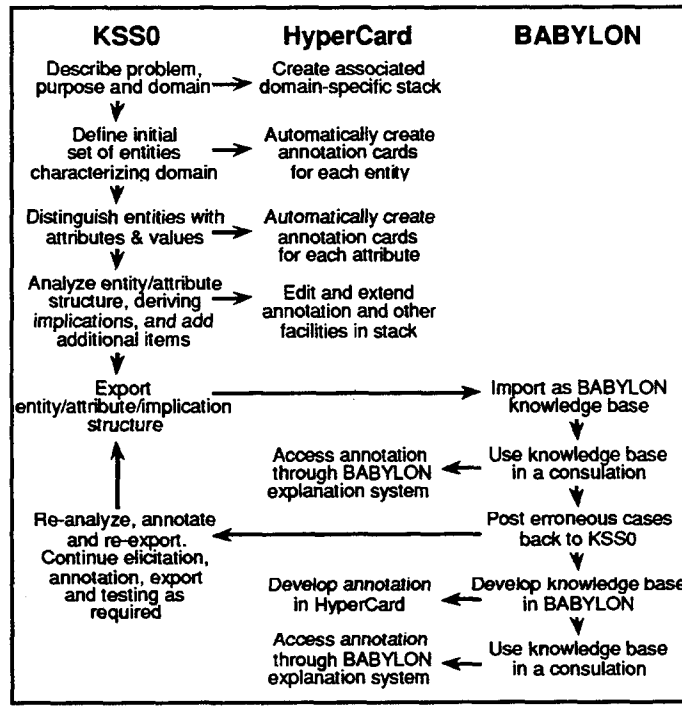

Fig. 3 Sequence of activities in using integrated system for knowledge base development

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.

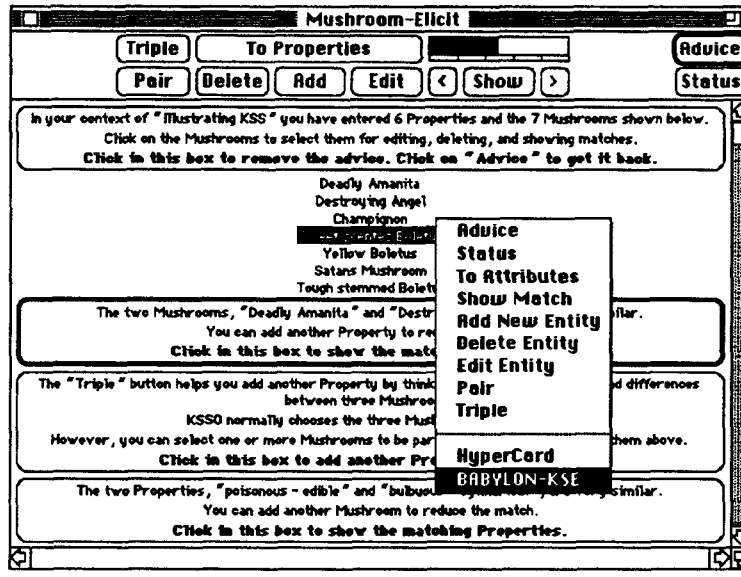

Fig. 4 Entity screen in KSS0 showing popup menu to access annotation and shell

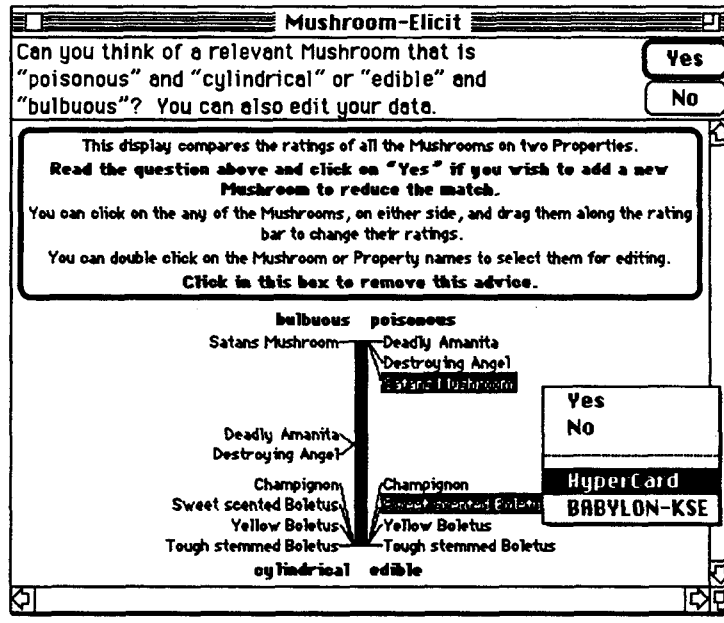

Fig. 5 Attribute match screen in KSS0 prompting entry of a new entity

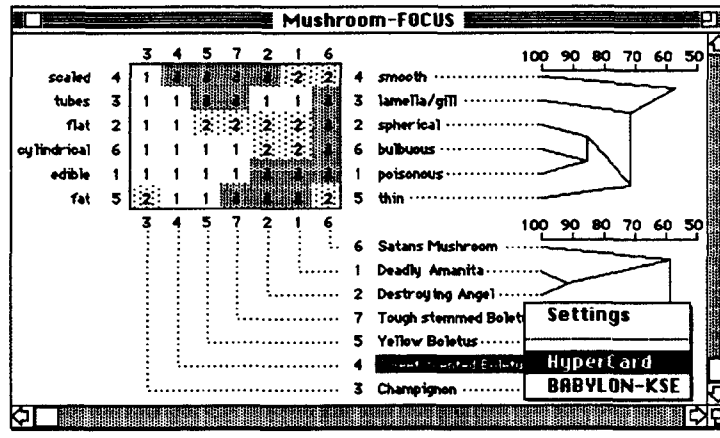

Fig. 6 FOCUS cluster analysis screen in KSS0

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.

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.

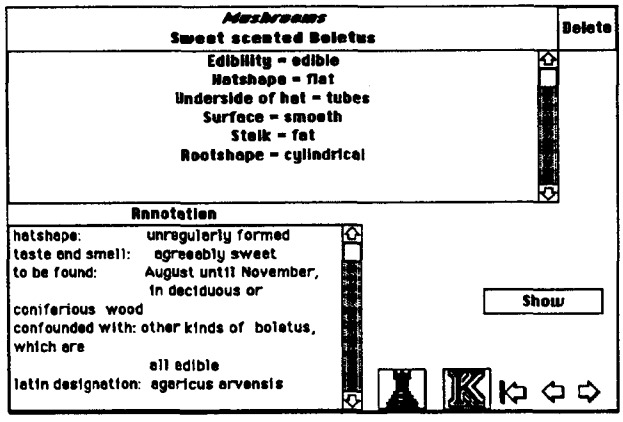

Fig. 7 Entity annotation card

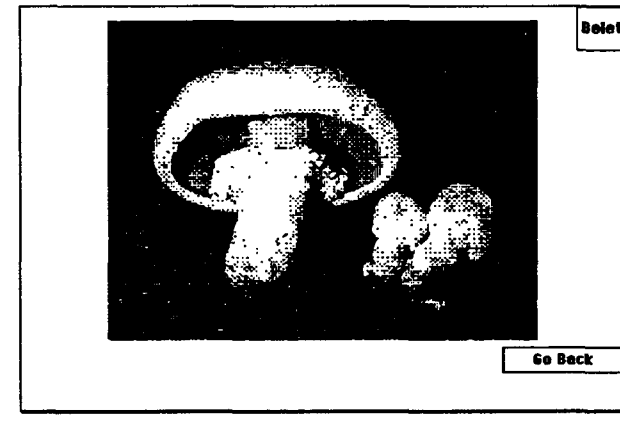

Fig. 8 Associated auxiliary annotation

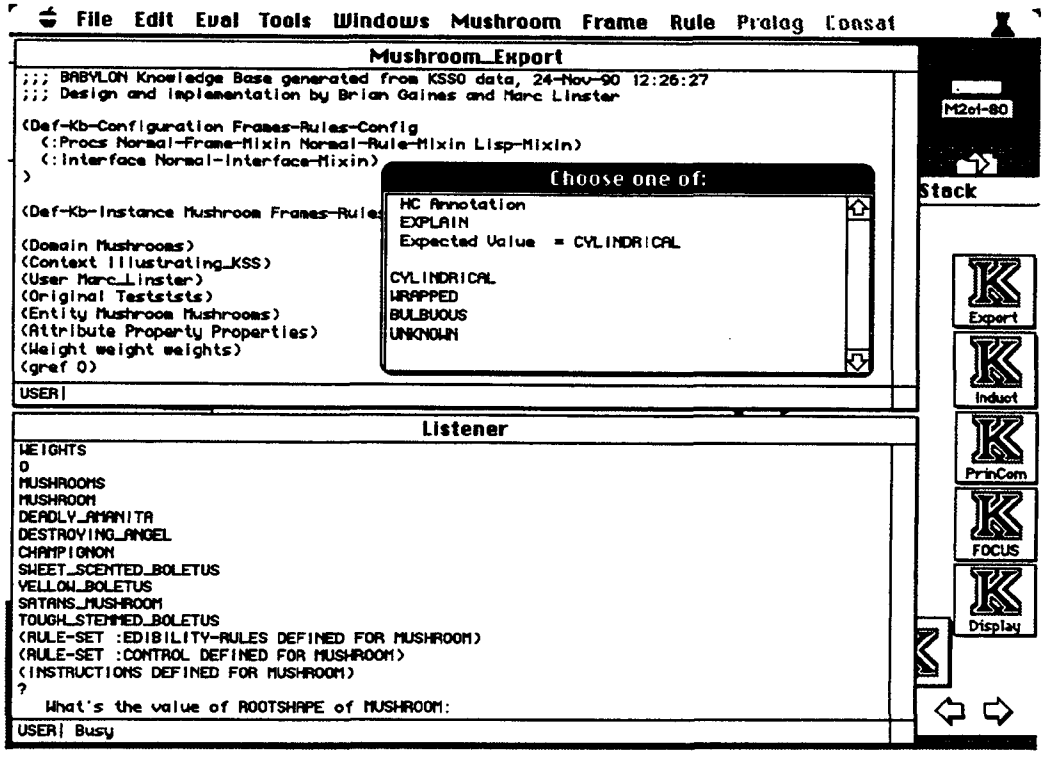

Fig. 9 Direct consultation with BABYLON

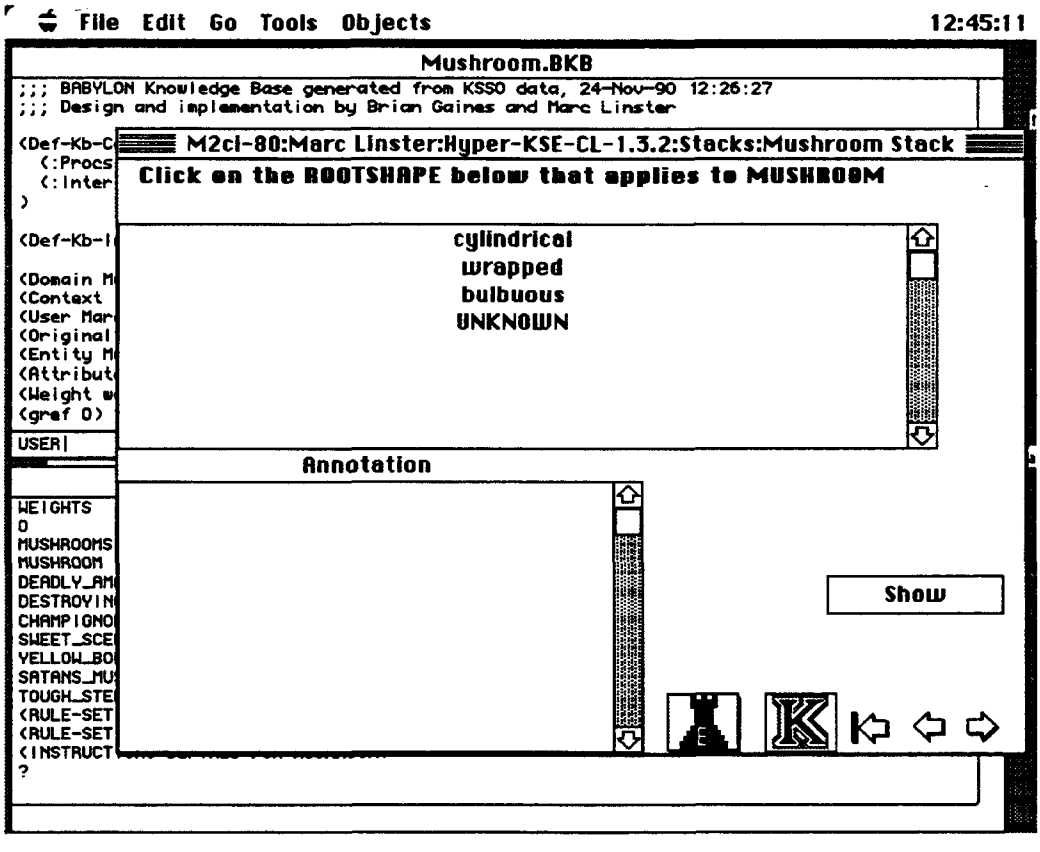

Fig. 10 Consultation with BABYLON using HyperCard as user interface

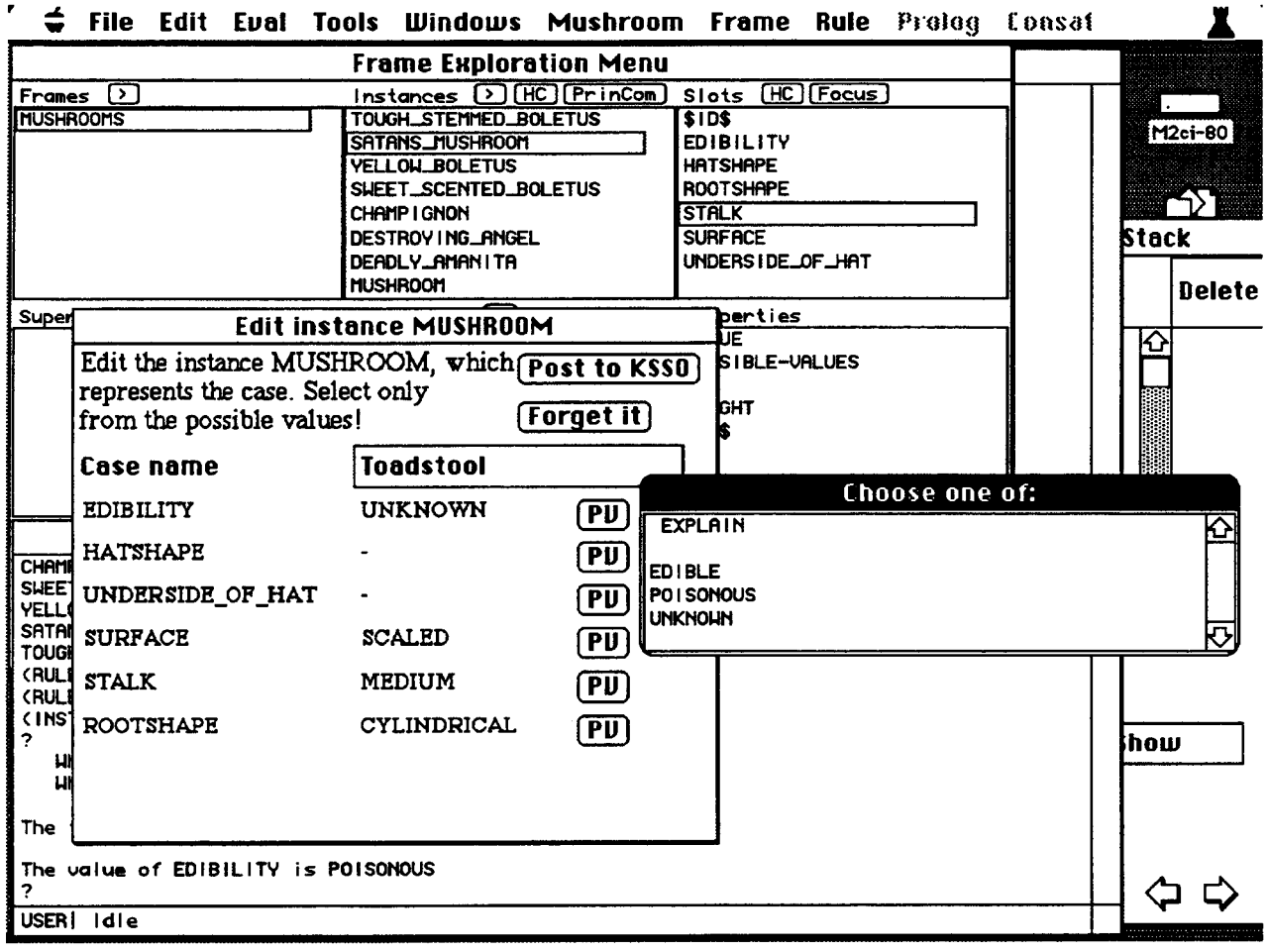

Fig. 11 Editing a case in BABYLON and posting it back to KSS0

CONCLUSIONS

Quality assurance in knowledge acquisition not only has to be supported in every acquisition activity at every stage of development and application, but is also multidimensional and cannot be subsumed into a monolithic validation and verification methodology. For example, much of the knowledge captured is non-computational and not subject to logical analysis, and yet it plays a major role in the use of the system, particularly in the interpretation of queries and results. Much of the informal knowledge captured is never formally represented in the knowledge base, and yet it provides a major check on the completeness of that knowledge base. This paper has presented some design principles for integrated knowledge acquisition and performance systems in terms of the dimensions of quality control, and illustrated them in terms of the operation of a complete system.

ACKNOWLEDGEMENTS

This work was funded in part by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

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

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.