2nd AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop,
Banff, Canada, Oct., 1987

ADVANCED EXPERT SYSTEM SUPPORT ENVIRONMENTS

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

Abstract: Tools and techniques for knowledge acquisition for knowledge based systems need to be integrated with the overall system and not treated as separate components. The variety of sources of knowledge, representations and applications within the system, and user roles, makes such integration complex. The complexity raises many issues not generally thought of as part of knowledge acquisition yet fundamentally significant in extending and integrating acquisition tools. This paper gives a preliminary analysis of the structure of knowledge-based systems as it relates to the integration of acquisition. It is intended that this structure be used as a basis for system design and it will be extended to form a 'requirements specification' for knowledge based systems integrating acquisition with other aspects of their operation.

INTRODUCTION

At the time of the first AAAI Workshop on Knowledge Acquisition for Knowledge Based Systems in Banff, November 1986, the literature on knowledge acquisition was sparse, scattered in conference publications, and difficult to access. The 1986 Workshop and the ensuing journal publication of some forty papers representing the state of the art world-wide changed that situation. The inclusion of sessions on knowledge acquisition in many conferences, the first European Workshop on Knowledge Acquisition for Knowledge Based Systems in London and Reading, September 1987, and the second Banff Workshop in October 1987, have changed the situation even further. We now have a world wide network of communicating researchers aware of one another's activities and an open, widely disseminated and readily available literature on knowledge acquisition.

The problem now is not so much to access research and experience in knowledge acquisition but to make sense of the diverse and wide ranging material available and, in particular, to apply the results to improve effectiveness in the development of knowledge-based systems. There are a number of major impediments to such understanding and application:

- A diversity of techniques and tools that overlap in their applications and where it is not clear whether they are competitive alternatives or complementary partners
- Lack of variety, detail and evaluation in the case histories of applications of the techniques and tools
- Lack of access to the techniques and tools outside the narrow research communities originating them
- Lack of standardization in the knowledge representation resulting from the techniques and tools making it difficult to integrate them and to interface them to existing knowledge based systems
- Lack of standardization in the forms of data required by the techniques and tools making it difficult to apply them in the same situation and compare them
- Lack of standardization in the user interfaces to the interactive tools making it difficult to integrate them in an effective environment for human-computer interaction
- Lack of portability in the run-time environments required by the tools making it difficult to integrate them with other systems

These are not unreasonable problems at this stage of development of knowledge based systems. We are in the midst not only of the evolution of a rapidly developing technology but also in a revolution of our understanding of the nature of knowledge and knowledge processes in society and technology. Premature standardization resulting in over-rigid systems would be inappropriate and a major impediment to progress. However, we have to balance the need for integration, dissemination and application of knowledge acquisition techniques and tools against the dangers of rigidity. This paper addresses the issues involved, the evolutionary trends, the key requirements for integration and the practical approaches that may be taken.

TRENDS IN KNOWLEDGE ENGINEERING

The basic model for knowledge engineering has been that the knowledge engineer mediates between the expert and knowledge base, eliciting knowledge from the expert, encoding it for the knowledge base, and refining it in collaboration with the expert to achieve acceptable performance. Figure 1 shows this basic model with manual acquisition of knowledge from an expert followed by interactive application of the knowledge with multiple clients through an expert system shell.

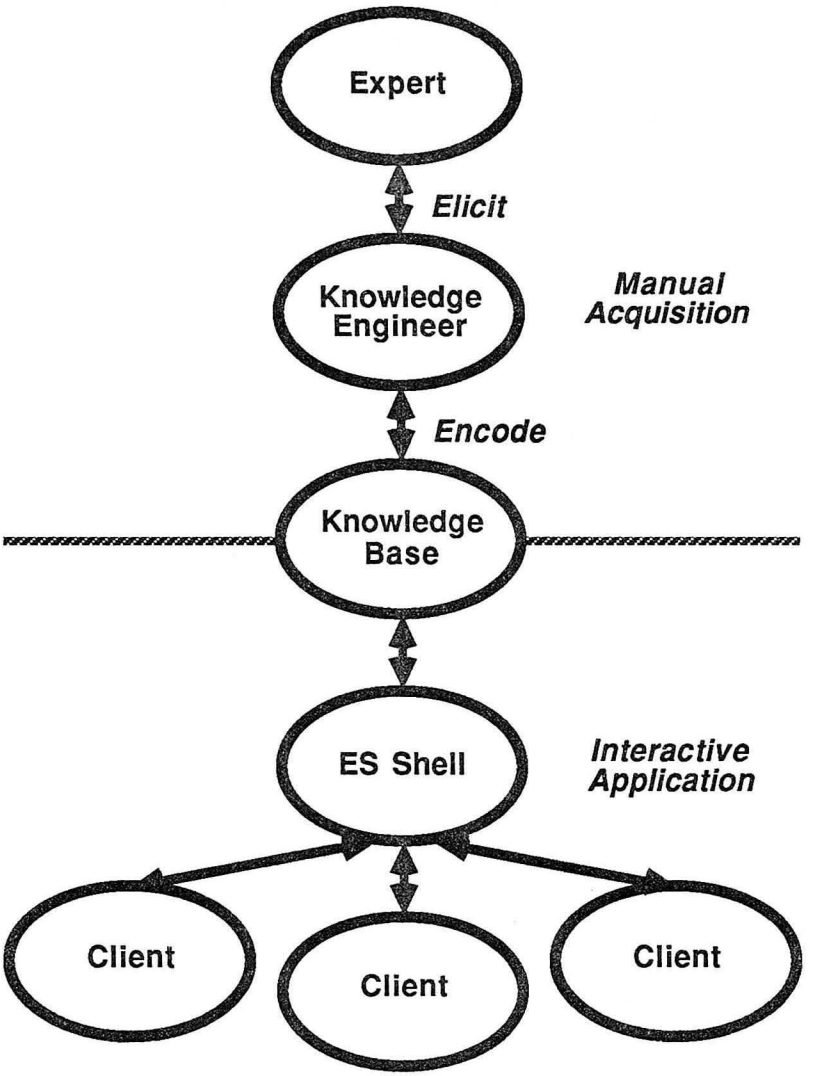


Fig.1 Basic model of manual knowledge acquisition

This basic model has been greatly extended by the introduction of interactive knowledge acquisition and encoding tools allowing the expert to transfer knowledge directly to the system without an intermediary. Such tools can greatly reduce the need for the knowledge engineer to act as an intermediary, but, in most applications, they leave a substantial role for the knowledge engineer. As shown in Figure 2, knowledge engineers can:

- Advise the expert on the process of interactive knowledge elicitation
- Manage the interactive knowledge acquisition tools, setting them up appropriately
- Edit the unencoded knowledge base in collaboration with the expert
- Manage the knowledge encoding tools, setting them up appropriately
- Edit the encoded knowledge base in collaboration with the expert
- Validate the application of the knowledge base in collaboration with the expert
- Train the clients in the effective use of the knowledge base in collaboration with the expert by developing operational and training procedures

This use of interactive elicitation can be combined with manual elicitation and with the use of the interactive tools by the knowledge engineer rather than, or in addition to, the expert. The knowledge engineer can:

- Directly elicit knowledge from the expert
- Use the interaction elicitation tools to enter knowledge into the knowledge base

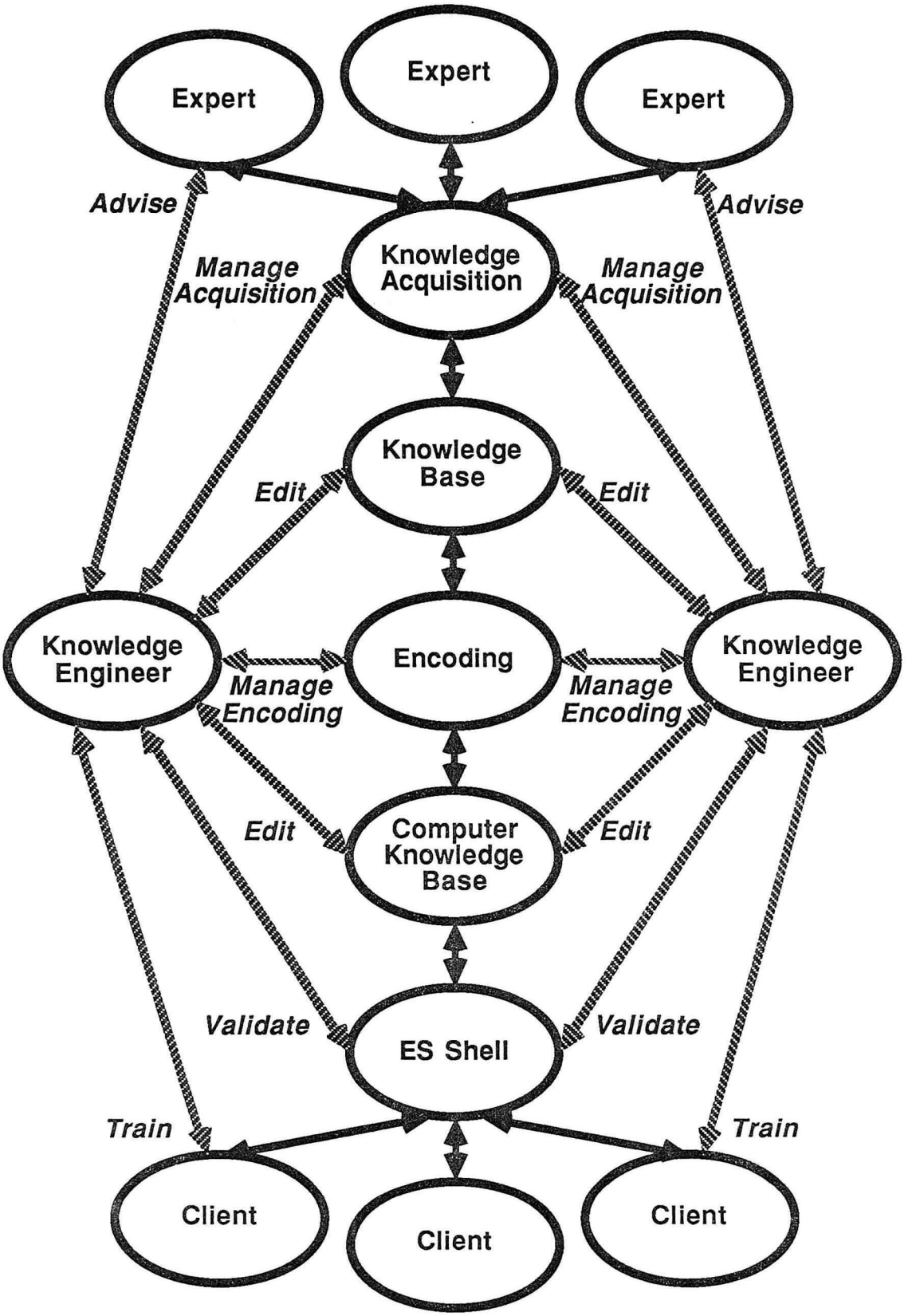


Fig.2 Knowledge engineers' roles in interactive knowledge acquisition

Figure 2 shows multiple knowledge engineers since the tasks above may require the effort of more than one person, and some specialization may be appropriate. Multiple experts are also shown since it is rare for one person to have all the knowledge required, and, even if this were so, comparative elicitation from multiple experts is itself a valuable knowledge elicitation technique.

Figure 2 shows the complexity of the knowledge engineer's role and the some of the support tools required. Figure 3 groups the support tools for editing, display, encoding and validation of the knowledge bases into an *Expert System Support Environment* (ESSE), and lumps the various forms of knowledge base together. It shows the overall structure of a knowledge-based system as a central knowledge base interacting through the knowledge acquisition tools, the expert system support environment, and expert system shell, with a user community of experts, knowledge engineers and clients.

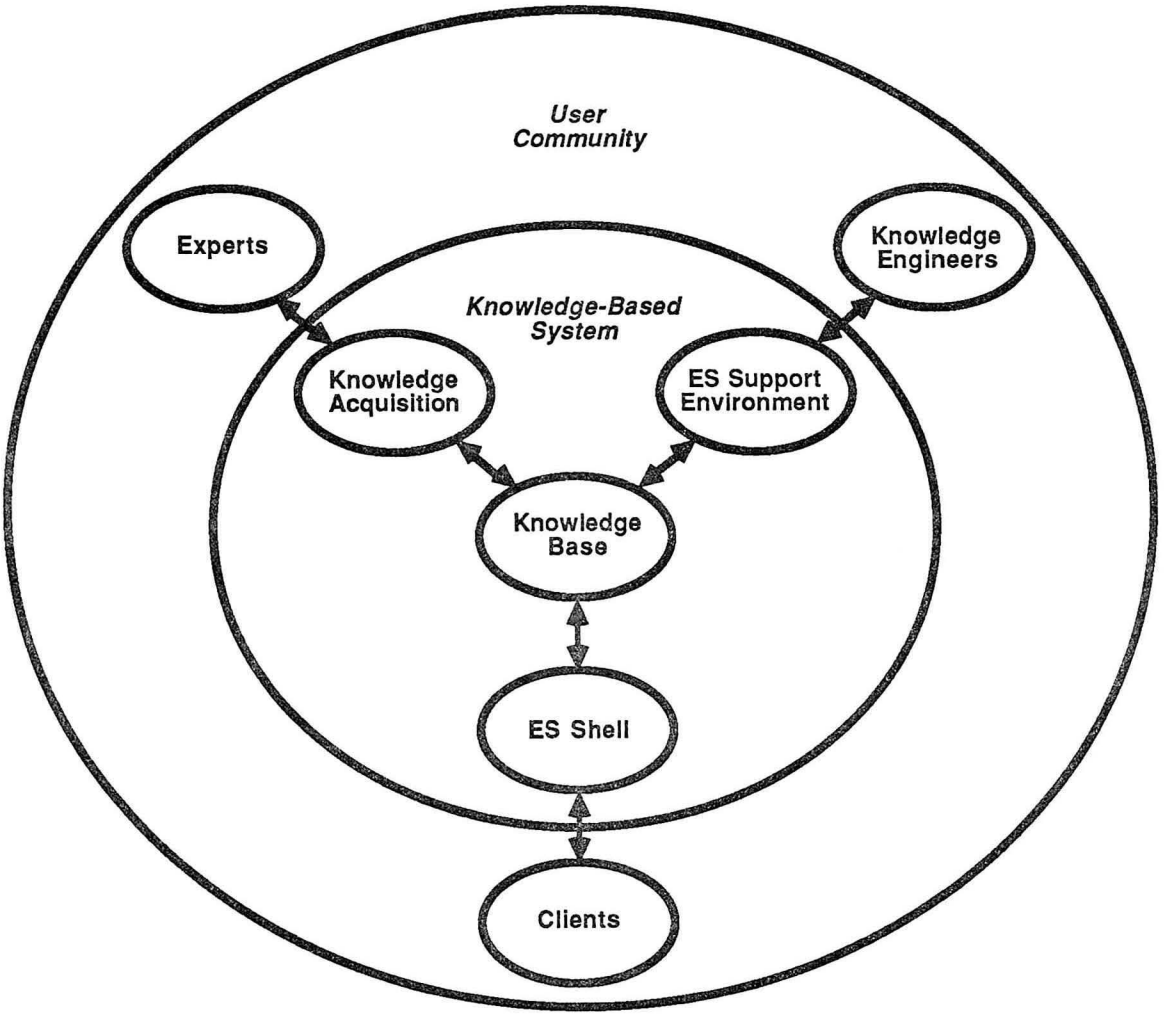


Fig.3 Major components of a knowledge based system

The notion of a user community is important in drawing attention to another trend in knowledge-based systems, that the distinction between 'experts', 'knowledge engineers' and 'clients' is often one of variable roles rather than hard and fast job definitions. The expert may be a major client for the system, or may play a major part in knowledge engineering, or both. Some clients may customize or update the system in the role of knowledge engineers or extend its capabilities in the role of experts.

INTEGRATION ISSUES IN THE USER COMMUNITY

The analysis of Figures 1 through 3 and the discussion of the previous sections raises a number of major integration issues in knowledge acquisition for knowledge based systems. Figure 4 shows these issues in terms of the major distinctions involved.

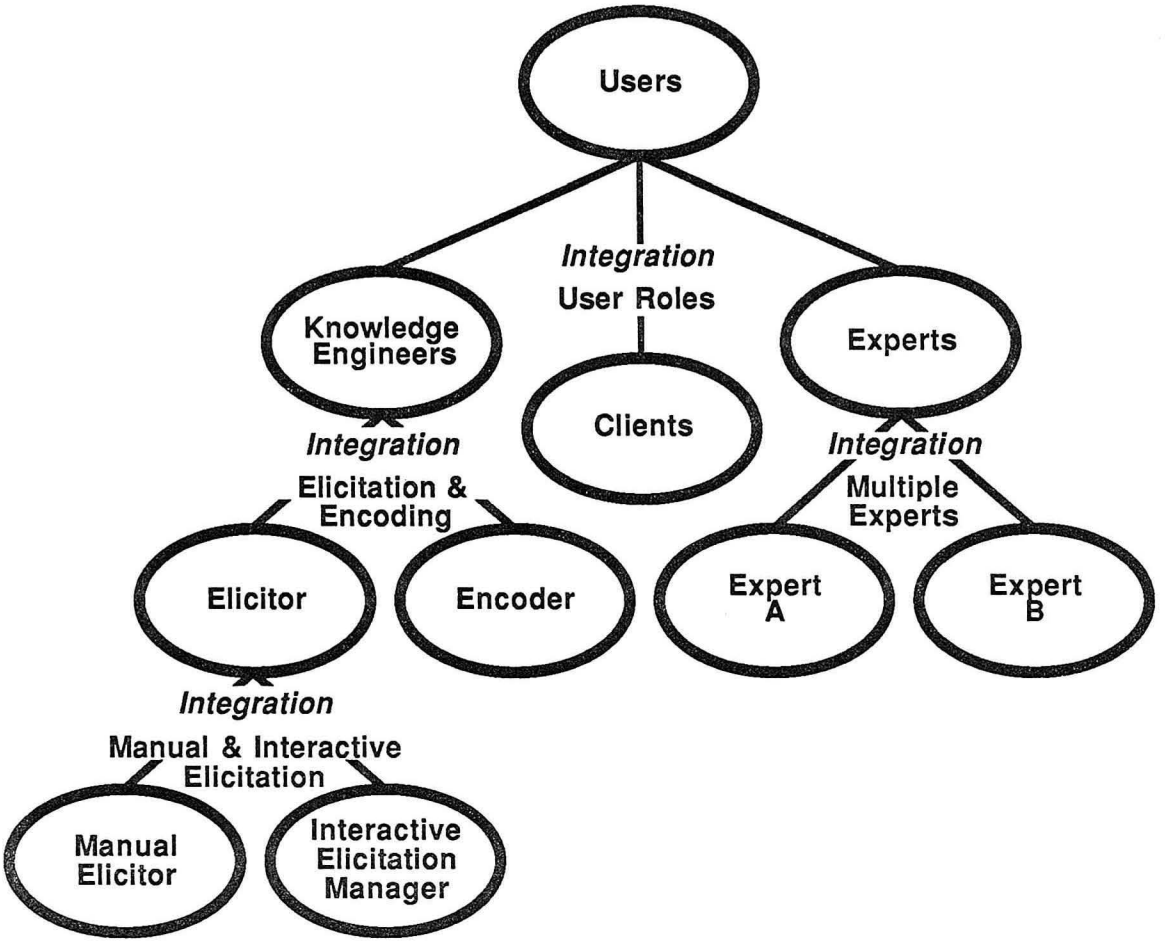


Fig.4 Major distinctions in user roles

In most current systems the knowledge acquisition tools, support tools and applications interface were designed separately and operate in very different ways on the assumption that they will be used by experts, knowledge engineers and clients, respectively. However, this diversity can be very confusing if users are flexible in their roles and switch between them. In addition, the complexity of the knowledge engineering role suggests that a variety of specializations will develop in building complex systems that will result in further roles and role switching. This leads to the design requirement:

Integration of User Roles: The user interface to a knowledge based system should be uniform and consistent across all user roles including expert, knowledge engineer and client.

However, it is important to note some features of the separation of roles:

Separation of Expert and Client Roles: The user interface to the clients should take into account the need to ensure that they use the terms in the same way as the expert. This may involve training procedures or the use of a different vocabulary.

The switch from manual to interactive elicitation has focused attention on the significance of the encoding process previously carried out tacitly by the knowledge engineer. Indeed, one of the problems in manual knowledge engineering may have been confusion between the knowledge representation meaningful to the expert and that usable by the expert system shell. A strong case has been made in manual elicitation for the use of mediating knowledge representations that are meaningful to the expert with encoding for the shell being treated as a separate phase (Johnson 1987). Such a separation is analogous to that between high level languages and machine code in conventional data processing. It may be seen as a weakness of current shell support environments that a sufficiently high level of knowledge representation is not available. The transition from Lisp code through OPS5 production rules through frames to generalized objects may be seen as the evolution of higher level knowledge representations.

The sheer variety of possible representations of knowledge that are meaningful to people makes it unlikely that the gap between a natural representation in elicitation and a computable one after encoding will ever be completely closed. Hence, as before, design requirements capturing both integration and separation are required:

Integration of Elicitation and Encoding: The knowledge representation processible by the shell in a knowledge based system should be as directly meaningful as possible to an expert.

Separation of Elicitation and Encoding: Intermediate knowledge representations should be used in elicitation for a knowledge based system if the shell representation is not directly meaningful to an expert.

The knowledge engineer's interviewing role is significant even when interactive elicitation is being used. The knowledge engineer gains information in discussion that may be used to advise the expert in the use of the tools, used to set up particular tools for elicitation, or loaded directly into the system. The system should support these various processes where appropriate:

Integration of Manual and Interactive Elicitation: Make provision for the knowledge engineer to use knowledge elicited from the expert in a variety of ways to enhance the interactive elicitation process.

And it should not be assumed that completely automatic elicitation is always possible:

Separation of Manual and Interactive Elicitation: The reduction in load on knowledge engineers through interactive elicitation should be used to enable them to be more effective in ensuring that knowledge not captured interactively is still transferred to the system.

Elicitation from multiple experts raises many problems of dealing with different vocabularies, contradictions, and so on. The system must make provision, as does AQUNIAS (Boose & Bradshaw 1987) both for integrating diverse sources of knowledge and for making contradictory sources of advice available if required. The design requirements are:

Integration of Multiple Experts: Provide tools for analyzing and expressing similarities and differences between different experts' knowledge to enable it to be combined when similar and a consensus to be negotiated when different.

Separation of Multiple Experts: When a consensus is not reached the system should be able to use contradictory sources of knowledge in generating advice by showing dissenting opinions.

The distinctions shown in Figure 4 can be extended further. For example, there may be major differences between clients making a separation between interfaces appropriate. However, the major issues relevant to knowledge acquisition are captured in the design considerations discussed.

INTEGRATION ISSUES IN THE EXPERT SYSTEM SUPPORT ENVIRONMENT

The significance of the expert system support environment shown in Figure 3 in providing tools for the knowledge engineer is apparent in the discussion above. The quality of this environment is very important in the development of expert systems, particularly in supporting the interaction between knowledge engineers and experts. Figure 5 shows some of the major issues in the ESSE in terms of the distinctions involved.

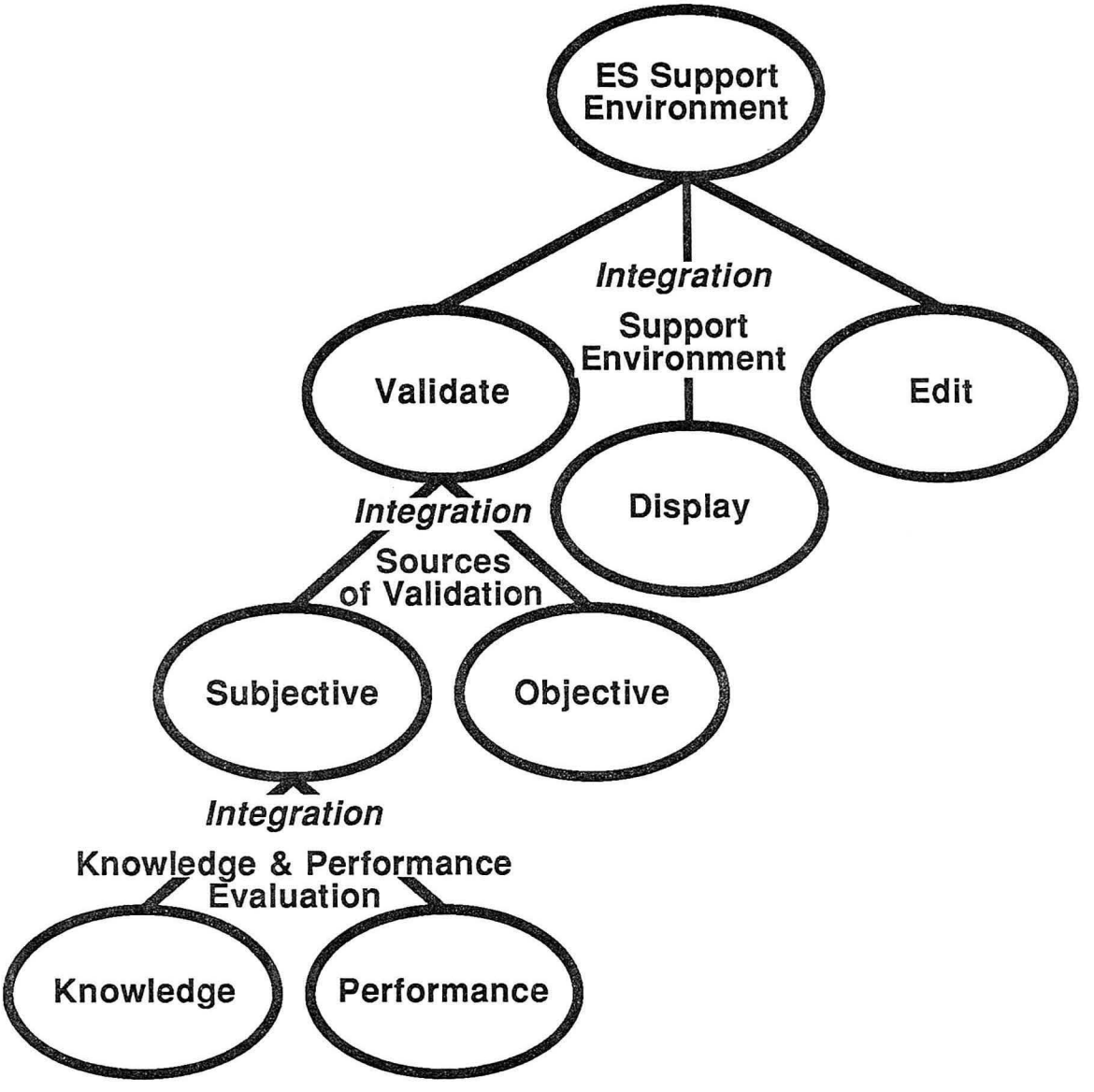


Fig.5 Major distinctions in the expert system suppport environment

The basic tools required are those for editing, displaying and validating the knowledge base. It is important that these tools present a uniform and consistent view of the

knowledge base and that they aid consistency maintenance in the knowledge base during editing. The design requirements are:

Integration of Support Environment: Provide tools for editing, displaying and validating the knowledge base which are mutually self-consistent and, if possible, automatically propagate edits to maintain consistency or, otherwise, show clearly where parts of the knowledge base have become inconsistent.

The limited converse is that different perspectives should continue to be available.

Separation of Support Environment: Support whatever variety of perspectives on the knowledge base is necessary to ensure users in different roles have access that they find meaningful.

Validation is a complex topic in its own right and it is important that a variety of tools are provided to support different forms of validation. In particular, the objective validation of the knowledge base in terms of its performance and its subjective validation in comparison with experts should be supported. The design requirements are:

Integration of Sources of Validation: Provide tools for validating the knowledge base both in terms of its absolute performance and in comparison to experts.

Objective validation may require significant case histories that can be collected only from an operational system in actual applications.

Separation of Sources of Validation: The tools for subjective and objective validation may need to be substantially different and operate over different timescales.

There is a natural separation in subjective validation between experts' direct agreement with the knowledge base and correspondence between their behavior and that of the system based on the knowledge base. Both types of validation should be supported. The design requirements are:

Integration of Knowledge and Performance Evaluation: Provide tools for enabling experts to evaluate the knowledge base directly and for comparing system and expert behavior in advising clients.

It is possible for there to be contradictions between these two forms of evaluation.

Separation of Knowledge and Performance Evaluation: Take into account that there may be contradictions between experts' evaluations of knowledge and the resulting correspondence in performance.

INTEGRATION ISSUES IN KNOWLEDGE ACQUISITION

The significant features of the knowledge acquisition component of Figure 3 are best analyzed by considering the natures of the classical and expert system design paradigms. The center column of Figure 6 shows a system for which a decision or control system is to be designed together with sources of knowledge about the system.

The left hand column shows the use of these sources in the classical system design paradigm:

- Knowledge of past case histories is used to select a class of system models
- The information required to discriminate within this class determines how the system should be instrumented for data acquisition
- Data is collected from the system through the instrumentation
- A model is identified from the model class which best fits the data
- This model is used to design a decision or control system for optimal performance

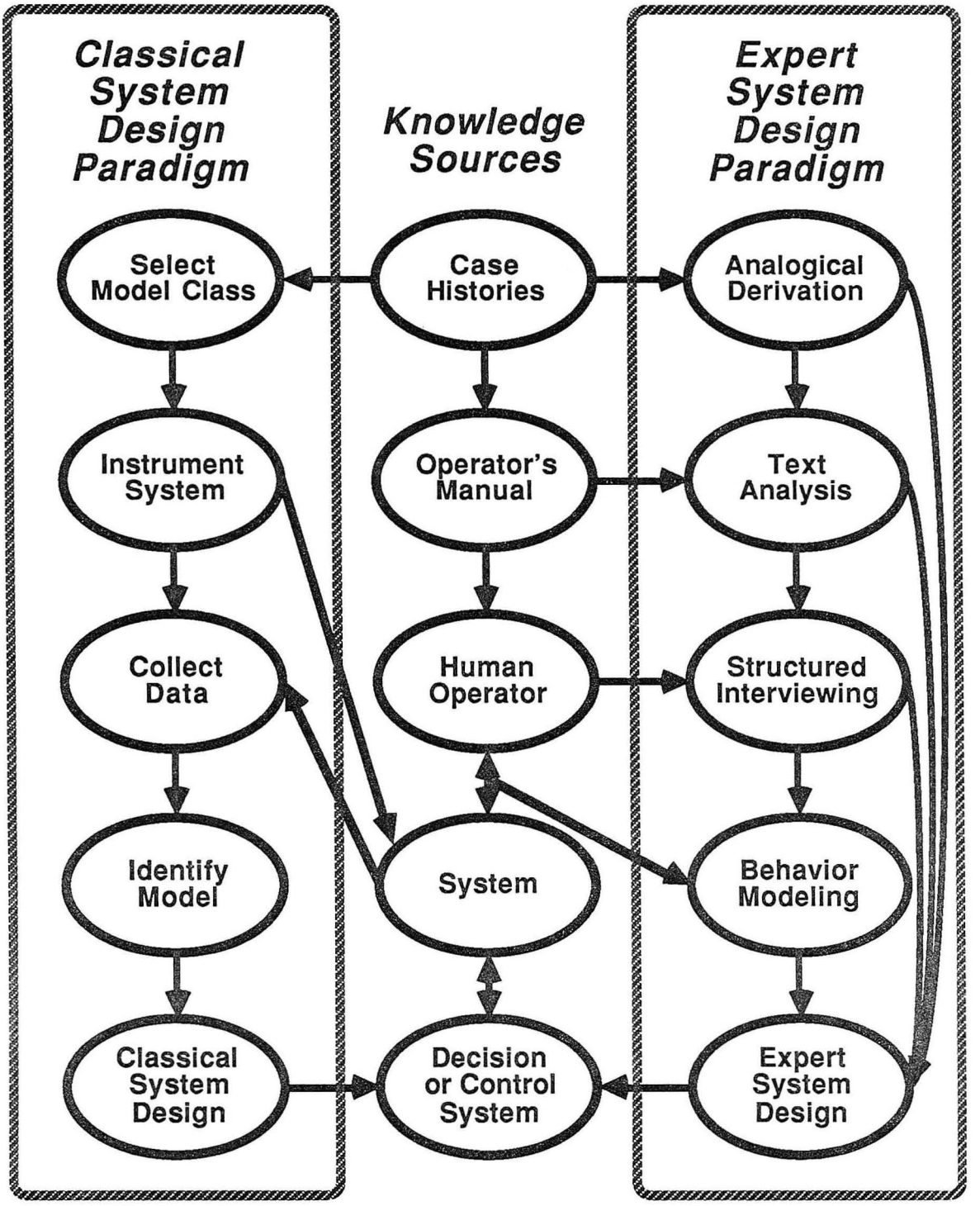


Fig.6 Classical and expert system design paradigms

The expert systems paradigm may be seen to be particularly applicable when it is not possible to model the system but there is an alternative source of data available because human operators are able to performance the decision or control task. The right hand column shows the use of knowledge sources in the expert system design paradigm:

• Structured interviewing may be used to acquire knowledge directly from the operators

- Behavior modeling may be used to identify the operators' strategies even if they are unaware of them or give incorrect ones in interviews
- Text analysis may be used with instructional material such as the operators' manuals
- Reasoning by analogy may be used based on the case histories without the data collection of the classical system design paradigm

Note that the classical and expert system design paradigms, and the various acquisition techniques for expert systems, need not be regarded as competitive. They may all be used in the system design, and it may be that a collection of weak methods is necessary because there is no overall strong method that can be applied. The resultant integration issues are analyzed in Figure 7 in terms of the distinctions involved.

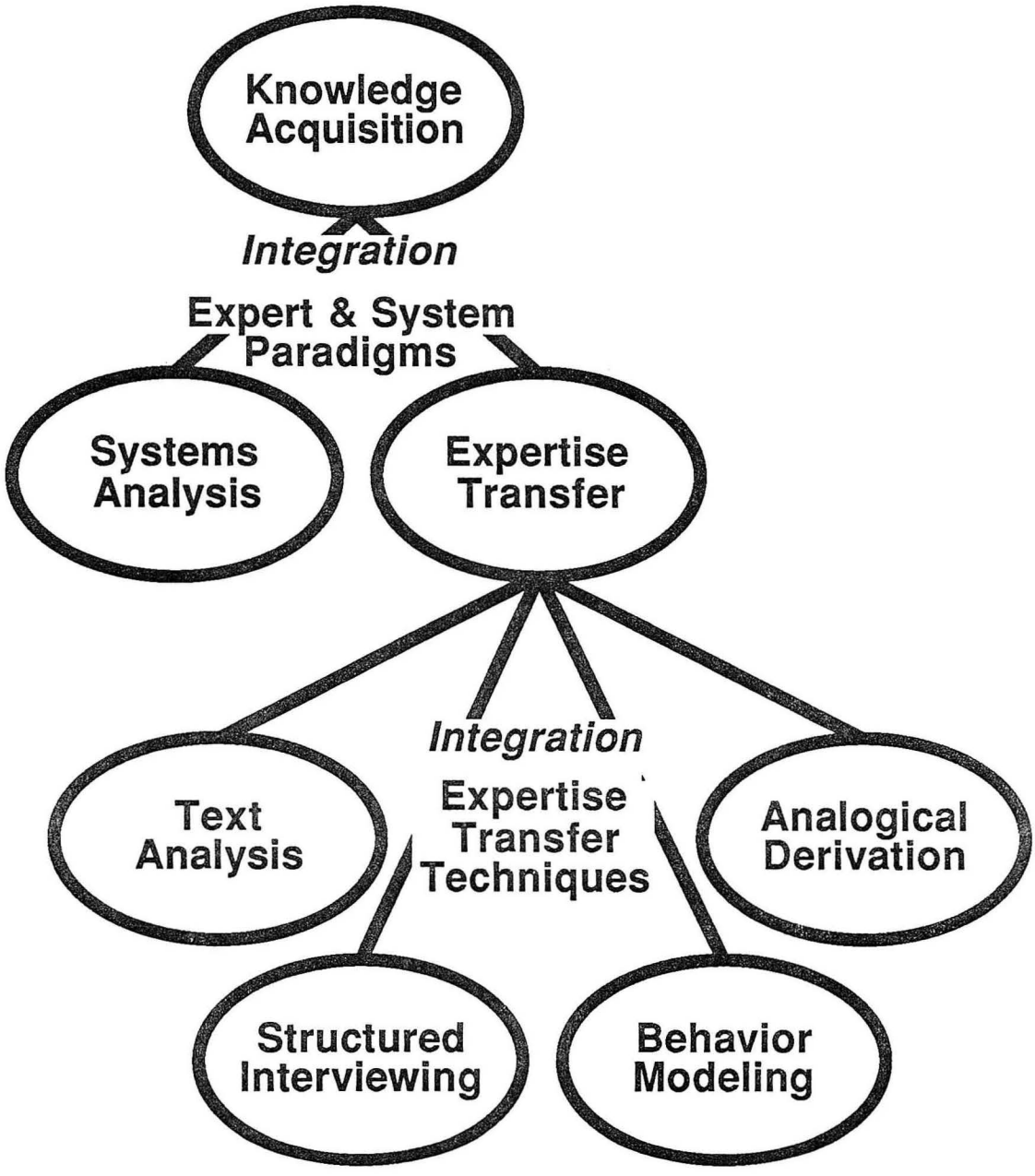


Fig.7 Major distinctions in knowledge acquisition approaches

The integration of the classical systems analysis with expertise transfer techniques is a major task. The primary problems currently are in the widely different forms of knowledge representation usually involved, and these will be noted in a later section in terms of integration of conventional and knowledge-based system components. However, at a knowledge acquisition level the problem can be seen as one of the imbalance in the types of knowledge structures studied in depth to date. There has been an imbalance favoring symbolic knowledge to redress the past emphasis on quantitative methods. This needs redressing with the knowledge structures underlying arithmetic being integrated with those underlying non-arithmetic symbol manipulation. The design requirements are:

Integration of Systems Analysis and Expertise Transfer: Provide knowledge representation that is adequate to encompass quantitative and qualitative data and reasoning.

The success of the expert systems approach should not detract from the long term significance of building deep models of the system itself. The objective performance evaluation of expert systems already discussed can provide data on deeper issues. It corresponds to learning from experience in human experts.

Separation of Systems Analysis and Expertise Transfer: Provide for system analysis of performance data from expert systems in applications to improve their performance and the depth of their models.

The integration of various techniques for expertise transfer is already a theme at the knowledge acquisition workshops. For example, case history and text analysis may be used to provide terms and concepts that aid interviewing and behavior description. The design requirements are:

Integration of Expertise Transfer Techniques: Provide intermediate knowledge representations that allow for transfer between different sources of knowledge.

The limited converse is that the different sources of knowledge, and the possibilities of contradictions between them, should be recognized.

Separation of Expertise Transfer Techniques: Keep track of the different sources of knowledge within the system and allow for contradictions in inferences from it.

INTEGRATION ISSUES OVERALL

Figure 8 combines the structures of Figures 4, 5 and 7 into an overall analysis of the distinctions underlying integration issues in knowledge acquisition for knowledge based systems. It raises some additional integration issues.

At the middle left of Figure 8 a distinction has been made between the application and explanation components of the expert system shell. This is important in acquisition because the knowledge required for explanation may be more extensive and more demanding than that required for application. Explanation facilities in most shells are very weak currently but as they develop to take into account the more advanced analyses of the nature of explanation (Hughes 1986) the requirements for explanation-oriented knowledge elicitation will increase. The design requirements are:

Integration of Application and Explanation: Explanation should be treated as a major component of application and knowledge acquisition should be targeted accordingly.

In many areas such as validation there are major differences between application and explanation that need to be taken into account.

Separation of Application and Explanation: The need for separate validation of performance in application and explanation should be taken into account.

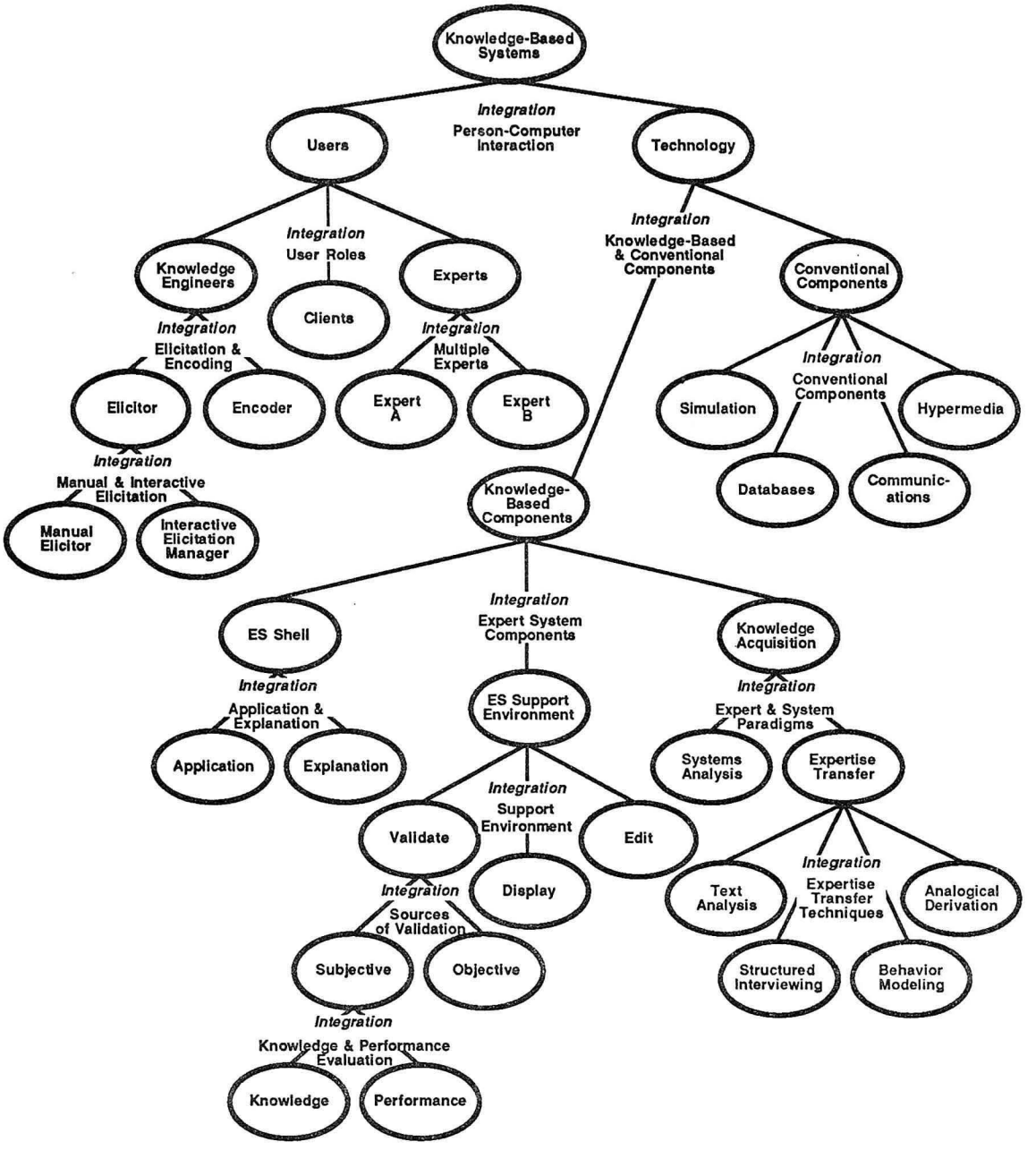


Fig.8 Overall structure of distinctions relating to knowledge acquisition integration in knowledge based systems

At the top right of Figure 8 some of the conventional components of a data processing system have been introduced and a distinction has been made between them and the knowledge-based components. The term 'knowledge' has tended to be used in a specialized way in the knowledge based system context as 'knowledge which is symbolic and computationally processible.' Much significant knowledge does not come within this narrower definition yet is still important in knowledge based systems. This leads to a need to integrate several conventional data processing technologies, for example:

- *Simulation* is important in showing the time sequences of the behavior of dynamic systems

- *Databases* provide a convenient means for storing large quantities of data with efficient retrieval, and can be used to build a variety of object hierarchies in knowledge bases from different perspectives
- *Communications* is important in allowing interaction between experts at remote sites during group elicitation processes
- *Hypermedia* provide a means for storing and retrieving knowledge even if it is not computationally processible and, in particular, provide for knowledge transfer through video material allowing mimicking at a non-verbal level

The design requirements are:

Integration of Knowledge-Based and Conventional Components: Extend the concept of a knowledge-based system to include the knowledge structures of conventional data processing.

The only danger is that some essential features of knowledge based systems are lacking in existing forms of data processing.

Separation of Knowledge-Based and Conventional Components: Extend conventional data processing to include features of knowledge based systems such as explanation.

At the top of Figure 8 the final integration is between the users and technology, that is the achievement of effective person-computer interaction. This is a major problem in all systems of the complexity of that shown in Figure 8, and is already apparent in knowledge acquisition even at the level of integration of Figure 7. The gap between functionality and usability can be very difficult to bridge, and it is easy to create multi-source, multi-format, multi-window operating environments that present great power but utter confusion to the user. The design requirements are:

Integration of Users and Technology: Provide a uniform and consistent user interface to the knowledge based system that maximizes the usability of the available functionality.

However, the interface must reflect reality and not be cosmetic.

Separation of Users and Technology: Do not use syntactic sugar and other interface tricks to give the user the impression of greater capabilities than exist in the system.

CONCLUSIONS

Tools and techniques for knowledge acquisition for knowledge based systems need to be integrated with the overall system and not treated as separate components. The variety of sources of knowledge, representations and applications within the system, and user roles, makes such integration complex. The complexity raises many issues not generally thought of as part of knowledge acquisition yet fundamentally significant in extending and integrating acquisition tools. This paper has given a preliminary analysis of the structure of knowledge-based systems as it relates to the integration of acquisition. It is intended that this structure be used as a basis for system design and it will be extended to form a 'requirements specification' for knowledge based systems integrating acquisition with other aspects of their operation.

ACKNOWLEDGEMENTS

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada. I am grateful to many colleagues at the knowledge acquisition workshops for discussions which have highlighted many of the issues raised in this paper.

REFERENCES

Boose, J.H. & Bradshaw, J.M. (1987). Expertise transfer and complex problems: using AQUINAS as a knowledge acquisition workbench for expert systems. **International Journal of Man-Machine Studies 26**(1) (January).

Hughes, S. (1986). Question classification in rule-based systems. Bramer, M.A., Ed. **Research and Development in Expert Systems III**. pp.123-131. Cambridge: University Press.

Johnson, N. (1987). Mediating representations in knowledge elicitation. **Proceedings of the First European Workshop on Knowledge Acquisition for Knowledge Based Systems**. pp.A2-1-10. Department of Computer Science, Reading University.