KNOWLEDGE ACQUISITION: SOME FOUNDATIONS, MANUAL METHODS AND FUTURE TRENDS

Mildred L. G. Shaw & Brian R. Gaines
Knowledge Science Institute
University of Calgary, Alberta, Canada T2N 1N4

INTRODUCTION

The problem of expertise elicitation from a skilled person is well-known in the literature of psychology (Nisbett & Wilson 1977, Broadbent, Fitzgerald and Broadbent 1986). Hawkins (1983) has analyzed the nature of expertise and emphasizes its severe limitations and dependence on critical assumptions which are often implicit. Dixon (1981) has surveyed studies showing that much human activity is not accessible to awareness. Collins (1985) has studied knowledge transfer processes among scientists and suggests that some knowledge may not be accessible through the expert, not only because he cannot express it, but also because he may not be aware of its significance to his activity. Bainbridge (1979, 1986) has reviewed the difficulties of verbal debriefing and notes that there is no necessary correlation between verbal reports and mental behavior, and that many psychologists feel strongly that verbal data are useless.

The main problems identified in accessing an expert's knowledge are:
- "Expertise" may be fortuitous. Results obtained may be dependent on features of the situation which the expert is not controlling.
- Expertise may not be available to awareness. An expert may not be able to transmit the expertise by critiquing the performance of others because he is not able to evaluate it.
- Expertise may not be expressible in language. An expert may not be able to transmit the expertise explicitly because he unable to express it.
- Expertise may not be understandable when expressed in language. An apprentice may not be able to understand the language in which the expertise is expressed.
- Expertise may not be applicable even when expressed in language. An apprentice may not be able to convert verbal comprehension of the basis of a skill into skilled performance.
- "Expertise" expressed may be irrelevant. Much of what is learnt, particularly under random reinforcement schedules, is superstitious behavior that neither contributes nor detracts from performance.
- Expertise expressed may incomplete. There will usually be implicit situational dependencies that make explicit expertise inadequate for performance.
- "Expertise" expressed may incorrect. Experts may make explicit statements which do not correspond to their actual behavior and lead to incorrect performance (Gaines 1987a).

WHAT IS 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.

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

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.

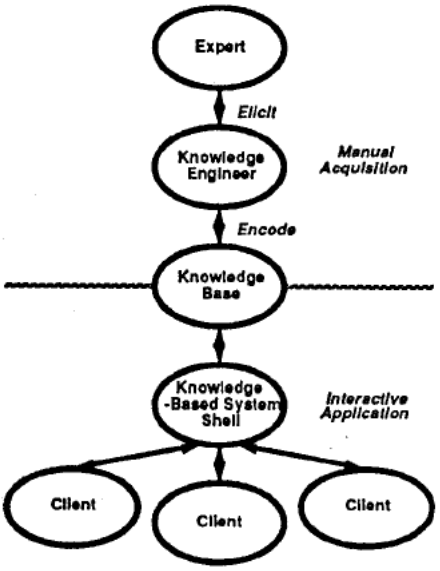

Fig.1 Basic model of manual knowledge acquisition

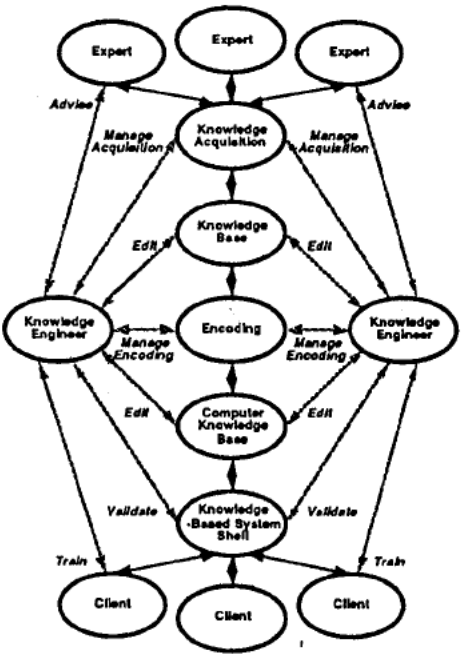

Fig.2 The knowledge engineers' roles in interactive knowledge acquisition

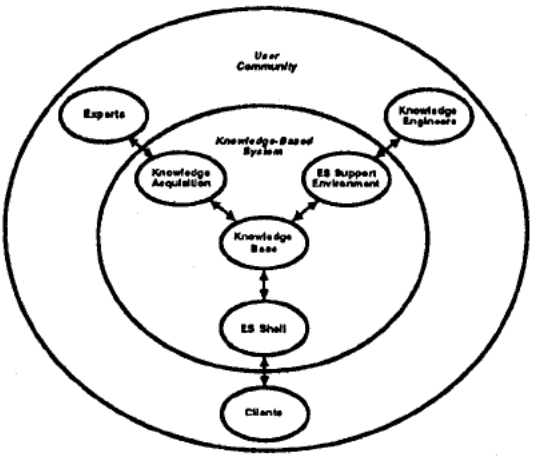

Fig.3 Major components of a knowledge based system

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.

MANUAL METHODS OF KNOWLEDGE ACQUISITION

The process of manual knowledge acquisition may be divided into phases: conceptualization of the domain, acquisition of knowledge and the analysis of results. Further phases in the development of the system include the implementation on a computer and the debugging of the operational system.. Figure 4 shows these phases of knowledge acquisition.

Domain organization involves the identification of functional goals and expectations, and the identification of sub-goals and tasks. In the preliminary knowledge acquisition stage it is necessary to identify knowledge acquisition needs and resources for each, decide how many experts are needed, decide on the number of sessions, how much time is required from each expert, how many knowledge engineers are required, what

types of knowledge acquisition will be used, the preliminary knowledge base design, and an estimate of the size and scope.

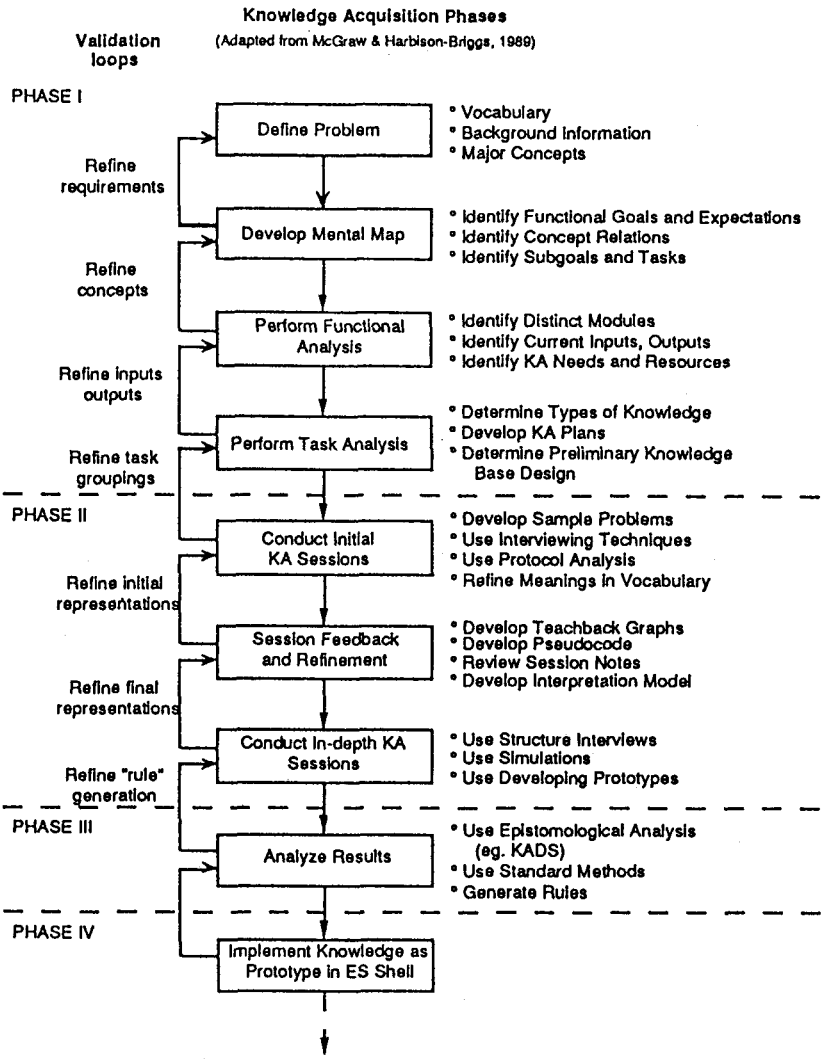

Fig.4 Phases in Knowledge Acquisition

The acquisition of knowledge needs the expert and the knowledge engineer to identify the general heuristics, the procedures, tasks, concepts and vocabulary. It is also necessary to identify the decision-making heuristics, and the analogies and problem-solving techniques used in the domain.

At the pre-implementation phases there are various techniques for domain organization such as generating concepts by brainstorming (van Gundy, 1981), the Crawford Slip method (Rusk and Krone, 1984) or Q-methodology (McKeown & Thomas, 1988), and organizing concepts into groups with the expert identifying common threads and providing labels while the knowledge engineer makes records in the form of lists, notes, diagrams, charts, a knowledge dictionary or glossary using audiotape, videotape, whiteboard or a wordprocessor.

Two examples of tools for domain organization are Cognosys (Woodward, 1988) and Knowledge Acquisition Design System (Breuker & Wielinga, 1987). In Cognosys the expert provides a description statement of overall domain and a list of general objectives which must be met in order to solve problems or make decisions successfully with the domain. He must the produce short paragraphs which are descriptive of the tasks required to meet each objective. These may be articles provided by the expert, such as case studies, text books or operating manuals. The knowledge engineer takes these and analyses the tasks into categories. He also completes the concept description categories and clarifies and elaborates such things as temporal relationships, event and state sequences, goal specification, cause-effect relationships, inferences, hierarchical relationships. It is important at this stage to clarify the domain boundaries with the expert and discuss specific problems in the

domain. He may identify other sources of domain knowledge, before he identifies knowledge acquisition methodologies and tools, report formats and inference processes.

The KADS methodology deals with the whole process from domain organization to developing a complete system. It breaks the domain organization into stages where the problem is decomposed into component parts and the analysis of knowledge is separated from its machine implementation. Data is then collected and tested to modify the framework. The three refinement cycles are applied: orientation which includes the acquisition of vocabulary, the assessment of characteristics, and the selection of the model and a test for problem feasibility; problem identification in which structures of domain concepts are uncovered, the functional analysis of the prospective expert system and task analysis; and problem analysis where the analysis of the user and operational environment results in the basic architecture of the system. At this point knowledge primitives are identified such as objects, knowledge sources, structures, strategies and models. Interpretation models are produced using a top-down model-driven approach which may include notions of causality, time, space, change and task-dependent concepts.

Building interpretation models is done by knowledge identification and knowledge conceptualization where knowledge primitives are ordered using relationships such as is-a, part-of, causes, depends-on, consists-of, defines, and several types of analysis are performed. The system is then coded using guidelines laid down.

SOME TECHNIQUES FOR THE MANUAL ACQUISITION OF KNOWLEDGE

Interviewing (McGraw & Harbison-Briggs, 1989) is used to identify tasks and major concepts, to structure and refine already-acquired information. Unstructured interviews are appropriate when the knowledge engineer wants to explore an issue or the goal is to establish rapport and to get a broad view. It facilitates the description of the domain in a way that is easy for the expert to understand. However, the data acquired is often unrelated and difficult to integrate due to its lack of structure. It may not allow gathering of specific knowledge and takes time and training to do well. Also, similar questions may be asked in future sessions and this may annoy the expert. Alternatively, structured interviews have the advantages of forcing an organization on the interview and being very goal-directed. They attempt to remove distortion from experts subjectively and allow better integration of material after the interview. They force the expert to be systematic, and the knowledge engineer is able to identify gaps in the knowledge which act as a basis for further questions. The purpose of the session is clear to the expert. However, they need more preparation by the knowledge engineer who must study the background material extensively.

During the interview, of any kind, the knowledge engineer must decide on a note taking approach, for example, video- or audio-taping, each of which has advantages and disadvantages. Both open and closed questions can be used. Open questions are broad and place few constraints on the responder. They are used for determining scope of understanding, response certainty, and the models used. These questions like *what, why, how* allow the expert to express information the knowledge engineer does not know about. He can obtain the expert's vocabulary, concepts, frames of reference, and can help with explanations and underlying theory. Closed questions such as *who, where, when, which* tend to set limits on the type, level and amount of information the expert provides since they often provide a choice of alternatives such as a bipolar or multiple choice response. These are used more for clarifying or probing questions or as feedback as they are less time consuming for specific information. Note-taking is much easier but sometimes one can get too little information as they may stop the expert from volunteering information. They requires an excellent command of vocabulary and concepts from the knowledge engineer. A balance of open and closed questions is optimal.

Two examples of semi-structured interviewing techniques are the Knowledge Acquisition Grid (LaFrance, 1987) and the Teachback technique of Johnson and Johnson (1987). In the Knowledge Acquisition Grid a matrix of knowledge types and forms is used:

Question Types	Forms of Knowledge: layouts	stories	scripts	metaphors	rules-of-thumb
grand tour					
cataloging categories					
ascertaining attributes					
determining inter-connections					
seeking advice					
cross-checking					

Examples of a *grand tour* are distinguishing boundaries of the domain, perspective, goals organization, e.g. Please describe the kinds of things you do; of *cataloging* may be an organized taxonomy of expert terms and concepts, e.g. How would you group and subgroup these things?; of *ascertaining attributes* are distinguishing features and range of values, e.g. Describe similarities and differences of concepts; of *determining interconnections* are uncovering relations and causal model(s), e.g. Why?; of *seeking advice* include revealing strategies, e.g. Which conditions warrant which actions?; and of *cross-checking* are validation concerns, e.g. knowledge engineer plays naive, devil's advocate, poses hypothetical situations and asks How sure are you? Are there exceptions?

The Teachback technique is based on Pask's (1975) *conversation theory* and on Systemic Grammar Networks from Linguistics. This method attempts to formalize a functional approach to grammar. It is an intermediate representation used to teach back to the expert the knowledge acquired in the interview. It is a *mediating* representation which is a machine language independent formalism between verbal data and software development environments. It is used as a tool to help proceed on early elicitation and conceptualization for synthesizing knowledge from talk with experts. It is mainly used as a means to prepare explicit descriptions of knowledge for the use of the development team to construct an associative network, graphs composed of nodes joined by links. It is said to be a generative form which captures all relevant information but excludes absurdities.

Two examples of Protocol Analysis, which are widely used, are Verbal Reports as Data (Ericsson & Simon, 1984) and the Specification of Expertise (Johnson, Zaulkernan & Garber, 1987). Ericsson's method is a well-known technique and one of the few established ways to elicit knowledge. It uses concurrent and retrospective verbalization for elicitation. *Concurrent report* reveals the sequence of heeded (attended to) information and *retrospective report* reveals the sequence of specific cognitive processes. A think-aloud transcript is then encoded into segments (assertions propositions, reading, doing arithmetic, visual searching, long-term memory recall, generalizing combinations) and relations of code are derived from verbs, propositions and adjectives. "Arguments" are often nouns and nouns phrases and the task analysis defines *a priori* a space of possible encodings representing information relevant to the task.

The method is based on the facts that:

- heeded (attended to) information is identified by analysis of corresponding verbalization through inference and task analysis
- the original encoding should reflect as closely as possible the verbalized information, preferably by automated or semi-automated means
- encoding uses a problem space approach (behavior is a search through a problem space involving the step-by-step accumulation of knowledge)
- the model distinguishes information directly stored in memory (facts, experiences, perceived events and behavior in past situations) from information generated or produced by inference (reactions and behavior in hypothetical situations)
- the criteria for inferring underlying cognitive processes from thinking aloud protocols ar relevance to task analysis and consistency with preceding verbalizations
- information in focal attention is vocalized directly or is encoded, recovery implies decoding
- segments are analyzed using syntax, pauses and intonation
- each segment is independent (the context is ignored for the reason cited above)
- encoding involves finding a category expressing the same information as the segments

The advantages of this method are that it can provide a complete record using both concurrent and retrospective reports then after definition of problem space and segmentation of the protocol the analysis can be automated or the task made much easier for the coder as it focuses the coder's attention on the segments. It requires consistency and explicitness of underlying assumptions, vocabulary and inference rules and the automated encoding lessens the risk of incorrect inference. A comparison of talking aloud protocols shows a very high consistency for practiced skills and successive protocols show stable task strategies and cognitive representations. It is an appropriate method for reporting specific, recent, episodic memories. On the other hand, retrospective reports may leave out important detail and include inferences and generated information (plausible explanations). It is often difficult to tell if the expert pays attention to the knowledge engineers's instructions. The expert cannot report cues allowing recognition of stimuli since these are in long term memory and his short term memory retains goals and intentions longer than heeded information. There are also incompleteness problems. Thought in non-oral form is much faster, for example, the knowledge engineer cannot instruct a juggler to slow down!

The Specification of Expertise (Johnson, Zaulkernan & Garber, 1987) is one way of using the material gained from protocol analysis. It is a psychology based manual technique in which the expert thinks aloud while working. It encompasses a syntactic and semantic analysis of the protocol using operations, episodes, and data cues. Expertise is characterized by semantic categories, actions abilities, goals, conditions, strategies and solutions. It is based on the facts that thinking is not directly observable, especially that of expert with considerable compiled knowledge and practice and that expertise is characterized by generativity and is a kind of knowledge that is used to perform a task. Part of this knowledge is represented by a set of abilities for problem solving. The representation or specification of expertise consists of: possible solutions and components of solutions; relevant information or data needed; goals for solutions; permissible ways to move between intermediate solutions and the abilities needed to make a transition.

ANALYSIS

Many existing statistical and clustering methods can be applied to the analysis of structured material such as factor analysis, Principal Component analysis or multidimensional scaling techniques. In recent years some

machine learning techniques have been found to be successful such as induction. There is no neat knowledge acquisition method currently for analyzing unstructured interview data. Data is normally long and complex and arranged in a non-standard format, and moreover, expert testimony usually relies on unspecified assumptions, background meanings and tacit knowledge. Pidgeon, Turner & Blockley (1989) have identified some issues that arise:

- Data is qualitative not quantitative
- Behavior and expertise are contextual
- Subjective orientation of the subject matter
- Complexity of expert knowledge
- Critical role of tacit knowledge
- Discovery of theory by testing hypotheses
- Data driven approach to elicitation
- Analysis is an interpretation of the expert's knowledge
- The knowledge engineer is a knowledge acquisition tool
- Elicitation and representation "fit" the phenomenon
- Results reflect and represent the intended meaning

The advice they give is to build as soon as possible after the interview and aim to arrive at a form of labelling which accurately characterizes the concepts, decision rules and strategies of the expert. The steps are:

1. Begin with first paragraph and ask what categories, concepts, or labels are needed in order to account for the phenomena of importance in the paragraph;
2. Record the label on a file card (or hypercard, or outliner) with a short description of the feature, and reference to the paragraph or line number;
3. Repeat for all paragraphs;
4. Recognize and list recurrences of labels;
5. Refine labels to produce appropriate concepts from all occurrences, then complete a written definition of the concept;
6. Definitions should contain descriptions of data patterns, generalizations, new categories, and links between categories;
7. Group sets of related concepts.

This process results in a set of fully saturated core primitives together with definitions, propositions linking categories, and graphical representations of relations among primitives.

FUTURE DEVELOPMENTS AND TRENDS IN KNOWLEDGE ACQUISITION

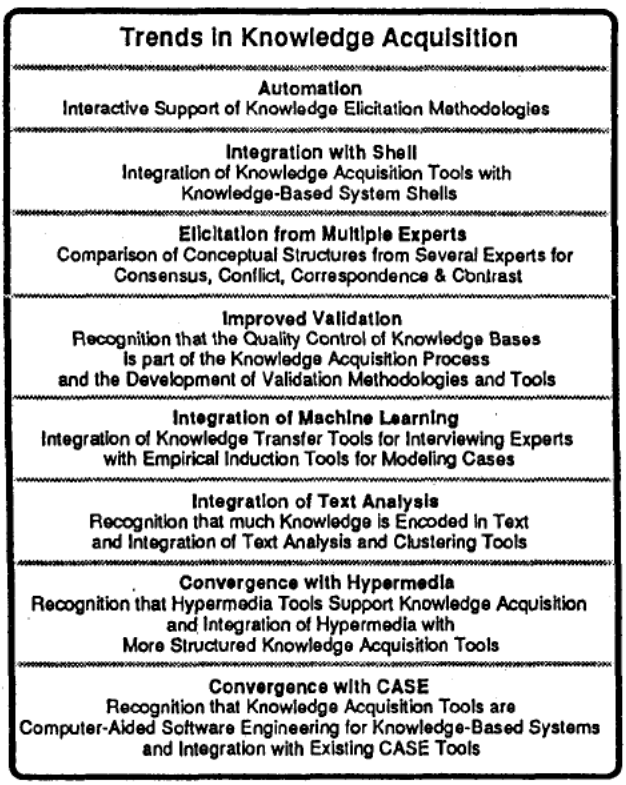

Fig.5 Significant Trends in Knowledge Acquisition

The initial development of knowledge acquisition techniques in the 1970s was based on human interviewers debriefing experts about their skills. The development of computer-based tools supporting knowledge acquisition in the 1980s has been based on the computer-aided application of specific techniques such as construct and conceptual mapping, behavior modeling and text analysis. However, neither manual nor tool-based approaches have yet made a significant impact on the knowledge acquisition bottleneck which continues to impede the development and effective application of knowledge-based systems (Gaines 1987a,b). Current research is attempting to establish a new paradigm for knowledge acquisition, one that recognizes the wide diversity of forms and sources of knowledge, its essential dynamism making it subject to contradiction and change, and the constructive nature of knowledge acquisition whereby it is often the knowledge acquisition process itself that creates knowledge. Figure 5 reflects some of the current trends in knowledge acquisition.

Steels (1986) has suggested that we are entering an era of *second generation* expert systems, "able to combine heuristic reasoning based on rules, with deep reasoning based on a model of the problem domain. This solves a number of major problems of current expert systems, most importantly the problem of knowledge acquisition: second generation expert systems can learn new rules by examining the results of deep reasoning." If one generalizes this concept and thinks of second generation systems as having *embedded knowledge acquisition* using a variety of sources including those suggested by Steels, then it captures the thrust of current research: to integrate knowledge acquisition and performance systems. The zeroth generation of manual knowledge acquisition techniques led to the first generation of computer-aided tools, and this is now leading to a second generation of integrated knowledge support systems.

Gaines (1988a) has suggested that second generation knowledge acquisition targeted on applications in the 1990s have to:

- Integrate a wide variety of weak methods encompassing the diversity of forms and sources of knowledge
- Integrate conflicting sources of knowledge to cope with essential contradictions that cannot be replaced by an artificial and misleading 'consensus'
- Integrate knowledge acquisition with system operation to facilitate a heavy activity of continuing system maintenance and upgrade as knowledge changes
- Integrate knowledge representation in the acquisition and performance systems, retaining the flexibility for new representations to be used that more accurately carry the knowledge

Thus, second generation developments have two objectives that may come into conflict:

- To integrate closely a variety of techniques and technologies to cope with the diversity of knowledge
- To provide flexibility within the integration framework for new techniques and technologies to be incorporated to cope with the dynamics of knowledge

AUTOMATION: KNOWLEDGE ACQUISITION AS INFORMATION TECHNOLOGY

To examine the technological support for second generation integrated knowledge acquisition systems it is necessary to have some model of the wide range of technologies underlying computing. The computer itself is only one component in an elaborate and highly differentiated infrastructure which has grown through a succession of generations of computers, each of which represents a major change in computing technology (Gaines & Shaw 1986). During the eight year span of each computing generation, revolutionary changes have taken place that correspond in magnitude to those taking place over some hundred years or more in other industries. The first generation commenced in 1948 and we are now in the sixth generation spanning 1988 through 1995.

The generation structure of information technology can itself be analyzed in terms of the knowledge flows involved. Like other basic technologies, the growth of computing has an infrastructure comprised of the envelope of learning curves in successive underlying technologies. Figure 6 shows this structure through the zeroth through fifth generations projected to the sixth and seventh. Each technology has a learning curve in which a *breakthrough* leads to a phase of: *replication* in which the results are duplicated; *empiricism* in which rules for design are derived from experience; *theory* in which generative principles are derived; *automation* in which the theory is made operational; and *maturity* when the knowledge becomes proven, widely available and widely used. The qualitative changes that have given computing its distinct character in each generation appear through the tiered succession of learning curves of higher level technologies based on the lower level developments:

- the initial breakthrough in *electronic device technology* leading to the zeroth generation is placed at 1940 about the time of the Atanasoff and Berry experiments with tube-based digital calculations
- the first generation breakthrough was the introduction by Mauchly and Von Neumann of stored program and subroutine concepts around 1947 which led to the transition from the ENIAC to the EDVAC designs—this detached computing as a separate discipline from electronics by substituting software for hardware in a *virtual machine architecture*
- the second generation breakthrough was to bridge the gap between machine and task through the development of *problem-oriented languages* such as FORTRAN around 1956
- the third generation breakthrough was to bridge the gap between the computer and the person with the development of interactive time-shared computers in 1964 allowing close *human-computer interaction*
- the fourth generation breakthrough was in the early 1970s with developments in expert systems based on knowledge-processing—such *knowledge-based systems* have the capability to store information through its inter-relations and make inferences about its consequence
- the fifth generation breakthrough was in 1980 with developments in machine learning and *inductive inference systems*
- the growth of robotics will provide the next breakthroughs in which goal-directed, mobile computational systems will act autonomously to achieve their objectives—the breakthrough into the sixth generation era commencing in 1988 is one of *autonomous activity systems*

• the interaction between these systems will become increasingly important in enabling them to cooperate to achieve goals—the seventh generation era commencing in 1996 will be one of *socially organized systems*.

The generations of knowledge acquisition systems are shown at the top of Figure 6—the current sixth generation of information technology is also marked. It coincides with the second generation of knowledge acquisition systems and this four generation gap indicates an analogy between the state of knowledge-based systems development now and that of data processing 32 years ago. We do indeed have a similar situation to the 1950s in computing when software development was new with problems of shortage of skills, unrealistic schedules and budget over-runs.

The detailed infrastructure for the development of the second generation of knowledge acquisition systems may also be seen in Figure 6. At the three lowest levels, the hardware machines architectures and languages are all mature technologies posing no problems. Human-computer interaction has entered its phase of automation with the increasing availability of machine-independent interactive systems such as X-windows. Knowledge-based systems have entered the theoretical development phase with increasing depth and structure to knowledge representation and inferencing schema. Inductive inference systems have entered the empirical phase where there are beginning to be available a range of usable tools. Autonomous activity systems are still in the research phase and the lack of experience and foundations will restrict the development of knowledge acquisition systems for goal-directed, planning activities during this generation.

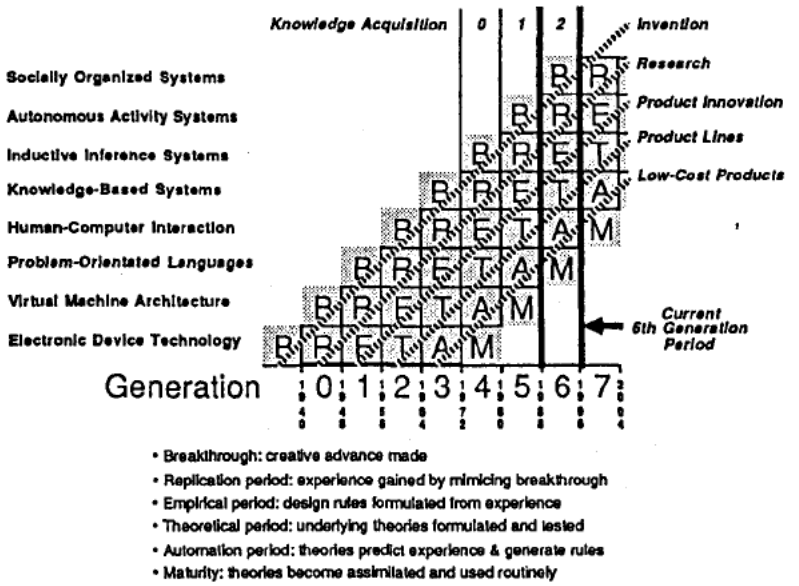

Fig.6 Second Generation Knowledge Acquisition Systems in the Infrastructure of Information Technology

Thus overall, we should expect the emphasis in second generation knowledge acquisition systems to be on the use of high-speed graphic workstations, programmed in high-level languages, using high-quality user interfaces to provide effective knowledge representation, some inductive inference and little or no capability to deal with planning.

INTEGRATION: KNOWLEDGE ACQUISITION AS PART OF A SOCIAL PROCESS

Hawkins (1983) has abstracted from industrial experience in developing mineral exploration expert systems and proposed a model of human expertise relevant to expert systems. Figure 7 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. 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. As Figure 7 shows, this implies that the expert 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 8 extends the model of Figure 7 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.

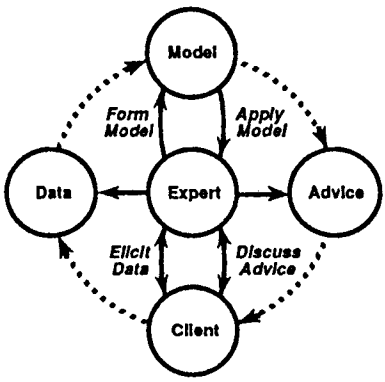

Fig.7 The negotiation cycle in expert-client interaction

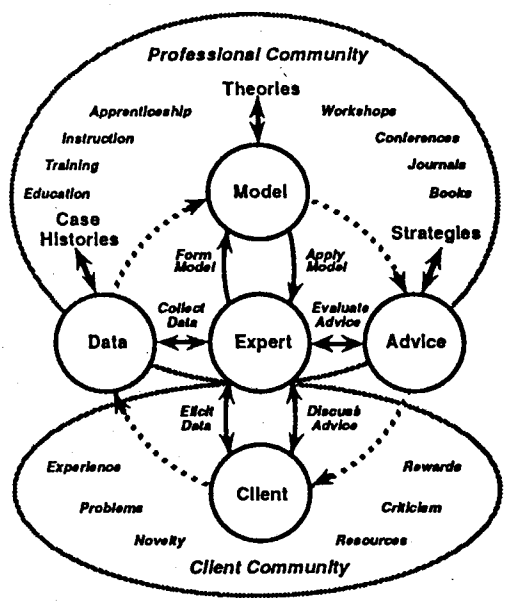

Fig.8 Extended knowledge acquisition process in expert-community interaction

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.

ELICITATION FROM MULTIPLE EXPERTS

The knowledge to be acquired for the development of knowledge based systems is often distributed across a group of experts rather than available for elicitation from a single expert . Group elicitation presents major problems because experts can disagree on the use of concepts and vocabulary, and this disagreement may be tacit causing confusion. Methodologies and tools are being developed for knowledge acquisition from groups in which the conceptual frameworks of different experts are compared in a way that makes such disagreements overt and readily identifiable (Shaw & Woodward 1987, Shaw & Gaines 1988). Studies have also been made of extending performance systems based on group elicitation to allow multiple knowledge structures from different experts to be combined without distortion, enabling "dissenting opinions" to be requested (Boose 1986).

Apparent dissent between experts may be due to conflict in using the same terminology for different concepts. In addition apparent difference in viewpoints can result from the use of different terminology for the same concept. These sources of dissent between experts apply equally to the conceptual frameworks of experts and clients. Figures 9 and 10 below illustrate some of the phenomena that can arise by considering the mappings from the concepts of experts or clients to the relevant *distinction* in the domain and the *term* used for that distinction.

In Figure 9: *consensus* arises if the conceptual systems of the two individuals assign the same term to the same distinction; *conflict* arises if they assign the same term to different distinctions. In Figure 10: *correspondence* arises if the conceptual systems of the two individuals assign different terms to the same distinction; *contrast* arises as they assign different terms to different distinctions. *Conflict* shows possible dissent over the application of a concept in a domain since the same term is being used in different ways. *Correspondence* shows possible dissent over the appropriate term to be used for a concept. *Consensus* is particularly important

in showing a shared conceptual sub-system. *Contrast* shows conceptual sub-systems that are not shared but do not lead to dissent.

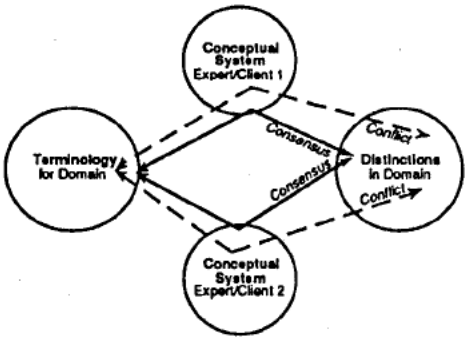

Fig.9 Consensus and conflict between conceptual systems

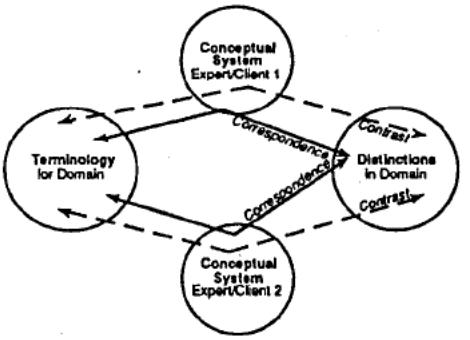

Fig.10 Correspondence and contrast between conceptual systems

Figure 11 is a graphic presentation of a consensus and conflict analysis from a grid exchanged between two experts in geographic mapping techniques. The differences are shown in the grid on the left in numeric form and as shading with darker gray indicating greater difference. The sets of attributes and entities have both been sorted so that those with highest matches indicating greatest consensus are at the top of their respective lists. The graphs on the right show the declining match values and may be used to decide where to place thresholds distinguishing consensus and conflict. Most importantly, this presentation contains sufficient raw and processed data that it may be used for a range of discussions, from simple identification of apparent disagreements, such as the experts disagree on the use of the term "linear interpolation" and the entity "probability matching," to a detailed analysis of how these disagreements have arisen. Measures of correspondence and contrast are presented in a similar graphical form with attributes paired from the two grids.

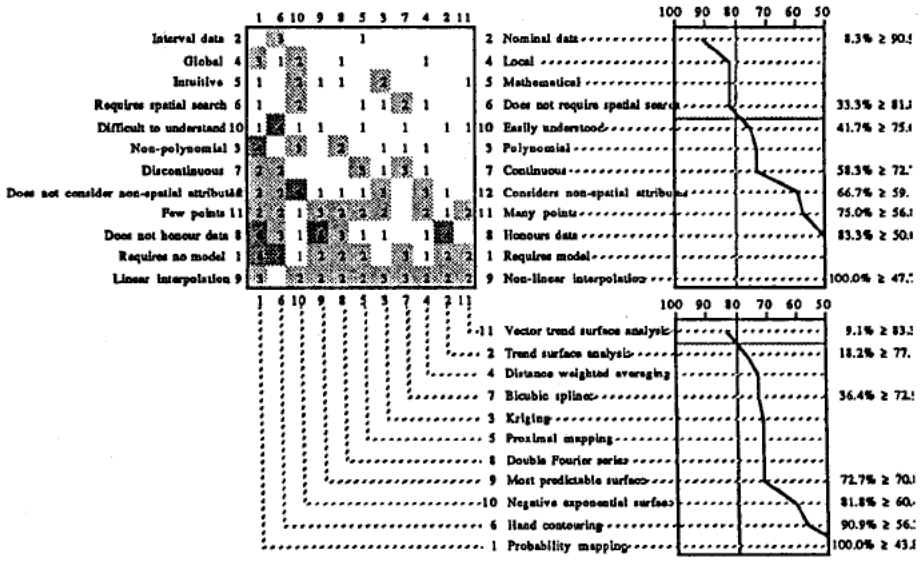

Fig.11 Presentation of measured consensus and conflict

The results of the comparisons are summarized and presented in the four quadrant diagram exemplified in Figure 12. The *consensual* items are simple to deal with—an obvious question is "to what extent can a knowledge-based system be built based only on these well-agreed distinctions?" The *conflicting* items can be looked at with the reverse implication, "to what extent can a knowledge-based system be built not using these ill-defined distinctions?" If some of the distinctions are agreed to be necessary for system development then the basis of the conflict can be examined in greater detail starting with the previous presentation. Many resolutions of the conflict are possible, but the computer-based methodology can go no further than making the basis of conflict and the possible need for resolution apparent.

The *corresponding* items may again be looked at in terms of their relevance to the problem. In addition their further interpretation often leads to significant insights into the knowledge structures involved. For example, the terms *low-level-data—high-level data* and *nominal data—interval or ratio data* corresponding may be interpreted as arising from different *levels of abstraction*. Questions then arise about whether the levels should be represented in the knowledge structures, "which will be best understood by a client?", and "should one term

be used as an explanation of the other in response to a *help* request?" Interpretation of the correspondences in the Figure is an important knowledge acquisition methodology in its own right.

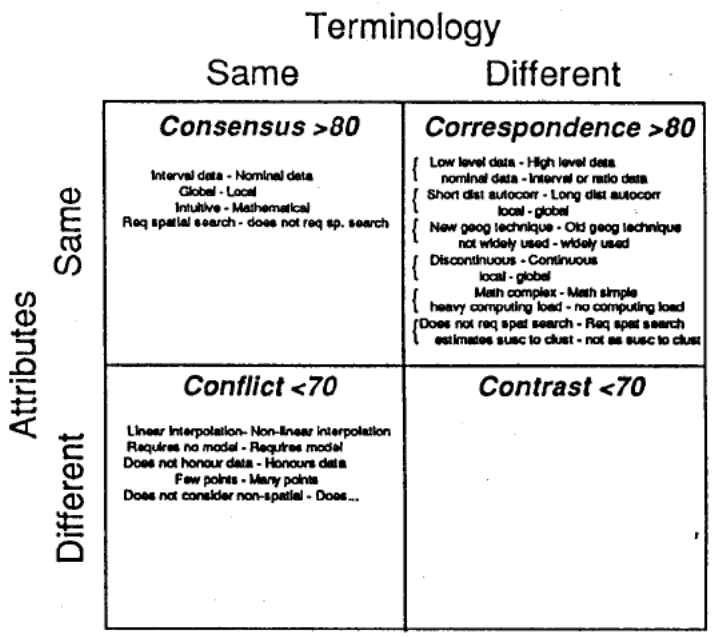

Fig.12 Summary of comparisons between two experts

The simple example shown brings to light abstraction hierarchies, physical and social causal links, and other relations between attributes. Such methodologies have the advantages of group *brainstorming* in stimulating knowledge elicitation, and capturing the results in a highly structured and usable form. Future knowledge acquisition tools will make increasing provision for such inter-expert comparisons.

IMPROVED VALIDATION

The validation of knowledge-based systems is a complex topic in its own right, and a variety of techniques and tools are being developed to support different forms of validation (Shaw & Woodward 1987 Benbasat & Dhaliwal 1988). 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 all 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 (Gaines 1988c). As shown in Figure 13, 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.

The expert systems paradigm introduces the two additional pathways shown on the right of Figure 13 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.

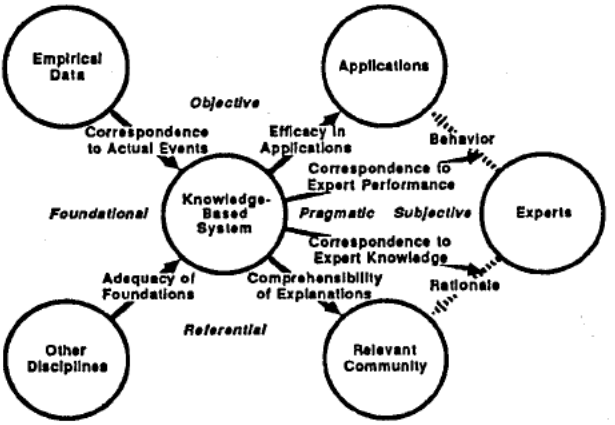

Fig.13 Dimensions of validation of knowledge-based systems

INTEGRATION OF MACHINE LEARNING

This tutorial is primarily concerned with *transfer* of knowledge from existing sources, such as experts and books, rather than *learning* of knowledge from experience. *Machine learning* is an important aspect of knowledge-based systems, but it has a major literature of its own (Michalski & Carbonell 1983, Michalski, Carbonell & Mitchell 1986) which is in large part distinct from issues of knowledge and expertise transfer from existing sources. In a broad sense, learning and transfer processes, are both part of 'knowledge acquisition' for knowledge-based systems, and have many issues in common, particularly those of knowledge representation and inductive modeling. However, most practical knowledge-based systems currently acquire their knowledge base through the transfer of existent knowledge rather than learning from experience, and a major literature has developed about such transfer, its problems, and tools and techniques to overcome them.

Thus, there is currently a major paradigm split in knowledge acquisition research and practice between techniques for the transfer of existing knowledge from human experts and those for the creation of new expertise through machine learning. There is, however, a fundamental relation between the two paradigms in that existing expertise was at some time derived through empirical induction. There is also continuity between the two paradigms in that existing expertise may be partial, erroneous, and of various forms, such that it cannot completely replace empirical induction but may serve to to guide and expedite it. For example human experts in a decision task might know: at one extreme, a complete, minimal set of correct decision rules; through a critical set of cases described in terms of a minimal set of relevant attributes with correct decisions; to, at the other extreme, a source of cases described in terms of attributes among which are those relevant to the decision and with only a greater than chance probability of correct decisions. This is a sequence of decreasing knowledge on the part of the human expert.

Gaines (1989) has studied how the amount of knowledge available from the expert affects the amount of data required for effective empirical induction in the range of situations above. He took a set of cases originally defined by Cendrowska (1987) as a test of empirical induction and generated a range of datasets from it by making incorrect decisions with known probability and adding a prescribed number of irrelevant attributes with random values. He then applied a powerful empirical induction algorithm, INDUCT, to the datasets to determine the average amount of data required to obtain a correctly performing set of rules. The original dataset can be correctly modeled with 6 rules involving 4 attributes.

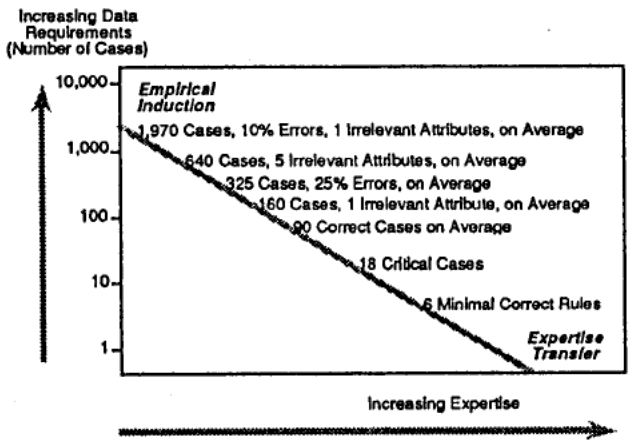

Fig.14 The Trade Off Between Data and Knowledge (Gaines 1989)

Figure 14 shows some of Gaines' results as a plot of the data required on average for empirical induction against the type of knowledge available from the simulated expert. The trade off between data and expertise is clearly shown, and what is also apparent is the continuum between dependence on empirical induction from cases on the left and dependence on transfer of expert knowledge on the right. It is particularly noteworthy that a single algorithm was used for empirical induction across the range of situations Gaines investigated. The results validate the claims of those with tools aimed primarily at eliciting relevant attributes and critical cases from experts that this is an efficient approach. They also show that empirical induction and expertise transfer can be combined to produce a more powerful and versatile acquisition tool than either alone. In future we can expect methodologies and tools to increasingly support a combination of techniques.

INTEGRATION OF TEXT ANALYSIS

A very significant component of human knowledge has already been expressed in text, diagrams and pictures and captured in books and journals. The extraction of knowledge from such material is a major research area in its own right which has major relevance to knowledge acquisition and may be expected to produce methodologies and tools that will be integrated with those described in this tutorial.

One class of research in concerned with the detailed analysis of knowledge structures in text. The CYC project at MCC is a notable example of an attempt to build a knowledge base from an encyclopaedia (Lenat, Prakash & Shepherd 1986). Slator (1988) has described a system for producing text-specific lexicons from selected machine-readable dictionary definitions. Szpakowicz (1988) has described a system that semi-automatically processes technical text and incrementally builds a conceptual model of a domain using Sowa's (1984) conceptual graphs. More simplistic forms of text analysis have been incorporated in a number of knowledge elicitation tools as means of pump-priming the elicitation with a range of entities and concepts. KSSO (Gaines 1987c), KITTEN (Shaw & Gaines 1987), KRITON (Linster 1988) and Cognosys (Woodward 1988) all provide tools for text presentation and semi-automatic analysis as part of the knowledge elicitation process.

As techniques for knowledge extraction from text are developed they will play an increasingly important role in knowledge acquisition. It is unlikely that textual sources will ever replace elicitation from experts for many of the reasons already discussed—knowledge is continually being created and experts, if they write it down at all, do so after considerable delay and with much loss of significant components. However, incorporation of what knowledge is available from written media will become an important source. The same considerations apply to diagrams where, however, the state of the art of knowledge extraction is rather less advanced than with text.

CONVERGENCE WITH HYPERMEDIA

Hypertext and hypermedia are computer-based extensions of existing media such as books and films, and the representation of knowledge in these media has important applications to knowledge acquisition (Nelson 1980, Conklin 1986, Gaines & Vickers 1988). A number of knowledge acquisition tools have already been written in Apple's HyperCard and this trend may be expected to continue. Hypermedia research has focused on the anarchical nature of much knowledge in that it is unstructured and non-algorithmic, where knowledge acquisition research has focused on the ordered nature of some knowledge in that it is well-structured and computer-processible. The term 'knowledge' has come to be used in a very narrow sense in the computing science literature, as 'those aspects of human knowledge that can be formally represented and processed.' This is a major restriction of the usual meaning of knowledge as 'acquaintance with ascertained truths facts or principles'. The philosophical definition of knowledge as 'more than opinion, less than truth' captures the colloquial sense and is a useful concept for system development.

Hypermedia are particularly important in knowledge acquisition because much human knowledge is implicit in action and not formally represented and processed. Hypermedia go beyond knowledge bases in not requiring that the information system be able to represent and process the knowledge encoded in the media used. They subsume knowledge bases but extend knowledge representation to include any media whereby human knowledge can be transferred even if there is no means for analyzing that knowledge or its transfer. For example, a film of a skilled person performing a task which enables another person to perform that task better contains knowledge which can be accessed and transferred through hypermedia (Vickers & Kingston 1987).

Hypermedia are very broadly defined and there is a sense in which they encompass all aspects of knowledge-based systems and their links with other information technologies. Figure 15 attempts to capture the key features of various concepts of hypermedia. Four major conceptual areas are involved: human society and culture, particularly its knowledge processes; instrumentation and control technology, particularly transducers that capture audio-visual events in the world; information technology, particularly databases and information retrieval; and media, ranging from basic drawing materials through to videodiscs.

The *strict hypermedia* definition sees hypermedia as integrating some components of information technology and modern graphic and videodisc media. The growing *hypermedia* concept is more general, seeing them as using information technology to extend all past media. The *generalized hypermedia* definition sees them as *knowledge support systems* (Gaines & Shaw 1987) that integrate existing human knowledge transfer processes with information technology, instrumentation and control technology, and the full range of media. There is no conflict between these interpretations—they differ only in their strictness and are usually differentiated by context.

Developers of knowledge-based systems shells are providing links from them to hypermedia products to support the application of the wide variety of knowledge representations offered. Knowledge acquisition tools are being integrated with hypermedia systems for similar reasons, and this raises interesting possibilities for the representation and use in explanation and tutoring of a much wider range of forms of knowledge than has been possible in the past. The support of knowledge acquisition methodologies through hypermedia tools and the

integration of knowledge acquisition systems, hypermedia systems and knowledge-based systems shells is a major industry trend.

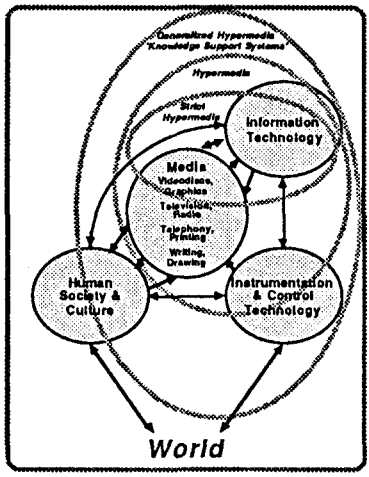

Fig.15 Hypermedia as Knowledge Support Systems

CONVERGENCE WITH CASE

Software engineering for knowledge-based systems development has developed outside the framework of most existing software engineering methodologies. The concepts of "artificial intelligence" and "expert systems" have been taken to have sufficient novelty to justify radically different approaches under the heading of "knowledge acquisition" and "knowledge engineering." However, while some significant differences are apparent, many of the problems of knowledge engineering are ones already addressed in software engineering. Some of the "knowledge acquisition" tools that have been developed are also applicable to the development of decision-support and database systems. However, from another perspective the integration of "knowledge" into information-processing, with the connotations of subjectivity, uncertainty, disagreement and contradiction, introduces new problems of software engineering (Gaines 1988b). We have developed methodologies for "hard" data that do not necessarily transfer easily to the "softer" data that is knowledge.

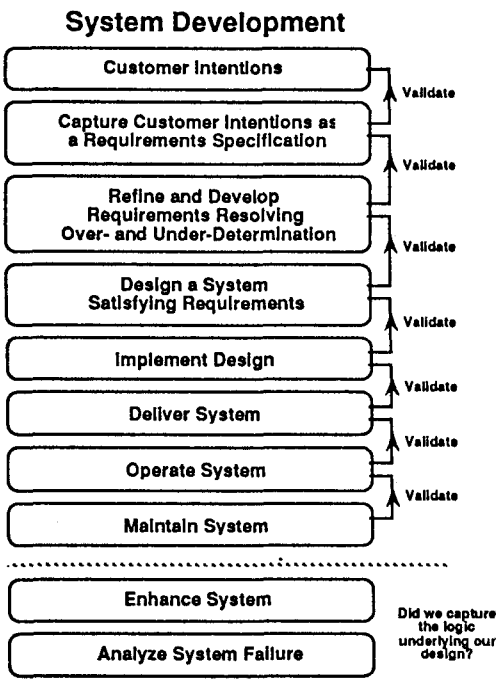

Fig.16 System Design as Supported through Computer-Aided Software Engineering (CASE)

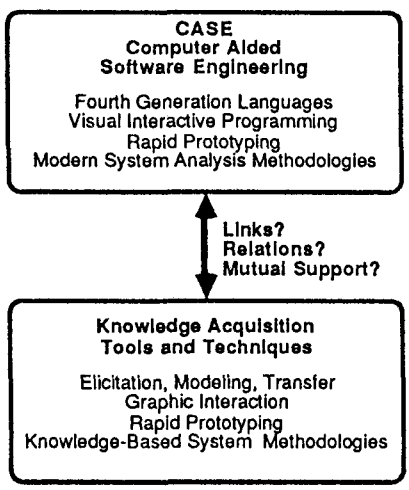

Fig.17 Synergisms between CASE and Knowledge Acquisition

Computer-Aided Software Engineering (CASE) tools range from those that are clearly technical and logical extensions of existing compiler and support environment techniques to those which attempt to deal with many of the "softer" issues of system development such as tracing customer requirements and design decisions (Fisher 1988). Figure 16 shows the many levels of system development for which there now exist structured development methodologies and CASE tools. The diagram is equally applicable to the development of knowledge-based systems, and it is interesting to examine potential applications of CASE tools in KBS development, and the relation between knowledge acquisition tools and similar CASE tools. For example, data design tools using entity-attribute-relation models are related to the entity-attribute, or repertory, grids used in many knowledge elicitation tools.

Figure 17, as a system development model, makes one realize how many aspects are neglected in current knowledge-based system developments. AI technologies are as yet highly immature compared with conventional software engineering. In particular, knowledge base maintenance and system failure analysis will become major problem areas as significant systems come into use. However, knowledge acquisition tools designed to work directly with experts and support them through non-technical interfaces show up very well compared with CASE tools eliciting the same knowledge. The two families of tools have been designed for different markets and user communities.

Since most CASE tools are designed around an open architecture database so that customer tools may be integrated with those supplied, there is tremendous scope for interfacing existing knowledge acquisition tools to CASE databases and using them for preliminary designs of data structures and decision protocols. This is an open area for research and development with exciting practical potential.

CONCLUSIONS

There are many possible approaches to knowledge engineering. However, most researcher and practitioners agree that knowlege acquisition is the bottleneck in the task of developing knowledge based systems. It has been shown in this paper how the role of the knowledge engineer is changing from the repository of expertise to the manager of the whole elicitation and encoding process. Several automated methods already exist, and in this paper some well-established methods of eliciting knowledge used in other disciplines are shown to be relevant to the knowledge acquisition problem. The underlying technological infrastructure was put into the perspective of computer science developments, and related to the social phenonemon of expertise. Some future developments in integrating knowledge acquisition with existing methodologies in other areas were elucidated such as dealing with the problems caused by having to use multiple experts; the problems of validation of knowledge acquisition and knowledge based systems; the use of tools from the field of machine learning; the relationship with hypermedia systems; and the convergence with CASE methodologies.

REFERENCES

Bainbridge, L. (1979). Verbal reports as evidence of the process operator's knowledge. **International Journal of Man-Machine Studies, 11(4), 411-436** (July).

Bainbridge, L. (1986). Asking questions and accessing knowledge. **Future Computing Systems 1(2),** 143-149.

Benbasat, I. & Dhaliwal, J.S. (1988) A framework for the validation of knowledge acquisition. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Third AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop.** pp.1-1-1-18. Banff (November).

Boose, J. H. (1986) Rapid Acquisition and Combination of Knowledge from Multiple Experts in the Same Domain. **Future Computing Systems 1(2)** 191-216

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 Press.

Broadbent, D.E., FitzGerald, P. & Broadbent, M.H.P. (1986). Implicit and explicit knowledge in the control of complex systems. **British Journal of Psychology 77,** 33-50.

Cendrowska, J. (1987) An algorithm for inducing modular rules. **International Journal of Man-Machine Studies 27** (4), 349-370 (October).

Collins, H.M. (1985). **Changing Order: Replication and Induction in Scientific Practice.** London: SAGE.

Conklin, J. (1986). A survey of hypertext. MCC Technical Report STP-356-86. Austin:Microelectronics & Computer Technology Corporation (October).

Dixon, N. (1981). **Preconscious Processing.** Chichester: Wiley.

Ericsson, K.A. & Simon, H.A. (1984). **Protocol Analysis: Verbal Reports as Data.** Cambridge, Mass. MIT Press.

Fisher, A.S. (1988) **CASE: Using Software Development Tools.** New York: John Wiley.

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

Gaines, B.R. (1987b) Advanced expert system support environments. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Second AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop.** pp.8-0-8-14. Banff (October).

Gaines, B.R. (1987c). 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.

Gaines, B.R. (1988a). 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).

Gaines, B.R. (1988b). Software engineering for knowledge-based systems. **Proceedings of CASE'88, Second International Workshop on Computer-Aided Software Engineering**. 14-1-14-7.
Gaines, B.R. (1988c) Knowledge acquisition and technology. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Third AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop**. pp.8-1-8-21. Banff (November).
Gaines, B.R. (1989) An Ounce of Knowledge is Worth a Ton of Data: Quantitative Studies of the Trade-Off between Expertise and Data based on Statistically Well-Founded Empirical Induction. KSI-89-2, Knowledge Science Institute, University of Calgary.
Gaines, B.R. & Shaw, M.L.G. (1986). A learning model for forecasting the future of information technology. **Future Computing Systems, 1**(1), 31-69.
Gaines, B.R. & Shaw, M.L.G. (1987). Knowledge support systems. **ACM MCC-University Research Symposium**. Austin, Texas: MCC. pp.47-66.
Gaines, B.R. & Vickers, J.N. (1988). Design considerations for hypermedia systems. **Microcomputers for Information Management**, 5(1), 1-27 (March).
van Gundy, A.B. (1981). **Techniques of Structured Problem Solving**, Van Nostrand Reihold:New York.
Hagstrom, W.O. (1965). **The Scientific Community**. New York: Basic Books.
Hawkins, D. (1983). An analysis of expert thinking. **International Journal of Man-Machine Studies, 18**(1), 1-47 (January).
Johnson L. & Johnson, N.E. (1987) Knowledge elicitation involving teachback interviewing. Kidd, A. (Ed) **Knowledge Elicitation for Expert Systems: A Practical Handbook.** Plenum Press.
Johnson, P.E., Zualkernan & Garber, S. (1987). Specification of expertise: **International Journal of Man-Machine Studies, 27**(2), 161-181 (February).
LaFrance, M. (1987) The knowledge acquisition grid: a method for training knowledge engineers. **International Journal of Man-Machine Studies**, 27(2), 245-255 (February).
Lenat, D., Prakash, M. & Shepherd, M. (1986). CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. **AI Magazine 6**(4), 65-85.
Linster, M. (1988) A critical look at KRITON. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Third AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop**. pp.17-1-17-19. Banff (November).
McGraw, K.L., and Harbison-Briggs, K. (1989). **Knowledge Acquisition Principles and Guidelines**, Prentice Hall: New Jersey.
McKeown, B., and Thomas, D. (1988). **Q Methodology** Sage California.
Michalski, R.S. & Carbonell, J.G., Eds. (1983). **Machine Learning: An Artificial Intelligence Approach**. Palo Alto, California: Tioga.
Michalski, R.S., Carbonell, J.G. & Mitchell, T.M., Eds. (1986). **Machine Learning: An Artificial Intelligence Approach Volume II**. Los Altos, California: Morgan Kaufmann.
Nelson, T. (1980). Interactive systems and the design of virtuality. **Creative Computing, 6**(11), 56-62 (November).
Nisbett, R.E. & Wilson, T.D.. (1977). Telling more than we can know: verbal reports on mental processes. **Psychological Review 84**, 231-259.
Pask, G. (1975). **Conversation, Cognition and Learning**, Elsevier, Amsterdam.
Pidgeon, N.F., Turner, B.A. and Blockley, D.I. (1989) The Use of Grounded Theory for Conceptual Analysis in Knowledge Elicitation, **International Journal of Man-Machine Studies**, (in press).
Rusk, R.A. & Krone, R.M. (1984) The Crawford Slip Method (CSM) as a tool for extraction of expert knowledge, Salvendy, G. (Ed.) **Human-Computer Interaction**,. Elsevier, New York, pp279-282.
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).
Shaw, M.L.G. & Gaines, B.R. (1988) A methodology for recognizing conflict, correspondance, consensus and contrast in a knowledge acquisition system. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Third AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop**. pp.30-1-30-19. Banff (November).
Shaw, M.L.G. & Woodward, J.B. (1987) Validation of a knowledge support system. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Second AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop**. pp.18-0-18-15. Banff (October).
Slator, B.M. (1988) Extracting lexical knowledge from dictionary text. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Third AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop**. pp.33-1-33-18. Banff (November).
Sowa, J. (1984) **Conceptual Structures**. Addison Wesley Publishing Co. Reading Ma. 1984.
Steels, L. (1986). Second generation expert systems. Bramer, M.A. (Ed). **Research and Development in Expert Systems III**. pp.175-183. Cambridge: University Press.
Szpakowicz, S. (1988) Semi-automatic acquisition of conceptual structure from technical texts. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Third AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop**. pp.35-1-35-16. Banff (November).
Vickers, J.N. & Kingston, G.E. (1987). Modelling the master coach: building an expert system for coaching. Norrie, D., Ed. **Learning in Future Education: Proceedings of International Conference on Computer Assisted Learning in Post-Secondary Education**. pp.207-212. University of Calgary.
Woodward, B. (1988) Knowledge engineering at the front-end: defining the domain. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Third AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop**. pp.37-1-37-25. Banff (November).