FOUNDATIONS OF KNOWLEDGE ACQUISITION

Brian R Gaines, Killam Memorial Chair, & Director, Knowledge Science Institute

Mildred L G Shaw, Professor, Department of Computer Science, and KSI

University of Calgary, Alberta, Canada T2N 1N4

gaines@cpsc.ucalgary.ca

Knowledge acquisition is recognized as a major bottleneck in the development of knowledge-based systems. The first generation of knowledge acquisition techniques in the 1970's were based on human interviewing. The second generation in the 1980's have been based on computer-aided application of specific techniques such as repertory grids, behavior modeling and text analysis. Current third generation research is targeting tools for the 1990's to provide integrated knowledge acquisition environments supporting a wide range of complementary tools and techniques. This paper surveys the foundations of knowledge acquisition, providing a framework for the wide range of techniques in use.

1 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 and book publication of some forty papers representing the state of the art world-wide, changed that situation [7, 22]. The inclusion of sessions on knowledge acquisition in many conferences, the European Workshops on Knowledge Acquisition for Knowledge Based Systems in London, Bonn, Paris and Amsterdam, and the later AAAI Banff Workshops, 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 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 impediments to such understanding and application:

- A diversity of techniques and tools that overlap in their applications—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 expected 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. Certainly we may expect the major thrust of research on knowledge acquisition for knowledge-based systems during the 1990's to be concerned with overcoming these impediments.

The workshops are playing a major role in this, and the involvement of technological communities world-wide is to be welcomed. The First Japanese Workshop on Knowledge Acquisition for Knowledge-Based Systems provides state of the art surveys of knowledge acquisition methodologies, techniques and tools, and research papers illustrating current developments. This paper provides a background to this material by surveying the foundations of knowledge engineering.

2 KNOWLEDGE-BASED SYSTEMS

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

Hawkins [26] has abstracted from industrial experience in developing mineral exploration expert systems and proposed a model of human expertise relevant to expert systems. Figure 1 presents a summary of the essential features of this model:

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

Thus, in Hawkins' model, the client plays an active role in further developing the model by providing more data until he or she is satisfied with the model and consequent advice. Expert advice giving and taking is part of a cycle of negotiation around a process of model formation as shown in Figure 1.

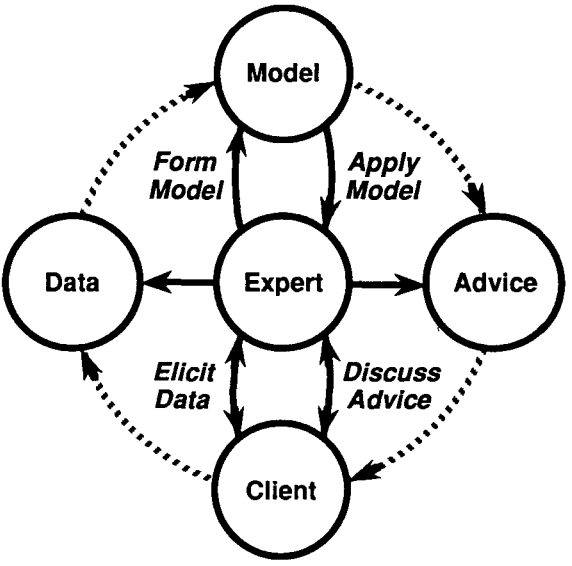

Fig.1 The negotiation cycle in expert-client interaction

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

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

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

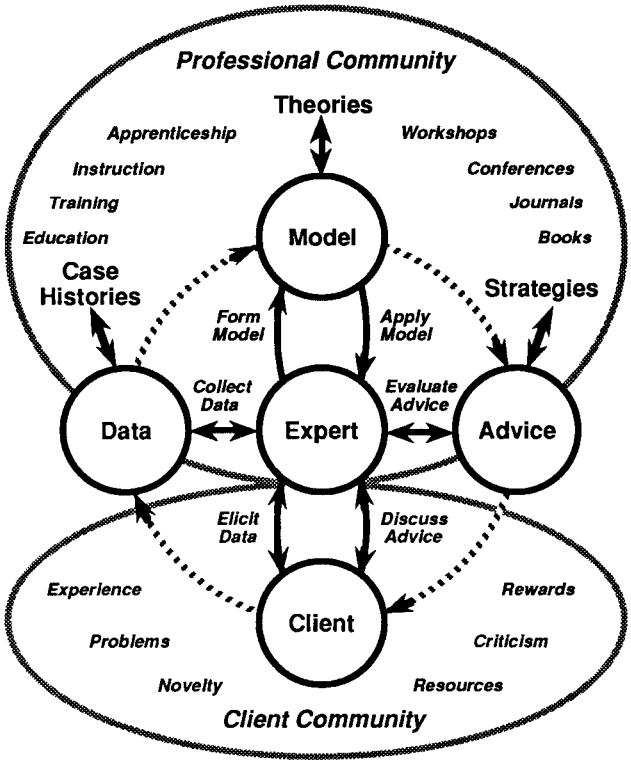

Fig.2 Extended knowledge acquisition process in expert-community interaction

This model of individual expertise as intrinsically a knowledge acquisition process at multiple levels, embedded in and supported by essential social processes, is consistent with Stich and Nesbitt's [42] view of the social role of experts as managers of the *reflective equilibrium* of the inductive process of knowledge generation in society. It is also consistent with the emphasis in professional development books on the importance of balancing experts' authoritarian roles as knowledge repositories and appliers with their entrepreneurial roles as knowledge acquirers [35].

Sociologists have noted the very strong positive feedback processes in the dynamics of knowledge acquisition in the scientific community [25]. Merton [32] coined the term the "Matthew effect" for those features of the reward system in research that were biased towards allocating greater credit for the same discovery to those with an already established reputation. The qualitative effect of such positive feedback processes is to amplify random differences within a population, creating strong distinctions which have no other basis than the feedback process itself. That is, even if all scientists were created equal in their capabilities, some would become experts relative to others because random differences in early performance generated credit that affected resource allocation and gave them greater opportunities to acquire knowledge. The strength of this effect can be demonstrated by studies of the behavior of 'experts' simulated by learning automata in a competitive learning environment with access to resources related to demonstrated 'expertise' [18].

In many areas of expertise where expert systems are expected to play a major role these positive feedback processes are exceptionally strong. In medicine, for example, the key learning resource is access to medical problems, and the 'owner' of such a problem has a strong personal interest in only allowing someone of very good reputation to handle it. The system, including considerations of legal liability, strongly funnels problems to those who are regarded as experts. It is, however, precisely these problems from which new knowledge is generated. Hence direct experience of case history data is made available primarily to those with already established reputations for having capabilities to deal with it. Similar considerations apply to the award of scholarships, invitations to scientific congresses, and so on [2]. They also apply not only to individuals but also to social units, such as a company subject to government procurement procedures that are heavily biassed to contractors with prior experience and with whom the government agency has prior experience.

The introduction of knowledge-based systems in the processes shown in Figure 2 may be seen as providing technological support to the 'expert' role in the center of the figure. The system developer operates in the same environment and Figure 2 may be restructured to show the sources available to the developer of a knowledge-based system. Figure 3 illustrates the knowledge acquisition process with the system developer drawing upon three major sources: knowledge encoded in *relevant media*, such as books, journals and videotapes; knowledge available by observing and modeling the *relevant domain*; and knowledge available from discourse with, and observation of, the *relevant community*. It is this last source that is most associated with expert systems as such, although all three sources play significant roles in the development of any practical system.

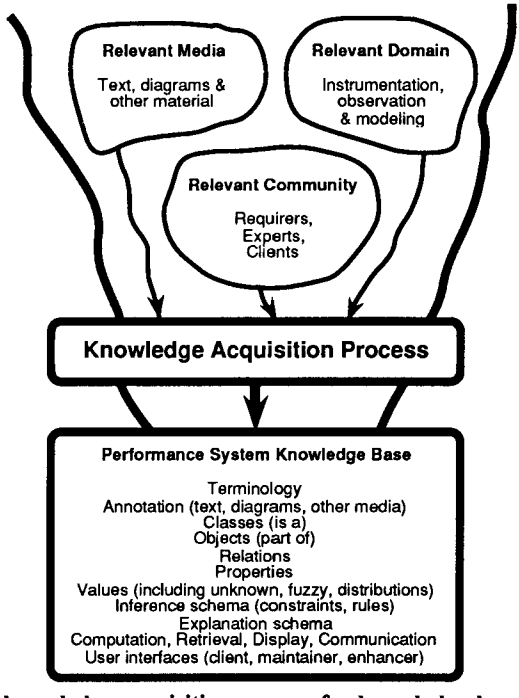

Fig.3 The knowledge acquisition process for knowledge-based systems

Note that the *relevant community* has been split into requirers, experts and clients. Again, domain 'experts' are the expected source of knowledge for an expert system, but detailed requirements specifications and involvement of end-users play as significant a role as they do in other computing system design. It is useful to think of those with these different roles as being all 'experts' in some different aspect of the system functioning, since the knowledge acquisition techniques used with domain experts are often also well-suited to establishing requirements specifications and end-user perspectives.

The annotation of the *knowledge base* in Figure 3 illustrates the range of technical features which an adequate knowledge-based system shell may be expected to support:

- *Terminology* is often neglected as computationally significant. However, it is particularly important in knowledge-based systems where it is critically important that the terms used in questions, recommendations and explanations are understood by clients in the same way as they are by experts. Obtaining the terminology direct from experts without distortion is an important design consideration for knowledge acquisition systems.
- *Annotation* is used here as a term for non-computational knowledge such as text, diagrams, photographs, videos, and other media, that may be used to impart knowledge to people. Some of it may be suitable for computer storage but not for inferential processing. Providing appropriate access to such knowledge is important in many knowledge-based systems, and collecting and organizing such knowledge is part of the knowledge acquisition process.
- *Classes, objects, relations, properties* and *values* form the abstract infrastructure of computational knowledge representation. Object-oriented knowledge-bases provide a formal conceptual and computational schema which subsumes and supports a wide variety of knowledge representation schema. These primitives also correspond to natural cognitive operations of classifying, instantiating, relating, attributing and measuring.

Where knowledge representation goes beyond conventional data representation is largely in the variety of types of value that must be supported. In a data base, a numeric field may be expected to have a clearly defined value. In a knowledge base, because knowledge is 'less than truth', it may be necessary to allow for the value being unknown, constrained by a fuzzy hedge, a range, a distribution, or some combination of these. There may also be mutual constraints between such ill-defined values.

- *Inference schema* are systems of constraints that express the relations between values in a knowledge base. These constraints may be thought of as rules expressing the relations and many of them may necessarily be expressed by explicit rules in the knowledge-base. However, the effective organization of the knowledge base in terms of generic classes and relationships should allow the majority of the constraints to be expressed implicitly in a natural way. This is the *framing problem* in knowledge representation for a domain, and an important aspect of knowledge acquisition is to determine a representation schema that minimizes the number of low level rules required.
- *Explanation schema* are forms of argument that generate satisfactory explanations of the inferences made from a knowledge base. Early expert systems generated explanations by displaying the chain of reasoning that was used from premises to conclusions. This approach is useful but does not necessarily 'explain' the conclusions in a way that an expert would, or in a way satisfying a client's need for knowledge. The acquisition of knowledge necessary for adequate explanation goes beyond that adequate for performance.
- *Computation, retrieval, display, communication* and other data processing activities play major roles in most knowledge-based systems. Modern expert system shells make provision for integration with data-processing, simulation, database, graphics and communications programs. The systems analysis for these needs to be integrated with the knowledge acquisition process.

- *User interfaces* are very important in knowledge-based systems as they are generally highly interactive and involve clients with little or no computer experience. The design of the interface in consultation with experts and clients is an important part of knowledge acquisition. Since most knowledge is fairly volatile, knowledge acquisition and updating the knowledge base are important aspects of system operation, and the interface to system maintainers and enhancers also needs careful attention.

The emphasis on the complexity of the social processes underlying knowledge creation, dissemination and application in society, and the need to treat knowledge-based systems as an integral part of information technology, makes the development of such systems appear somewhat daunting. And so it is—it is easy to underestimate the effort required for the development of effective 'expert systems.' However, knowledge acquisition for knowledge-based systems has also tracked other trends in information technology, notably the use of the computer itself to support the system designer through computer-aided software engineering (CASE) [19]. The following section describes these trends and their effects on the role of the 'knowledge engineer.'

3 KNOWLEDGE ENGINEERING

Knowledge acquisition for practical system development has come to be termed *knowledge engineering*, following Feigenbaum's [14] use of the term to describe the reduction of a large body of knowledge to a precise set of facts and rules. The term *knowledge engineer* has come to be used for the person responsible for such system development, and concise job descriptions for knowledge engineers have been given:

> "Knowledge acquisition is a bottleneck in the construction of expert systems. The knowledge engineer's job is to act as a go-between to help an expert build a system. Since the knowledge engineer has far less knowledge of the domain than the expert, however, communication problems impede the process of transferring expertise into a program. The vocabulary initially used by the expert to talk about the domain with a novice is often inadequate for problem-solving; thus the knowledge engineer and expert must work together to extend and refine it. One of the most difficult aspects of the knowledge engineer's task is helping the expert to structure the domain knowledge, to identify and formalize the domain concepts." [27]

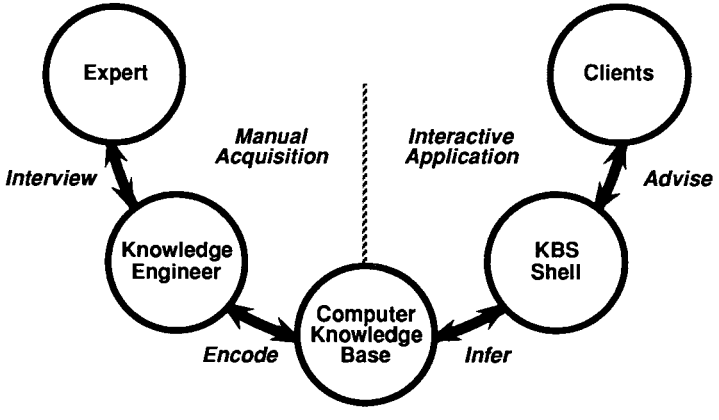

Fig.4 Manual knowledge acquisition by a knowledge engineer interviewing an expert

Thus, 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 4 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:

- The knowledge engineer interviews the expert to elicit his or her knowledge;
- The knowledge engineer encodes the elicited knowledge for the knowledge base;
- The shell uses the knowledge base to make inferences about a particular case;
- Clients use the shell to obtain advice about particular cases.

It seemed possible that knowledge engineering might develop as a profession on a par with systems analysis and programing, and that a major shortage of skilled knowledge engineers would cause problems initially that would be overcome eventually as the profession developed. However, this scenario now appears less and less likely. There is certainly a shortage of knowledge engineers, and major problems in developing applications, but doubts have been cast on the notion that human labor is the appropriate solution to the knowledge engineering problem:

- The decline in costs of both hardware and software support for expert systems has brought the technology into a mass-market situation far more rapidly than originally envisioned;
- This has lead to a growth in demand for expert systems that is proceeding far more rapidly than the growth in supply of trained and experienced knowledge engineers;
- The declining costs of expert system technology are also making the expense of human labor in tailoring the technology for particular applications appear to be the dominating constraint and an excessive cost;
- A move towards a labor-intensive activity such as knowledge engineering is contrary to all trends in industry;
- In particular, it is contrary to the trend towards automatic programing techniques in the computing industry;
- The role of the knowledge engineer as an intermediary between the expert and the technology is being questioned not only on cost grounds but also in relation to its effectiveness—knowledge may be lost through the intermediary and the expert's lack of knowledge of the technology may be less of a detriment than the knowledge engineer's lack of domain knowledge.

These considerations aroused interest in the possibility of providing computer-based knowledge elicitation systems to automate knowledge engineering as a process of direct interaction between domain experts and the computer [7, 15]. The tools and techniques used in such systems are described in later sections. Figure 5 shows how the role of the knowledge engineer has changed as such systems have come into use.

Interactive knowledge acquisition and encoding 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 5, knowledge engineers have responsibility for:

- Advising the experts on the process of interactive knowledge elicitation;
- Managing the interactive knowledge acquisition tools, setting them up appropriately;
- Editing the unencoded knowledge base in collaboration with the experts;
- Managing the knowledge encoding tools, setting them up appropriately;
- Editing the encoded knowledge base in collaboration with the experts;

- Validating the application of the knowledge base in collaboration with the experts;
- Setting up the user interface in collaboration with the experts and clients;
- Training the clients in the effective use of the knowledge base in collaboration with the expert by developing operational and training procedures.

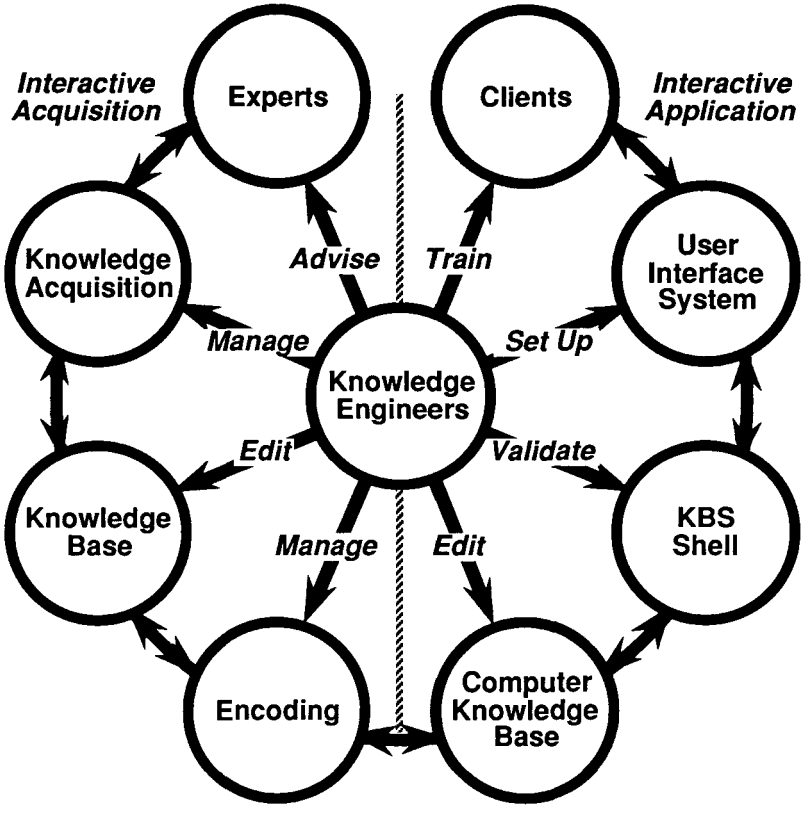

Fig.5 The knowledge engineers' roles using automated knowledge acquisition tools

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

- Directly elicit knowledge from the expert;
- Use the interactive elicitation tools to enter knowledge into the knowledge base.

One of the most important developments in knowledge engineering in recent years has been the development of general domain and task ontologies that provide widely applicable generic models for knowledge engineering [10, 31]. These are the basis of general knowledge-based systems analysis techniques such as those of KADS [9]. The applicability of such generic models reflects the perspective presented by Clancey [11] that *all* knowledge-based systems are essentially models at some level.

4 THE KNOWLEDGE-BASED SYSTEM VIRTUAL MACHINE HIERARCHY

The quality of the knowledge-based system support environment is very important in the development of expert systems, particularly in supporting the interaction between knowledge engineers and experts. The relation of this environment to knowledge acquisition is best seen by putting both in the context of the complete virtual machine hierarchy for knowledge-based systems. Figure 6 shows significant levels of this hierarchy:

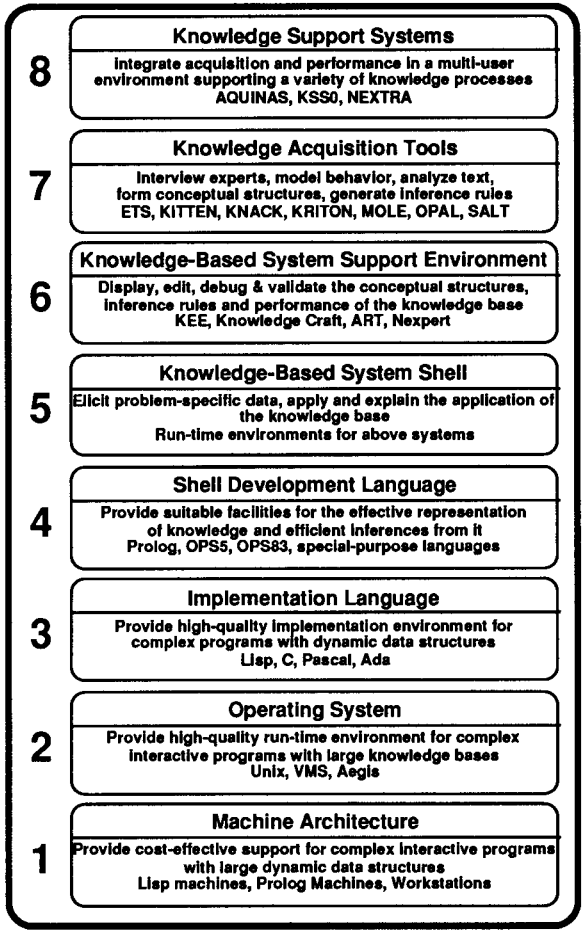

Fig.6 The virtual machine hierarchy for knowledge-based systems

- *Machine Architecture*: at the lowest level of the hierarchy is the machine on which the knowledge-based system runs. This is really irrelevant, but use of Lisp in the early days of expert systems has focused attention on Lisp machines. Nowadays any good workstation is adequate.

- *Operating System*: at the second level of the hierarchy is the operating system within which the implementation runs. This needs to provide good interfaces to other programs, large databases and communications.
- *Implementation Language*: at the third level of the hierarchy is the implementation language which actually interfaces to the computer. This tended to be Lisp in the early days of expert systems but as speed and space efficiency have become significant, and knowledge representation has become better understood, other languages that support dynamic data structures such as C and Pascal have become widely used.
- *Knowledge-Based System Shell Development Language*: at the fourth level of the hierarchy is the language in which the knowledge-based system shell is written, generally a special-purpose environment for coping with knowledge representation and inference.
- *Knowledge-Based System Shell*: at the fifth level of the hierarchy is the expert system shell as a run-time environment that elicits problem-specific information from the user, provides advice based on its knowledge base, and explains that advice in as much detail as required.
- *Knowledge-Based System Support Environment*: at the sixth level of the hierarchy is the equivalent of the Application Programing Support Environment (APSE) in conventional systems. The Knowledge-Based System Support Environment provides editing, browsing and debugging tools for the knowledge base. This level encompasses the major differences between the high-quality knowledge-based system shells that support major system development through extensive support tools, and the simpler shells that do not.
- *Knowledge Acquisition Tools*: at the seventh level of the hierarchy are the tools for automating the systems analysis for expert systems, through automatic interview procedures, modeling actual expert behavior, and analysis of knowledge in textual form. Automation of systems analysis is more significant for knowledge-based systems than for conventional data processing because they tend to be targeted on narrow areas of in-depth specialist knowledge, rather than broad general problems. In conventional terms, knowledge-based systems are satisfying the requirement for highly customized data processing based on idiosyncratic operations rather than general algorithms.
- *Knowledge Support Systems*: at the top of the hierarchy represent the objectives of recent research in knowledge acquisition that focuses on the integration of acquisition and performance systems to provide an environment supporting a wide variety of knowledge processes.

The hierarchy of Figure 6 highlights some of the problem that system managers often face in establishing development capabilities for knowledge-based—should they use Lisp machines?—should they train their staff in Lisp or Prolog?—how does an expert system shell compare with existing programing languages?—what is knowledge engineering and how does it differ from systems analysis? The problem is that knowledge-based systems are at the top of the virtual machine hierarchy which has developed in computing, and there are many layers of technology between a knowledge-based system and the computer on which it is running. In theory, system developers should not need to know about the lower levels of the hierarchy—machine architectures, operating systems and implementation languages are remote from knowledge processing. However, in practice, these lower levels are the foundations on which systems are built and any defects in them can undermine the functionality of the upper levels.

The virtual machine hierarchy for computing is very complex with many branches. Figure 6 focuses on a path through the hierarchy that links the basic hardware and software to knowledge-based systems and knowledge acquisition issues. It omits other parts of the hierarchy that illustrate the integration issues discussed previously. Figure 7 shows some of these issues within a hierarchical framework [21].

The split between users and technology at the top of Figure 7 emphasizes the importance of effective person-computer interaction in knowledge-based systems. As with much modern information technology, usability considerations dominate effectiveness—if the interface is poor no amount of advanced technology in the rest of the system can compensate.

The tree below this to the top left depicts the distinctions between user roles and, in particular, the tasks of the knowledge engineer already discussed.

The tree to the top right first splits the technology between knowledge-based and conventional components, depicting some of the integration issues already discussed, and then further splits the knowledge-based components into support environment, shell, and acquisition as already discussed.

The remainder of this paper focuses first on the sub-tree to the bottom left, analyzing the knowledge-based system support environment and validation, and then on that to the bottom right, analyzing and exemplifying the methodologies, tools and techniques used in knowledge acquisition.

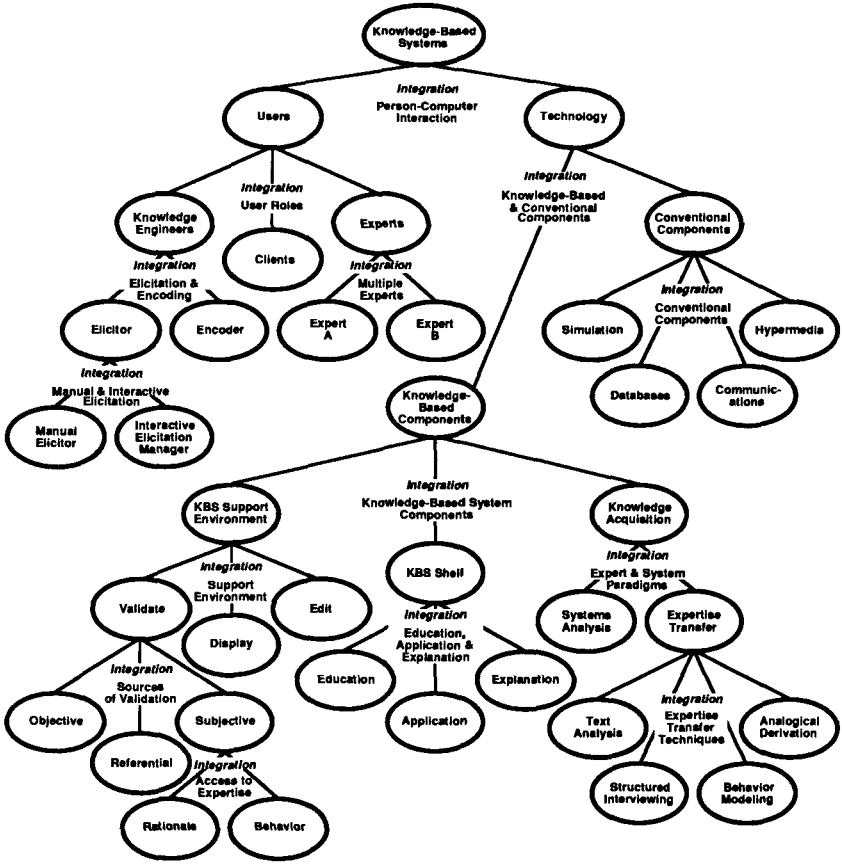

Fig.7 Integration issues in knowledge acquisition for knowledge-based systems

5 EDITING AND VALIDATION

The sub-tree on the lower left of Figure 7 shows some of the major issues in the knowledge-based system support environment at level 6 in Figure 6 in terms of the distinctions involved. 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.

Editing is often regarded as a mundane requirement. However, the variety of related knowledge representations that will usually be present in a knowledge-based system, particularly in the acquisition phase, makes the editing of knowledge bases very difficult to support. If representation A, for example text, is processed to give representation B, for example entities, attributes and values, then the editing of the material in representation A should result in changes to that in representation B, and vice versa. However, the relation between representations is usually not functional but merely constrained, that is knowledge in representation A does not determine that in representation B, and vice versa, but only constrains it in some way as shown in Figure 8. Thus, the result of editing knowledge in one representation does not necessarily result in well-defined changes in other representations, only the detection of constraint violations. The editing system should minimally highlight the violations, but usually it can do more, for example it can adumbrate "likely" changes in other representations that will remove the violation.

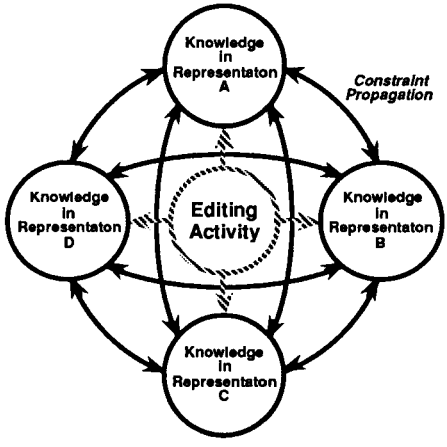

Fig.8 Constraint propagation in editing different representations

Validation is also a complex topic in its own right [1, 40, 41], and it is important that a variety of tools are provided to support different forms of validation. The objective validation of the knowledge base in terms of its performance, its subjective validation in comparison with experts, and its referential validation in comparison with other knowledge sources should be supported. Objective validation may require significant case histories that can be collected only from an operational system in actual applications. Referential validation may require deep analysis of derivations of the knowledge base from other knowledge bases. There is a natural separation in subjective validation between experts' direct agreement that their rationale corresponds to that of the knowledge base, and observed correspondence between their behavior and that of the system based on the knowledge base. Both types of validation should be supported.

The validation of knowledge-based systems has all the dimensions of normal scientific validation together with those concerned specifically with the emulation of expertise [17, 20]. As shown in Figure 9, the system can be validated through four primary pathways:

- Through its correspondence to actual events checked against empirical data;
- Through its efficacy in applications checked against actual applications;
- Through the adequacy of its foundations checked against other disciplines corresponding to the derivation of presuppositions from other disciplines—this is sometimes termed 'deep knowledge';
- Through the comprehensibility of explanations given checked against the relevant reference community—corresponding to explanation and communicative action.

Fig.9 Dimensions of validation of knowledge-based systems

The expert systems paradigm introduces the two additional pathways shown on the right of Figure 9 by taking into account the possibility that there may be "experts" who are competent in applications and explanations:

- Through its correspondence to expert performance checked against the behavior of experts in applications;
- Through its correspondence to expert knowledge checked against the rationale of experts in communicating with their reference community.

This diagram may be iterated a number of times to show the source of the knowledge structures of experts from empirical data and other disciplines, introducing considerations of the validity of experts and reference communities. Similarly, it may be iterated to show the way in which other disciplines in turn are validated against their own empirical data, deeper foundations, applications and reference communities. There is no intrinsic conflict between the additional pathways of validation introduced and the positivist viewpoint of science emphasizing the primacy of empirical data. However, there is a major increase in the realism of this model in modeling the actual processes of knowledge acquisition rather than the normative criteria adopted in 'science' as ultimate arbitrators.

6 THE MAJOR SOURCES FOR KNOWLEDGE ACQUISITION

The significant features of the knowledge acquisition component on the lower right of Figure 10 are best analyzed by considering the natures of the classical and knowledge-based system design paradigms. It is clear that knowledge-based systems use some new computational techniques, but perhaps less apparent that the system design approach involved is itself fundamentally changed. However, major paradigm shifts are involved [23]. The classical approach in decision and control system design is the instrumentation, data collection, modeling and optimization sequence shown on the left of the diagram of Figure 10:

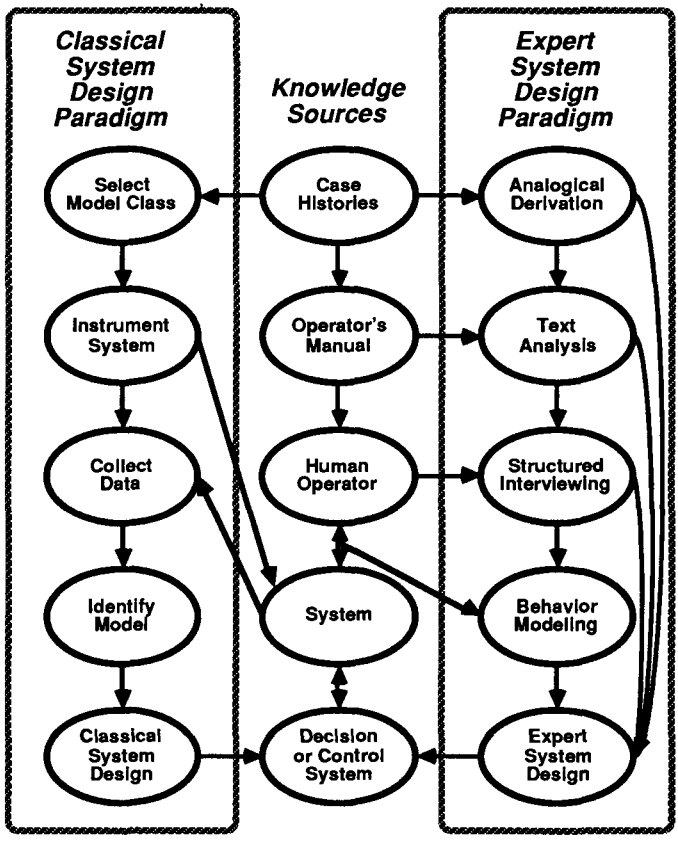

Fig.10 Classical and knowledge-based system design paradigms

- 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.

This approach to system design underlies the methodologies of the physical sciences and technologies based on them. It has the merit that it has been extremely successful in engineering much of the technological infrastructure of our current society including our manufacturing industries. However, this approach is successful only to the extent that the systems under consideration are amenable to instrumentation and modeling. Its greatest successes have been where this amenability can be achieved normatively, that is in cases where the system to be controlled is itself a human artifact. For example, linear systems theory has become a major tool in systems engineering not because most natural systems are linear—they are not. The implication is in the opposite direction: that linear systems are mathematically tractable and that we design artificial systems to be linear so that we may model them readily.

The knowledge-based 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 of Figure 10 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 knowledge-based system design paradigms, and the various acquisition techniques for knowledge-based 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 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.

7 KNOWLEDGE ACQUISITION TOOLS

A number of tools have been developed for the rapid prototyping of knowledge-based systems automating some or all of the methodologies on the right of Figure 10. The objective of systems such as PLANET [36, 37, 38], ETS [3, 4], MOLE [13], SALT [29], KITTEN [39], KNACK [28], KRITON [12], OPAL [33], AQUINAS [6], C4.5 [34] and KSS0 [16,24] is to expedite the process of knowledge acquisition and transfer to knowledge-based systems.

Boose [5] has surveyed a very wide range of knowledge acquisition tools and classified them along a variety of dimensions, analyzing the applicability of tools to tasks. He distinguishes three classes of tasks:-

• Analysis tasks
—Classification: categorizing based on observables;
—Debugging: prescribing remedies for malfunctions;
—Diagnosis: inferring system malfunctions from observable;
—Interpretation: inferring situation descriptions from sensor data.
•Synthesis tasks
—Configuration: configuring collections of objects under constraints in relatively small search spaces;
—Design: configuring collections of objects under constraints in relatively large search spaces;
—Planning: designing actions;
—Scheduling: planning with strong time and/or space constraints.
• Tasks combining analysis and synthesis
—Command and control: ordering and governing overall system control;
—Instruction: diagnosing, debugging and repairing student behavior;
—Monitoring: comparing observations to expected outcomes;
—Prediction: inferring likely consequences of given situations;
—Repair: executing plans to administer prescribed remedies.

Boose notes that most knowledge acquisition tools currently are best suited to analysis tasks and are weak in acquiring expertise in synthesis.

The general requirements for a system to automate knowledge acquisition for knowledge-based systems are [39]:

- The knowledge acquisition system tools should be domain independent;
- The tools should be directly applicable by experts without intermediaries;
- The tools should be able to access a diversity of knowledge sources including text, interviews with experts, and observations of expert behavior;
- The knowledge acquisition system should be able to encompass a diversity of perspectives including partial or contradictory input from different experts;
- The system should be able to encompass a diversity of forms of knowledge and relationships between knowledge;
- The system should be able to present knowledge from a diversity of sources with clarity as to its derivation, consequences and structural relations;
- Users of the knowledge acquisition system should be able to apply the knowledge in a variety of familiar domains and freely experiment with its implications;
- The system should make provision for algorithmic expression of knowledge where appropriate;
- The system should make provision for validation studies;
- As much of the operation of the system as possible should be founded on well-developed and explicit theories of knowledge acquisition, elicitation and representation;
- As the overall knowledge acquisition develops it should converge to an integrated system.

It is particularly important in describing tools that they are presented in the context of user interaction, since their primary function is to operate directly with experts. It is impossible to describe all the knowledge acquisition tools currently in the literature, and the references cited at the beginning of this section and specialist collections should be consulted for details and examples of use [7, 8, 22, 30].

8 CONCLUSIONS: IMPEDIMENTS TO SYSTEM DEVELOPMENT

Expert systems technology, like many other computing developments, has been 'oversold', and the expectations of what may be achieved through the use of knowledge-based system technology have been raised too high. The 'bottleneck' of knowledge engineering discussed in Section 4 still remains. The quotation at the head of that section dates from 1983 and meanwhile major effort has swung into research on knowledge acquisition techniques and there has been a substantial increase in the experience of developing expert systems. Notwithstanding this research and experience, there are still major problems in the development of knowledge-based systems, and it is appropriate to conclude this chapter by summarizing these problems—Figure 11 gives a synopsis of the major impediments to knowledge-based system development.

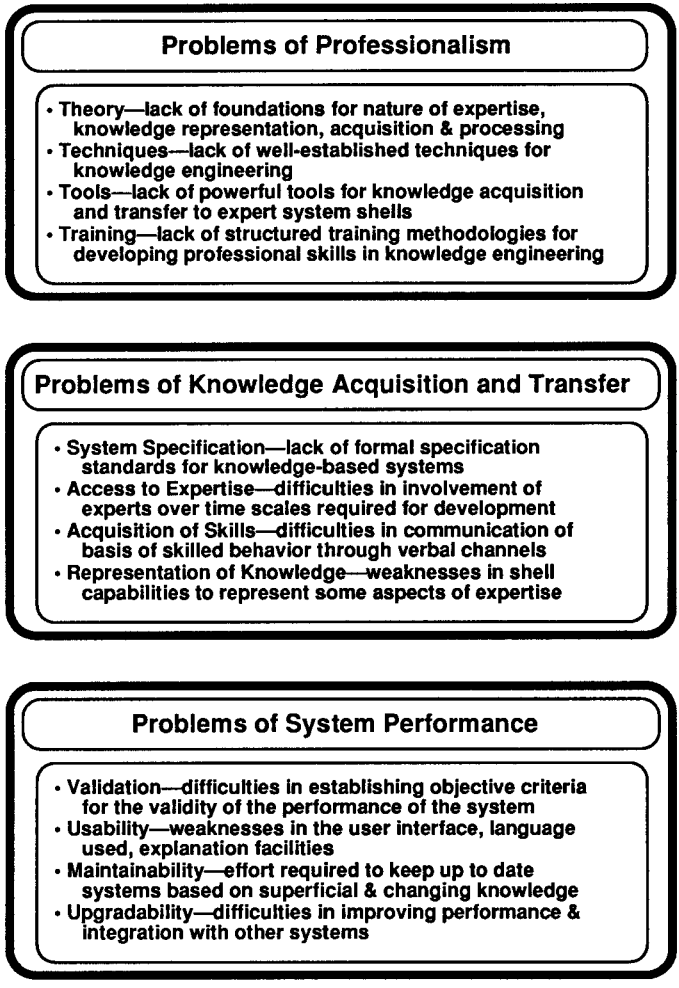

Fig.11 Impediments to knowledge-based system development

The top box in Figure 11 summarizes what may be termed the problems of *professionalism* in developing knowledge-based systems:

- *Theory*: the lack of scientific and engineering foundations for knowledge-based systems is a major impediment to the development of professional standards for expert system development. We understand remarkably little about the nature of expertise even though it one of the most valued commodities in our society. We also have little fundamental understanding of knowledge, the ways in which it is represented in human thought processes, the ways in which it is used in human problem solving, and the ways in which it is transmitted between people.
- *Techniques*: the lack of well-established techniques for knowledge engineering is a major problem for new development teams. There is some folklore, generally mutually contradictory, a few rules of thumb based on experience, but so far no in-depth studies of what is actually done and how successful it has been.
- *Tools*: the lack of commercially available tools for knowledge acquisition and transfer is also a major problem. There are now many research reports of such tools but few of those developed are available as products.
- *Training*: the training situation is improving all the time, largely because it has become the major secondary industry to the supply of expert system shells. A range of courses is being offered by commercial and university organizations. It is still a problem to get good quality courses targeted to specific requirements and on the schedules required.

The middle box in Figure 11 summarizes problems of *knowledge acquisition and transfer* in developing knowledge-based systems. Knowledge acquisition and transfer processes in human society are complex and poorly understood. Neurology, psychology, linguistics, education, sociology, anthropology, philosophy and systems theory all have significant contributions to make. However, integrating them, disambiguating jargon, and combining different objectives and perspectives are major problems. There is a variety of short-term problems arising from these weaknesses in foundations:

- *System Specification*: the formal specification of software that has made software engineering a professional discipline which can ensure quality within well-defined budgets and schedules is notably lacking for expert system developments. We seem at the stage of conventional software development of twenty years ago. This should change as expert system developments move out of the laboratories into the hands of experienced software development groups trained in software management. There is little in the literature to support and expedite this change currently, and we need to develop specification disciplines for expert systems.
- *Access to Expertise*: the problems of getting experts to make the time commitments necessary to the development of expert systems should never be underestimated. Successful projects inevitably seem to have involved the experts in such a way that they become personally committed to the success of the system.
- *Acquisition of Skills*: communicating the basis of skilled performance between people, and between people and computers, is very difficult.
- *Representation of Knowledge*: encoding knowledge within the representation schemes of current shells can present major difficulties. Production rules as used in early expert systems offer very limited facilities for encoding simple knowledge structures at a single level. Frames and their generalization through objects allow conceptual hierarchies to be better expressed. Neither rules nor frames are adequate to encode the procedural or temporal knowledge which is often a key component of expertise.

The bottom box in Figure 11 summarizes problems of *system performance* in developing knowledge-based systems. Rapid prototyping is a significant practical objective and techniques for accelerated initial development without loss of long-term quality are now common in conventional data processing. However, expert system development shows

weaknesses in some areas of performance which we tend to take for granted in standard information technology. There are problems with:

- *Validation*: because we can rarely define and quantify expertise, the validation of expert systems can be very difficult. Do we subjectively validate against the opinions of the expert or do we attempt objective validation of performance against performance standards, accepting that the experts may not themselves satisfy such criteria. How, do we determine whether the knowledge base captures the expert's relevant knowledge, accurately, consistently and completely? There are very few validation studies in the literature.
- *Usability*: the usability of expert systems raises many problems. If we accept the jargon of the expert and the rules of inference based on it, how do we ensure that the operators will use this jargon in the same way as the expert so that the inferences remain valid. It is such considerations that make nonsense of the popular stereotype that access to encoded expertise can make experts of all of us. Understanding the questions, advice and explanations from an expert system generally requires an operator with the same professional background as the expert.
- *Maintainability*: the superficiality of the knowledge encoded in current expert systems raises major problems of maintenance. Keeping the knowledge base up to date may require an ongoing effort at the same level as the original development, a situation very different from most conventional data processing. The usual problems of extensive maintenance not introducing problems of its own are at least as severe for expert systems as for standard data processing.
- *Upgradability*: many expert systems, and the shells supporting them have been designed as stand-alone packages with little scope for integration into large-scale data processing environments. These systems will be difficult to upgrade and integrate with other software.

This emphasis on problems impeding knowledge-based system development is not intended to be negative. In any major technological advance there are significant problems and it is the recognition and classification of these that indicates the growing maturity of the field. There is intensive, world-wide research in all the problem areas noted above, and growing experience in how to overcome them. By recognizing the problems project managers may be able to target their objectives to avoid some, and allocate their resources to overcome others.

ACKNOWLEDGEMENTS

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada. We are grateful to many colleagues for discussions over the years that have influenced this paper. In particular we would like to thank John Boose, Jeff Bradshaw, Alain Rappaport and Brian Woodward for access to their own research and many stimulating discussions.

REFERENCES

[1] Benbasat, I. & Dhaliwal, J.S. (1989) A framework for the validation of knowledge acquisition. **Knowledge Acquisition 1**(2), 215-233 (June).

[2] Blume, S.S. (1974) **Towards a Political Sociology of Science**. New York: Free Press.

[3] Boose, J.H. (1984) Personal construct theory and the transfer of human expertise. **Proceedings AAAI-84**, 27-33. California: American Association for Artificial Intelligence.

[4] Boose, J. H. (1986) **Expertise Transfer for Expert System Design**. New York: Elsevier.

[5] Boose, J. H. (1989) A survey of knowledge acquisition techniques and tools. **Knowledge Acquisition 1**(1), 3-37 (March).

[6] 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),** 3-28.

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

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

[9] Breuker, J. & Wielinga, B. (1987) Use of models in the interpretation of verbal data. Kidd, A. (Ed) **Knowledge Elicitation for Expert Systems: A Practical Handbook.** Plenum.

[10] 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.

[11] Clancey, W. (1989) Viewing knowledge bases as qualitative models. **IEEE Expert, 4**(2), 9-23.

[12] Diederich, J., Ruhmann, I. & May, M. (1987) KRITON: A knowledge acquisition tool for expert systems. **International Journal of Man-Machine Studies, 26**(1), 29-40.

[13] Eshelman, L., Ehret, D., McDermott, J. & Tan, M. (1987) MOLE: A tenacious knowledge acquisition tool. **International Journal of Man-Machine Studies, 26**(1), 41-54

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

[15] Gaines, B.R. (1987) An overview of knowledge acquisition and transfer. **International Journal of Man-Machine Studies 26**(4), 453-472 (April).

[16] Gaines, B.R. (1987) Rapid prototyping for expert systems. Oliff, M.D., Ed. **Intelligent Manufacturing: Proceedings from First International Conference on Expert Systems and the Leading Edge in Production Planning and Control**. pp.45-73. Menlo Park, California, Benjamin Cummins.

[17] Gaines, B.R. (1988) Second generation knowledge acquisition systems. **Proceedings of the Second European Workshop on Knowledge Acquisition for Knowledge-Based Systems (EKAW'88)**. GMD-Studien Nr.143. pp.17-1-17-14. Bonn, Germany: Gellsellschaft für Mathematik und Datenverarbeitung (June).

[18] Gaines, B.R. (1988) Positive feedback processes underlying the formation of expertise. **IEEE Transactions on Systems, Man & Cybernetics SMC-18**(6), 1016-1020 (November).

[19] Gaines, B.R. (1988) Software engineering for knowledge-based systems. **Proceedings of CASE'88, Second International Workshop on Computer-Aided Software Engineering**. pp. 14-1-14-7.

[20] Gaines, B.R. (1989) Social and cognitive processes in knowledge acquisition. **Knowledge Acquisition 1**(1), 39-58 (March).

[21] Gaines, B.R. (1989) Integration issues in knowledge support systems. **International Journal of Man-Machine Studies 31**(5), 495-515 (November).

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

[23] Gaines, B.R. & Shaw, M.L.G. (1985) From fuzzy sets to expert systems. **Information Science 36** (1-2), 5-16 (July).

[24] Gaines, B.R. & Shaw, M.L.G. (1987) Knowledge support systems. **ACM MCC-University Research Symposium.** Austin, Texas: MCC. 47-66.

[25] Hagstrom, W.O. (1965) **The Scientific Community**. New York: Basic Books.

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

[27] Hayes-Roth, F., Waterman, D.A. & Lenat, D.B., Eds. (1983) **Building Expert Systems**. Reading, Massachusetts: Addison-Wesley.

[28] Klinker, G., Bentolila, J., Genetet, S., Grimes, M. & McDermott, J. (1987) KNACK—report-driven knowledge acquisition. **International Journal of Man-Machine Studies, 26**(1), 65-79.

[29] Marcus, S. (1987) Taking backtracking with a grain of SALT. **International Journal of Man-Machine Studies 26**(4), 383-398 (April).

[30] Marcus, S. (Ed.) (1988) **Automating Knowledge Acquisition for Expert Systems**. New York: Kluwer.

[31] McDermott, J. (1988) Preliminary steps toward a taxonomy of problem solving methods. Marcus, S. (Ed) **Automating Knowledge Acquisition for Expert Systems,** Kluwer: Boston.

[32] Merton, R.K. (1973) **The Sociology of Science: Theoretical and Empirical Investigations**. Chicago: University Press.

[33] Musen, M.A. (1989) **Automated Generation of Model-Based Knowledge Acquisition Tools.** London: Pitman.

[34] Quinlan, J.R. (1987) Simplifying decision trees. **International Journal of Man-Machine Studies 27** (3), 221-234 (September).

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

[36] 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.

[37] Shaw, M.L.G. & Gaines, B.R. (1986) Interactive elicitation of knowledge from experts. **Future Computing Systems, 1**(2), 151-190.

[38] Shaw, M.L.G. & Gaines, B.R. (1987) An interactive knowledge elicitation technique using personal construct technology. Kidd, A., Ed. **Knowledge Eliciation for Expert Systems: A Practical Handbook**. pp.109-136. Plenum Press.

[39] Shaw, M.L.G. & Gaines, B.R. (1987) KITTEN: Knowledge Initiation & Transfer Tools for Experts & Novices. **International Journal of Man-Machine Studies 27**(3), 251-280 (September).

[40] 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).

[41] Shaw, M.L.G. & Woodward, J.B. (1988) Validation in a knowledge support system: construing and consistency with multiple experts. **International Journal of Man-Machine Studies, 29,** 329-350.

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