Journal of Intelligent Information Systems 1(1) 1992

INTEGRATED KNOWLEDGE ACQUISITION ARCHITECTURES

Brian R Gaines and Mildred L.G. Shaw

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

ABSTRACT
An architecture for knowledge acquisition systems is proposed based upon the integration of existing methodologies, techniques and tools developed within the knowledge acquisition, machine learning, expert systems, hypermedia and knowledge representation research communities. Existing tools are analyzed within a common framework to show that their integration can be achieved in a natural and principled fashion. A detailed architecture for integrated knowledge acquisition systems is proposed that also derives from parallel cognitive and theoretical studies.

1 INTRODUCTION

The past decade has seen an explosion in research on, and application of, knowledge acquisition methodologies, techniques and tools (Marcus, 1988; Boose & Gaines, 1988, 1990; Gaines & Boose, 1988; Boose, 1989). The knowledge acquisition community world-wide has grown in numbers and scope of projects. There are significant international collaborative developments involving the sharing of ideas and software. The problem now is not so much to access research and experience in knowledge acquisition but to make sense of the diverse and wide ranging material available and, in particular, to apply the results to improve effectiveness in the development of knowledge-based systems. There are a number of major impediments to such understanding and application:

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

This paper draws on the research and results of many researchers in the knowledge acquisition and knowledge representation communities to propose a practical architecture for future knowledge-based systems that addresses these issues. It synthesizes a system design from what already exists, putting a diversity of well-founded and widely used approaches to knowledge acquisition within an integrative framework. The design is intended to be clean and simple, easy to understand, and easy to implement. It is very open in its architecture, providing a frame within which to experiment with new approaches to knowledge acquisition rather than a cage that locks us in to static paradigms. It is also intended to be practically applicable by being based on what we already know how to do well.

Example screen dumps illustrating the operation of various types of knowledge acquisition tools will be drawn largely from our own rsearch, notably the knowledge acquisition tools: KSS0. (Shaw & Gaines, 1987; Gaines, 1988) based on text analysis, repertory grid elicitation and inductive modeling; and KSSn (Gaines, 1991a), which integrates this with visual editing of semantic nets and a term subsumption knowledge representation server. However, the integrative framework developed is intended to be widely application, and references are given to a range of publications from other research groups exemplifying the approaches discussed.

2 KNOWLEDGE ACQUISITION PARADIGMS

The overall objective of a knowledge acquisition process is to develop a computational knowledge base that is operational in that it forms a basis for automatic inference when loaded into a knowledge-based system shell. To arrive at the computational knowledge base, however, usually involves a major data collection activity that commences with largely informal knowledge from a wide variety of sources, and then focuses on more structured knowledge elicitation from key knowledge sources, transforming what is elicited into formal knowledge structures that form the basis of the computational knowledge base. The management of this data collection activity and the, often very large, corpus of material resulting, has been a major candidate for support through knowledge acquisition tools, generally based on hypermedia technology. Use of these tools results in a knowledge acquisition database containing system development materials supporting the knowledge modeling central to the knowledge engineering process.

In the first phase of knowledge-based system development the knowledge acquisition database was seen as relevant to system development only, and put aside once a system was in operation. However, as the limitations of the ‘explanations’ generated by backward chaining reason from computational knowledge structures became apparent (Hughes, 1986; Swartout, 1986), the rich, informal but humanly understandable material available in the knowledge acquisition database was seen to be a major asset. In the second phase of knowledge-based system development the chain of derivation of the computational knowledge base from the informal knowledge was maintained as part of the run time system enabling meaningful explanations to made available for the system’s rules and terminology (Abu-Hakima & Oppacher, 1990). It also became apparent that a major characteristic of most knowledge-based systems is that knowledge itself is never static but is subject to continuous revision and enhancement, both from sources outside the system and from experience gained in the use of the system. Hence, in the third phase of knowledge-based system development, the integration of the knowledge acquisition process with the knowledge-based system shell, and the continuing use of knowledge acquisition tools and techniques during the application of the system has become a major design requirement (Gaines & Linster, 1990; Gaines, Rappaport & Shaw, 1992).

Figure 1 left illustrates the relations between various forms of knowledge gathered during the knowledge acquisition process and the current concept of a ‘knowledge base’ as composite of informal, structured, formal and computational knowledge, all linked together through dependency relations providing mutual support in explanation and ongoing development. It is this total structure that any integrated knowledge acquisition architecture has to support. There are four major paradigms underlying current tools and techniques designed to support the knowledge acquisition process itself: the use of hypertext and hypermedia tools to capture informal knowledge and begin to structure it; direct editing of knowledge in a semantic network, frame, or rule, representation; indirect elicitation through repertory grids in which critical cases are described in terms of relevant attributes; and inductive derivation of knowledge from data sets of varying quality. These paradigms, instances of their application, and relations between them are discussed in the following sub-sections. The way in which these basic paradigms are used, separately and in combination, depends on the knowledge modeling strategy adopted, and this is discussed in the next major section.

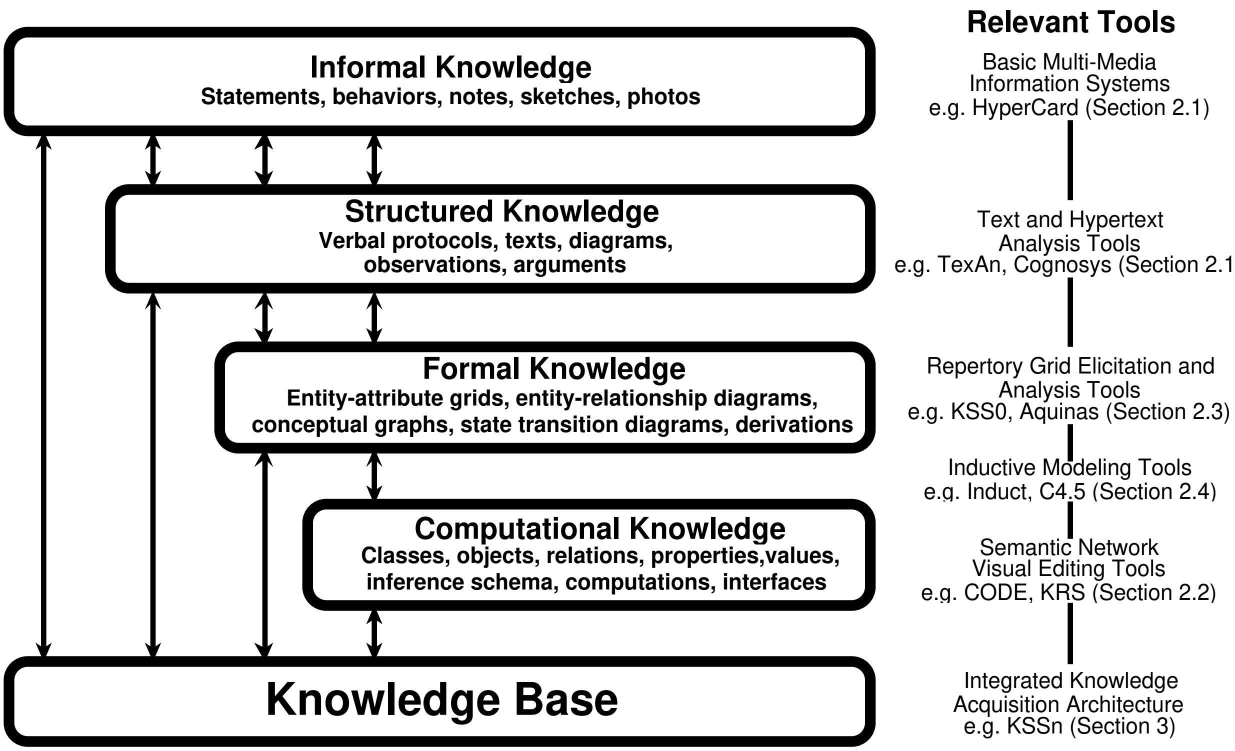

Fig.1 Knowledge acquisition for a knowledge-based system

Figure 1 right shows the topics covered in this paper and some of the tools illustrating the relevant technologies discussed in each section. The objective is to show that all the different approaches may be brought together in a common framework leading to a unified architecture in which they can be combined.

2.1 Semi-formal elicitation and structuring through hypertext and hypermedia

As already noted, much knowledge is informal yet still valuable in an knowledge-based system. Text and pictures can encode and impart usable expertise, supplementing computational knowledge. Thus, the parallel development of hypertext and hypermedia is having a substantial impact on expert system architectures and knowledge acquisition tools. Figure 2 shows some of the features of modern document processing systems that impinge on knowledge acquisition. Documents may be acquired from many sources, displayed, parts re-used in other documents, and parts linked for hypertext navigation. The text in documents may also be analyzed for associative clusters and these clusters may be grouped to indicate significant concepts. For example, Figure 3 shows the text of this paper clustered using the TexAn tool in KSS0 which derives word associations through estimates of statistically significant links. This can be used to extract the terminology for a domain from a relevant document and present a domain expert with an initial concept map which he or she can refine without having to enter a great deal of text on a blank screen before knowledge elicitation can effectively commence. Figure 4 summarizes the features of hypermedia systems that need to be incorporated in an integrated architecture—the provision of capabilities for domain experts to be able to manage a database of informal material and analyze it to initiate data structure in other tools.

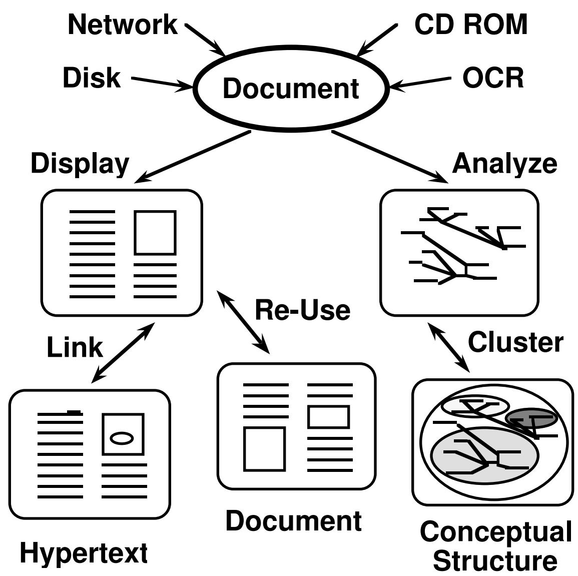

Fig.2 Hypermedia and text processing

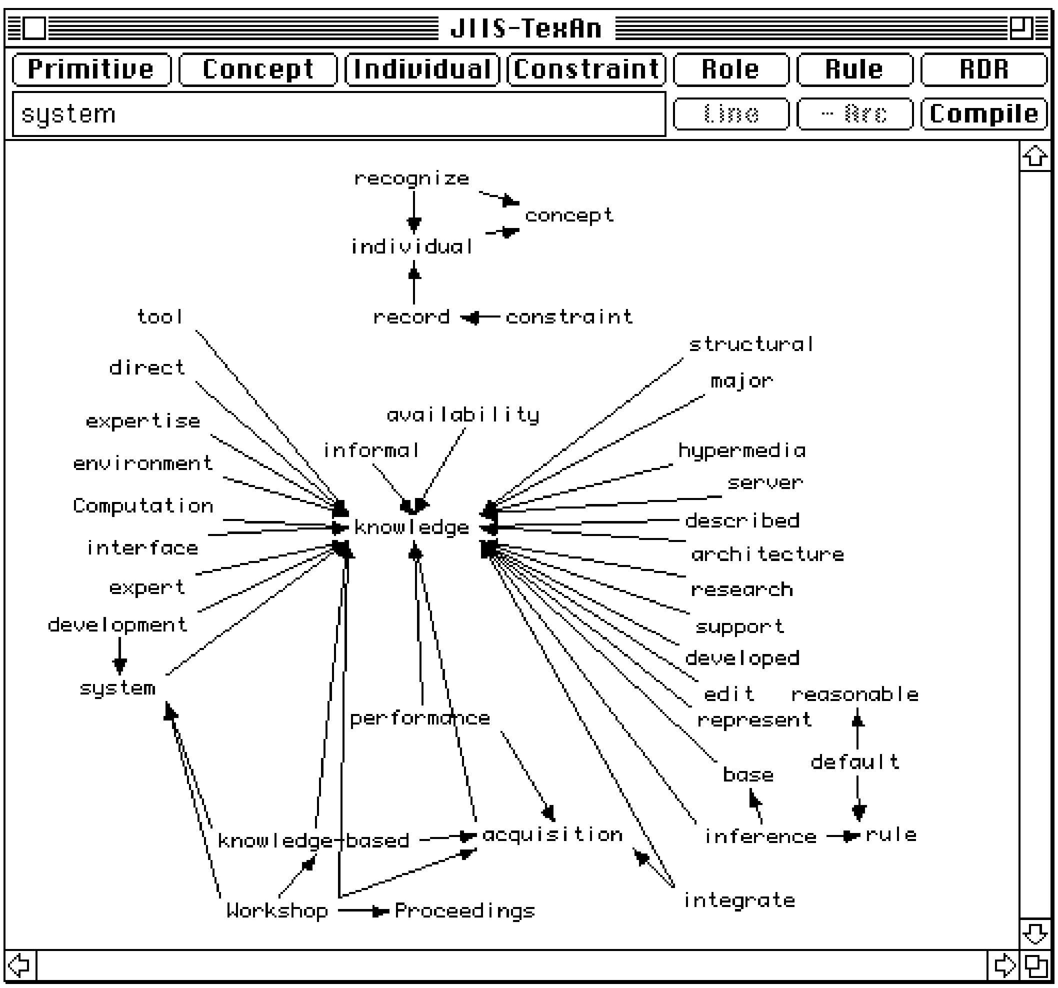

Fig.3 Cluster analysis of text in this paper using TexAn

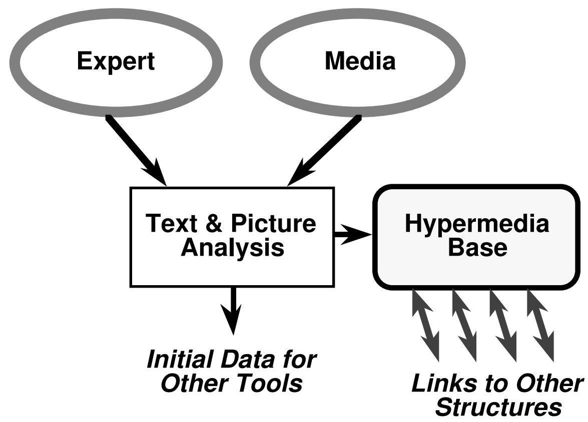

Fig.4 Hypermedia architecture

Hypertext-based knowledge acquisition tools have also been developed for use by domain experts to enter relevant case histories directly (Jones, 1990; Rantanen, 1990). They have also been used to support the knowledge engineer in structured analyses of interview material. For example, Woodward's (1990) Cognosys supports the analysis of protocols in terms of Graesser and Clark's (1985) linguistically derived "general knowledge structures". Figure 5 shows annotated text in Cognosys, and Figure 6 shows this exported to the same visual knowledge editing tool as the TexAn results in Figure 3, again for further editing and development of knowledge structures by a domain expert. Other knowledge acquisition tools such as KEATS (Motta, Rajan, Domingue & Eisenstadt, 1991) have been built around a hypertext environment specifically designed for knowledge acquisition.

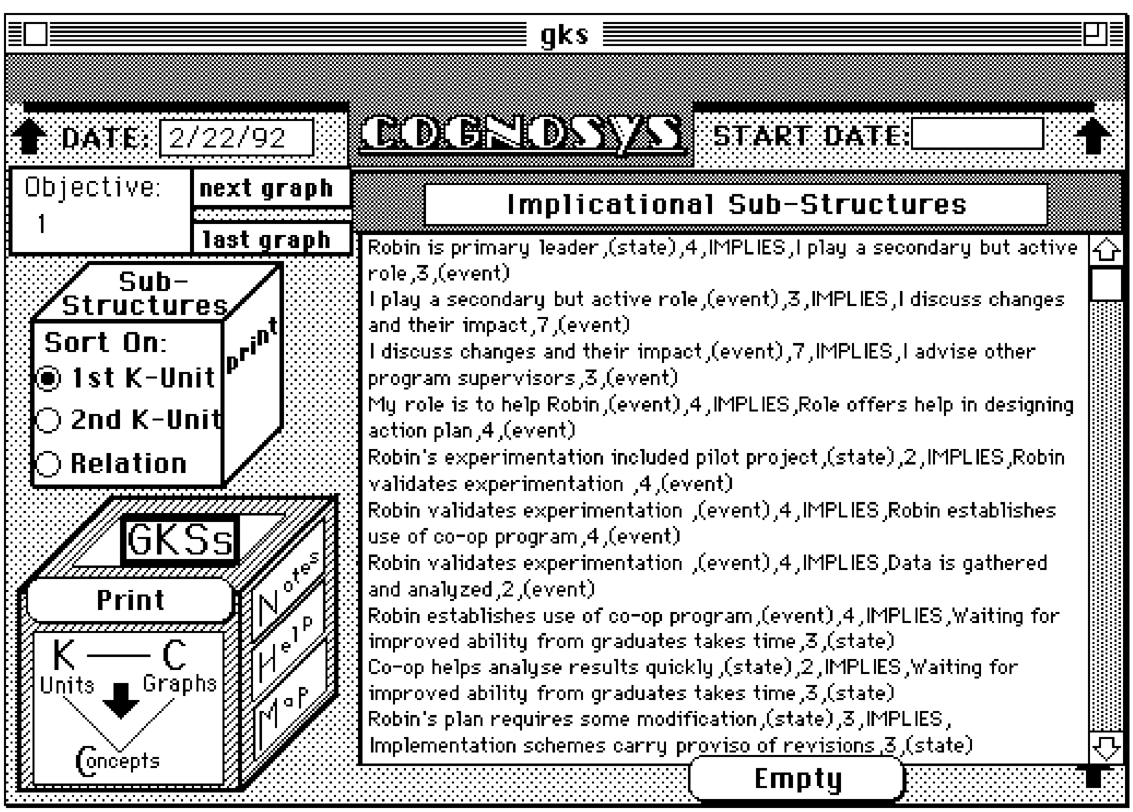

Fig.5 Development of knowledge structures from interview text in Cognosys

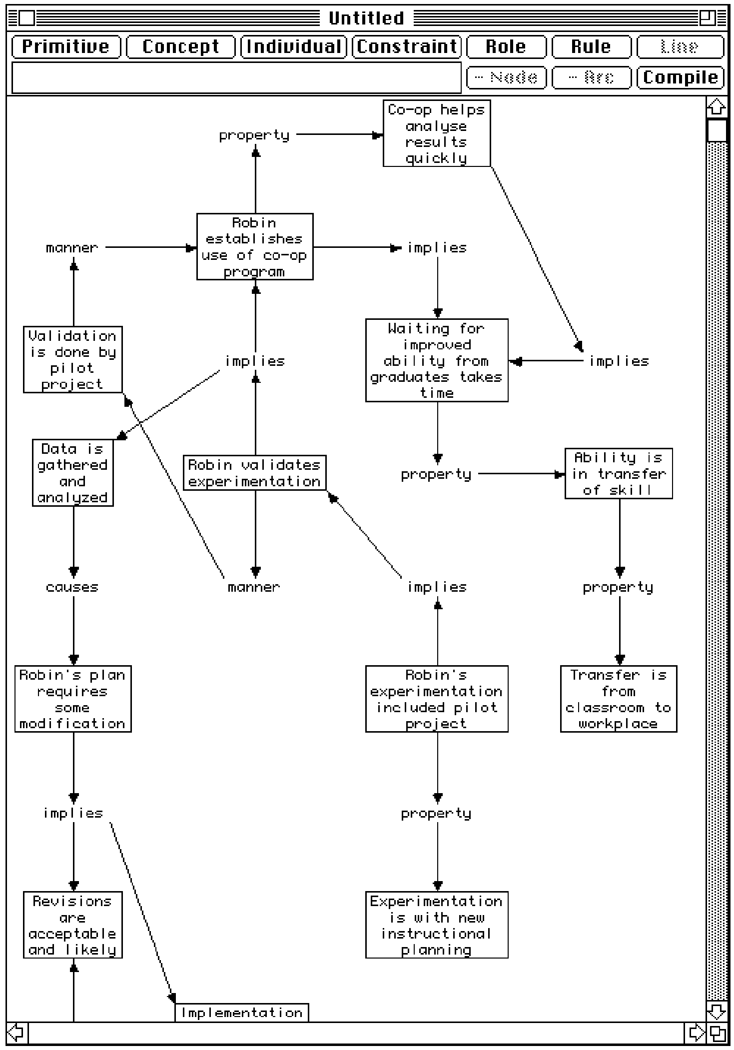

Fig.6 Development of knowledge structures from interview text in Cognosys

There is also knowledge acquisition and linguistics research targeted on the direct transfer of knowledge expressed in text to structures of frames and rules (Hausser, 1989; Gomez, 1990). Since so much knowledge is already overtly encoded in text and diagrams, in the long term this will become an essential knowledge acquisition technology. However, it is still at an early stage of development that does not satisfy the "what we already know how to do well" criterion, and is not yet suitable for incorporation in integrated systems.

Hypertext systems have been coupled to knowledge acquisition tools to provide annotation of the distinctions made and cases described which can then be used to provide explanation facilities in the final performance system (Gaines & Linster, 1990). Figure 7 shows an entity annotation card against the background of a knowledge-based system shell in a knowledge acquisition, annotation and performance system in which an annotation system in HyperCard, the knowledge acquisit ion tool KSS0, and the knowledge-based system shell Babylon (Christaller, di Primio & Voss, 1989) operate together as a single application as far as the end user can determine (Gaines, Linster & Shaw, 1992). Each item of knowledge entered through the acquisition tool automatically generates a card in the hypertext tool which is linked to that item both in the acquisition and performance phases. This card is kept up to date with the formal knowledge structures automatically, and has fields available for notes, explanations and links to text, diagrams, pictures and video sequences supplied by the domain expert or knowledge engineer.

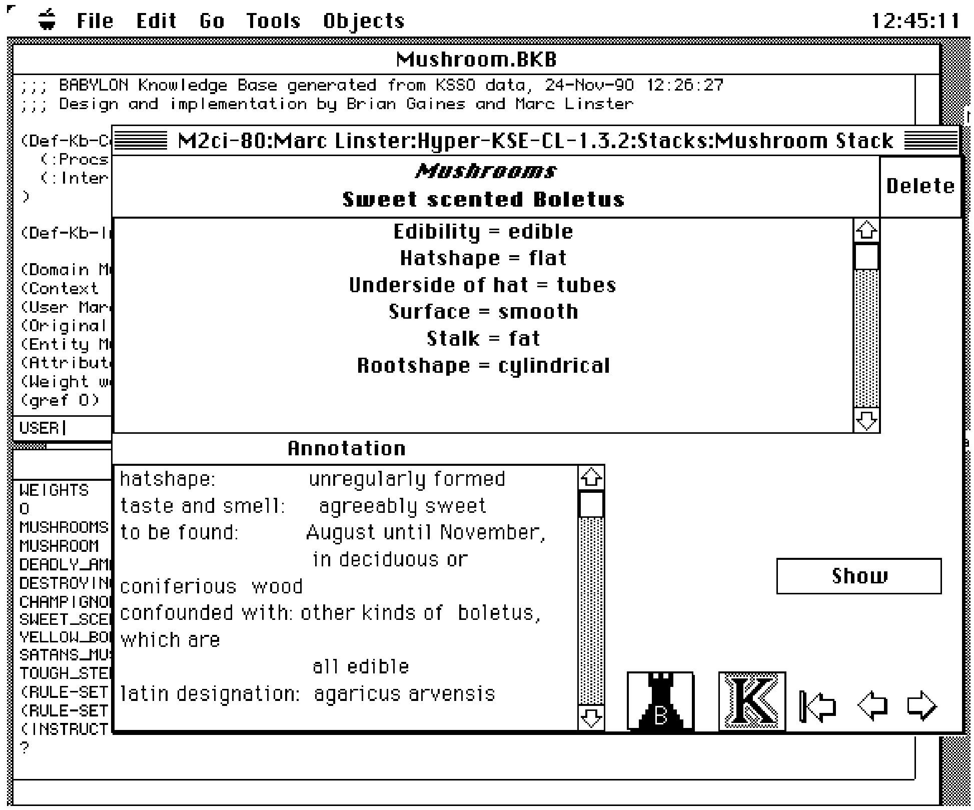

Fig.7 Entity annotation card in KSS0/HyperCard/Babylon integration

2.2 Direct editing of knowledge in a semantic network, frame, rule, representation

Once some informal perspective on a domain has been developed as shown in the upper levels of figure 1, and domain experts have been identified, in some domains where knowledge is already overt it may be possible to move directly to knowledge modeling. Graphic editors providing direct access to semantic network representations allowing knowledge to be encoded in frames and rules provide the most common development environment for knowledge-based systems. They are part of the application programming support environment of most expert system shells, and the widespread availability of modern graphic workstations has made it possible to provide excellent knowledge visualization environments.

A wide range of knowledge acquisition tools have been developed that structure and improve the graphic editing environment, often taking advantage of domain knowledge to provide a more specific, meaningful and familiar knowledge framework to the expert. Examples are MOLE (Eshelman, Ehret., McDermott & Tan, 1987), KNACK (Klinker, Bentolila, Genetet, Grimes, & McDermott, 1987), SALT (Marcus, 1987), ONTOS (Monarch & Nirenburg, 1987), KEATS (Motta, Eisenstadt, Pitman & West, 1988), CODE (Skuce, Shenkang & Beauville, 1989) and KRS (Gaines, 1991a).

Figure 8 characterizes the major features of these direct editing systems. The expert interacts through a graphic interface with an underlying semantic network knowledge representation schema which may already contain pre-encoded domain knowledge. What is elicited are:

- The distinctions that the expert makes about domain entities (attributes and relations).
- The way in which these distinctions are grouped and constrained to form major concepts.
- The entailments between concepts that constitute procedural, decision-making rules in the domain.

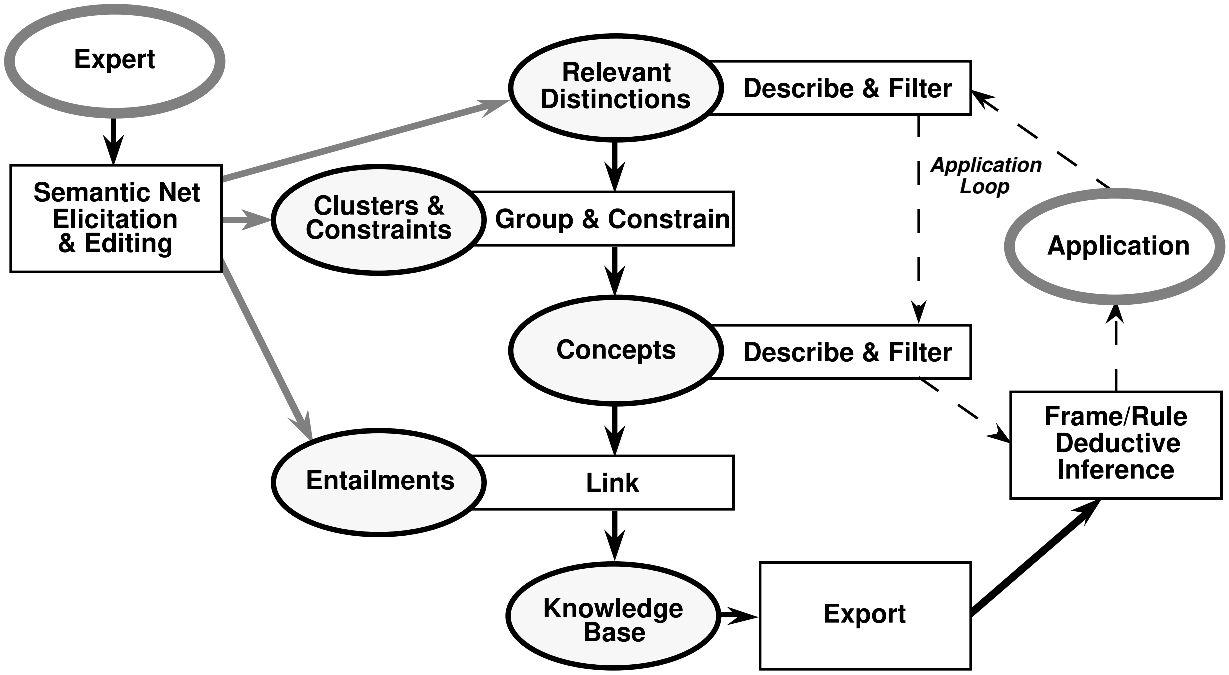

Fig.8 Semantic net architecture

Older systems have less well-structured knowledge representations but Figure 8 captures the essence of recent developments in knowledge representation that are moving towards very clean, and theoretically well-founded schema.

The knowledge base of frames and rules developed in this way is then usually exported to a performance tool and validated against test case data. This generates the application loop shown on the right of Figure 7 in which the expert's distinctions lead to a description of the problem which is structured through the concepts leading to the application of the inference rules that link them. Many acquisition tools also incorporate or link to some form of performance tool so that this validation can be made part of the elicitation process.

Figure 9 shows a knowledge structure being developed in KDraw, the visual knowledge representation language of KSSn, the most recent development of KSS0. The problem is one of room allocation derived from an ESPRIT project (Voß, Karbach, Drouven, Lorek & Schuckey, 1990) that has recently been placed in the public domain as part of Project Sisyphus. Sisyphus is research program to encourage international collaboration in knowledge-based system development developed initially by the European Knowledge Acquisition Workshop in 1989.

The visual language used in Figure 9 is precisely defined (Gaines, 199c). Concepts are ovals, primitive concepts are ovals with small horizontal lines inside each side, individuals are rectangles, roles are unboxed text, rules are rounded-corner boxes, and constraint expressions are rounded-corner boxes with small horizontal lines. Lines without arrows connecting primitive concepts denote that the concepts are disjoint, and connecting roles that they are inverse. The interpretation of the arrows in the editor is overloaded but well-defined by the types of the objects at their head and tail. For example: an arrow from one concept to another represents definitional subsumption, such as a "person" is an "animate" entity at the top left of Figure 9; and a concept → role → concept triple represents a definitional role with a conceptual constraint, such as an "organization" has exactly "1" (cardinality constraint) "head" who is a "head" (conceptual constraint) near the top right of Figure 9.

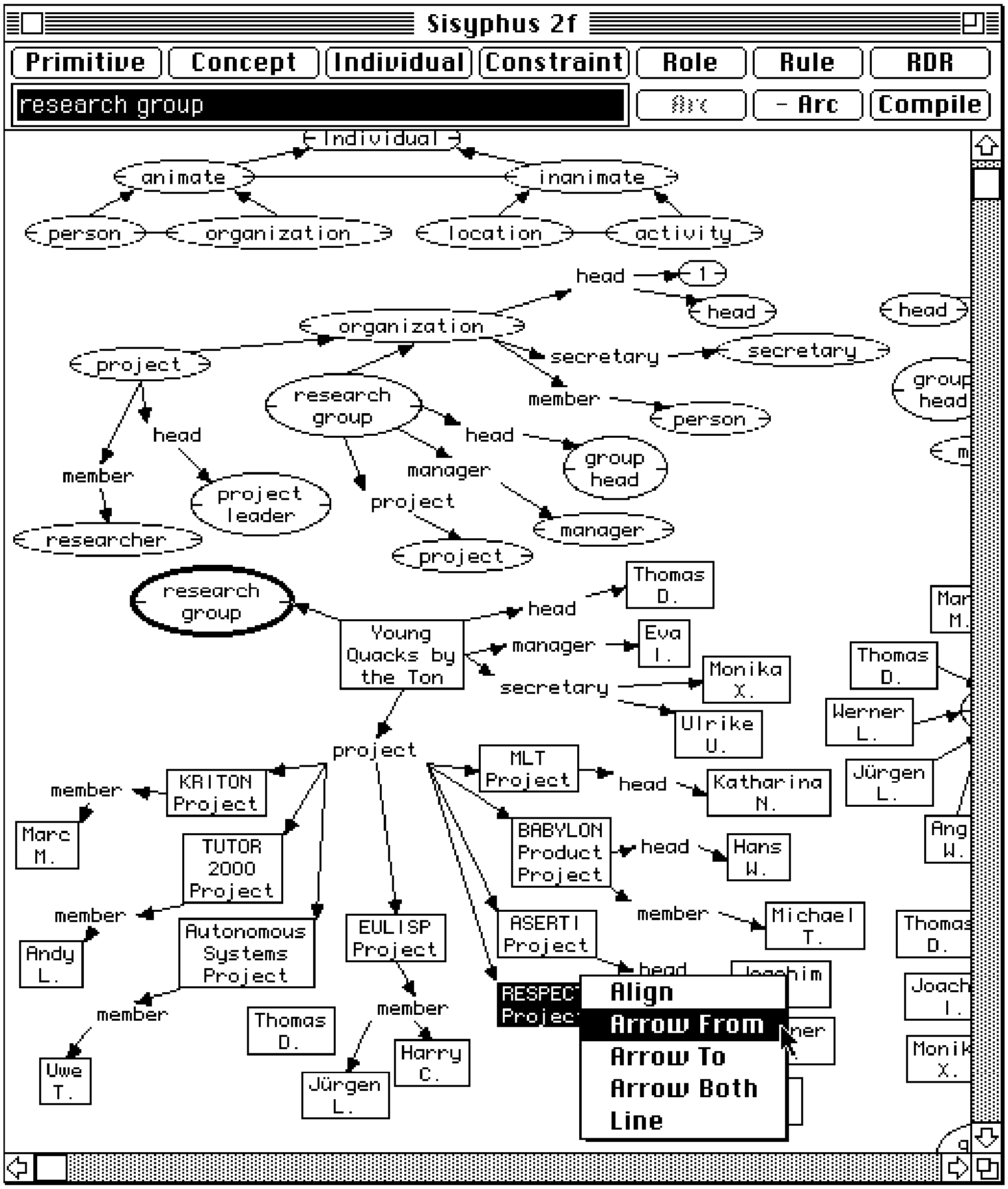

Fig.9 Visual knowledge editing as a formal semantic net in KDraw

Visual representation of knowledge structures with the potential for editing and enhancement is an attractive way of dealing with the results of other forms of elicitation, and hence semantic network editors are not so much competitors to other approaches but rather important complements to them. Thus, integration with knowledge acquisition sources as well as the capability to export to performance systems are important capabilities of any knowledge editing tool. Many indirect knowledge acquisition tools leave the knowledge presentation and editing to the associated performance tools since these often have excellent facilities. However, in an integrated architecture it is important to incorporate editors in the knowledge acquisition tool that interact effectively with all the different forms of knowledge captured. One of the major problems to be overcome is that once the knowledge has been exported and edited in the performance system it has lost its relation to the acquisition system. Most current knowledge acquisition tools do not support long-term development and knowledge base maintenance largely because of this lack of integration.

2.3 Indirect elicitation through critical cases described in relevant attributes

The advantage of knowledge editing tools is that, if the knowledge is overtly available, they provide a fast and effective rapid prototyping environment for its direct capture. Other techniques assume that the nature of expertise is such that much knowledge is not so directly available. However, effective editing tools may be seen as essential bedrock to any integrated knowledge acquisition architecture. There will usually be significant sub-domains where direct elicitation is possible, and provides the fastest, most effective means of knowledge capture. There is always a need to present and make available for validation and editing the knowledge structures captured by indirect methods.

Repertory grids provide a technique useful for knowledge elicitation when experts cannot directly enter a knowledge structure. They prompt the expert for distinctions relevant to the problem domain and for critical cases that exhibit significant phenomena in the domain. The prompting is done through online analysis of the data being entered leading to feed back to the expert suggesting missing distinctions and cases. This highly focused feedback aids the expert in developing his or her mental model of the domain. It also reduces the inefficiencies of duplication and the mental blocks of psychological set, supporting rapid prototyping.

A wide range of knowledge acquisition tools have been developed that incorporate repertory grid elicitation and analysis as their major interface to the expert. Examples are PLANET (Shaw & Gaines, 1983), ETS (Boose, 1984, 1986), AQUINAS (Boose & Bradshaw, 1987), KRITON (Diederich, Ruhmann & May, 1987), KITTEN (Shaw & Gaines, 1987), and KSS0 (Gaines & Shaw, 1987).

Figure 10 characterizes the major features of repertory grid elicitation systems. The expert interacts through a graphic interface to enter individuals in the domain (elements) and bipolar distinctions (constructs). Conceptual clustering techniques are used to feedback the elicited domain structure in an easily assimilated form for validation. Rule induction is used to generate entailments, or, more recently, conceptual induction as discussed in the next section to generate a default rule structure. What is elicited are:

- The distinctions that the expert makes about domain entities.
- Critical cases exhibiting the major phenomena affecting decision-making in the domain.
- The way in which distinctions are grouped and constrained to form concepts.
- The entailments between concepts as induced from the critical cases.

Older systems did not have explicit conceptual induction but left grouping into concepts or frames as a task for the export module.

The clustering and induction modules in Figure 10 are extensions of the basic repertory grid technique incorporated in PLANET and KSS0, and other major extensions have been incorporated in other tools. In particular, AQUINAS makes provision for a wider range of data types than the rating scales of the basic grid, and also allows hierarchies of cases and attributes to be specified that are related to those of semantic nets. Both AQUINAS and KSS0 also incorporate tools supporting multiple sources of expertise and analyzing the relationships between different sources (Gaines & Shaw, 1989; Shaw & Gaines, .

Figure 11 shows a screen from the repertory grid elicitation program KSS0 being used to elicit knowledge about factory layout planning. Two of the constructs elicited have been found to be highly matched and the program is showing the user the basis for the match and prompting for a new element to be entered that will not conform to it—thus attempting to ensure that false implications will not be drawn from the cases entered due to missing data. Figure 12 shows a FOCUS cluster analysis of the grid entered about layout planning. This gives an overview of correlations between both constructs and elements, allowing the domain expert to evaluate the entered data for expected consistencies.

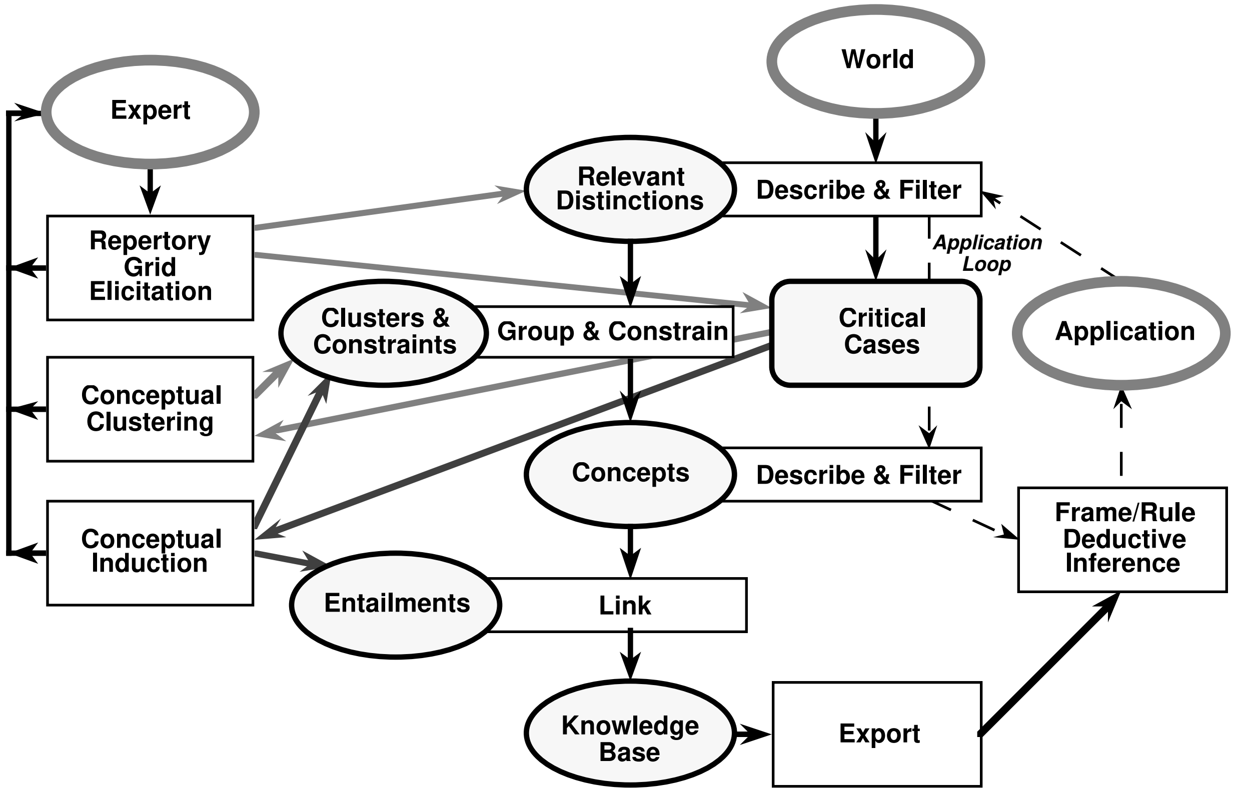

Fig.10 Repertory grid architecture

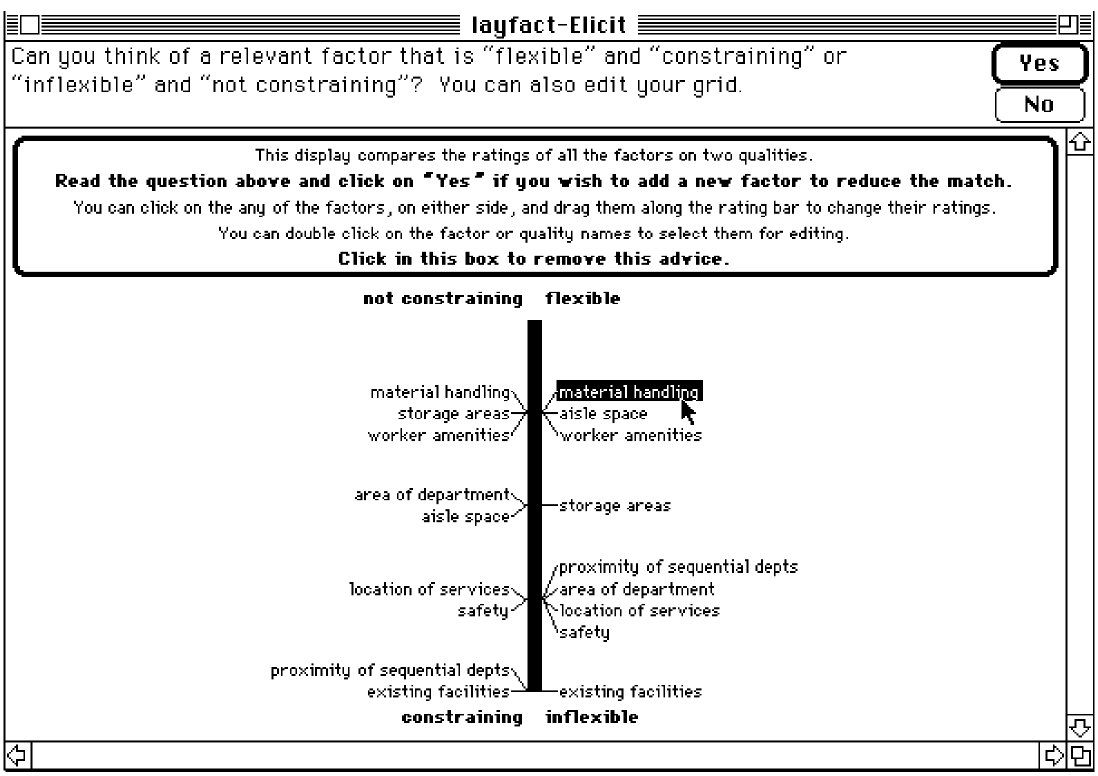

Fig.11 Construct match being used to elicit a new element

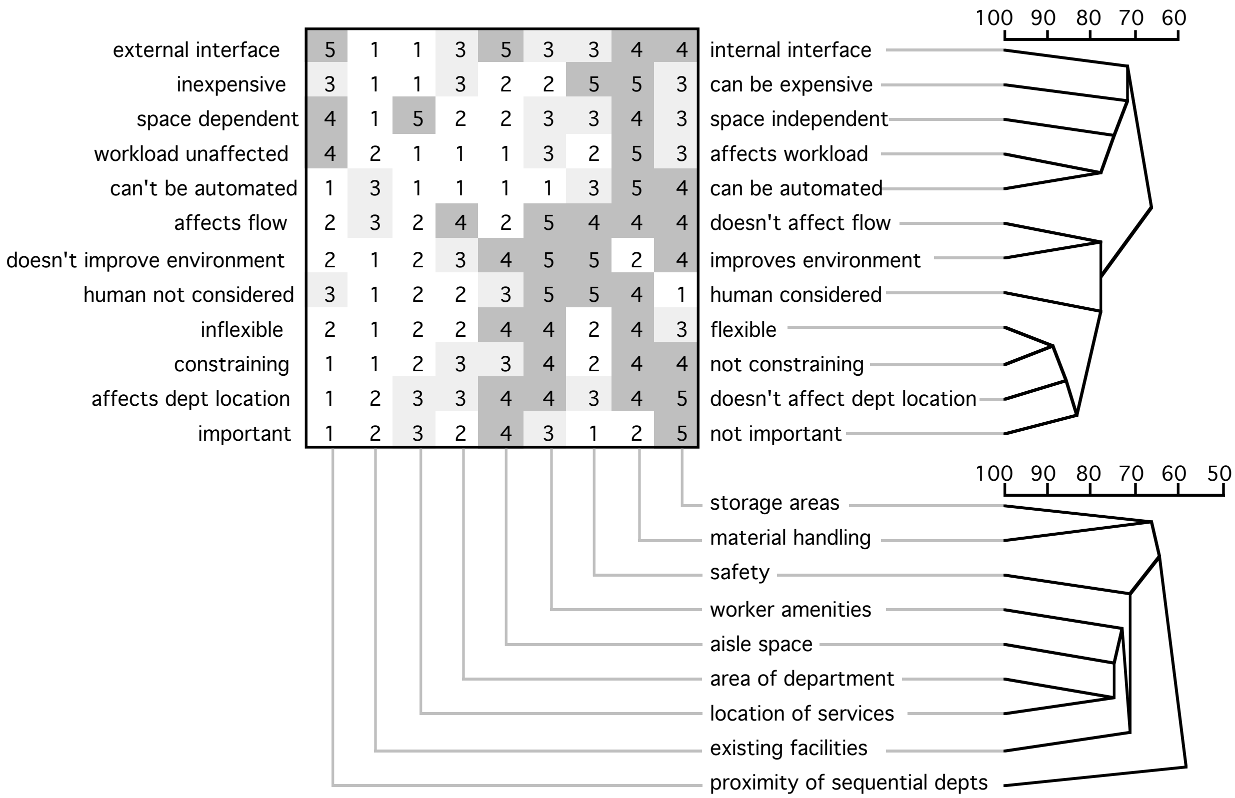

Fig.12 FOCUS cluster analysis of a repertory grid

2.4 Inductive derivation of knowledge from data sets of varying quality

When experts can neither directly enter a knowledge structure emulating their expertise nor enter critical cases stereotyping that expertise, they may still be able to point the knowledge engineer towards case histories that incorporate that expertise and are described in terms of largely relevant attributes and largely correct decisions. Empirical induction techniques may then be used to derive knowledge structures underlying the decisions made in these cases (Michalski & Chilausky, 1980).

The best known empirical induction methodology is that of Quinlan's ID3 which has been refined in many ways, particularly to tolerate noise (incorrect decisions), resulting in the current implementation, C4.5 (Quinlan, 1987). The original decision tree structure of ID3 based on a subsumption hierarchy of concepts with rules at the leaf nodes only is unnecessarily large in many situations, and extensions to ID3 have been developed that generate modular production rules directly, such as Cendrowska's (1987) Prism. This also can be extended to deal with noisy data as in Induct (Gaines, 1989). The extensions that deal with noisy data also make it possible to combine the decision tree and modular rule methodologies to generate default reasoning in which rules are placed at non-leaf nodes in the subsumption structure, and more specialized rules override more generalized ones (Gaines, 1991b). Such default rule structures are more compact than either decision trees or modular rules—they are generated by both C4.5 and Induct. Inductive methodologies have also been combined with direct knowledge editing tools, for example in BLIP (Morik, 1987).

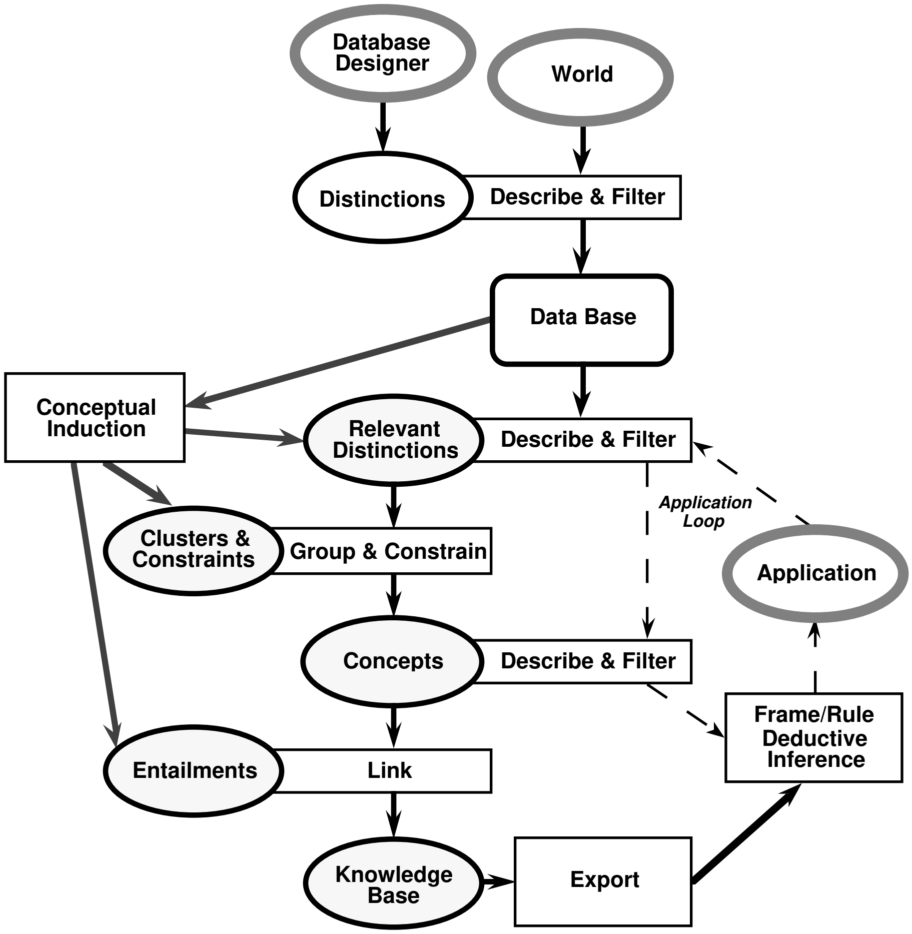

Fig.13 Conceptual induction architecture

Figure 13 characterizes the major features of conceptual induction systems. The expert indicates a database whose designer has supplied distinctions to categorize the world and cases described in terms of these distinctions to represent it. What is derived are:

• That subset of the distinctions that are relevant to the decisions.
• The way in which these distinctions are grouped and constrained to form concepts.
• The entailments between concepts that regenerate the decisions in the database.

Classic empirical induction tools do not generate the conceptual structure but this is a fairly simple extension.

Figure 14 shows the minimal default rule structure generated by Induct for Cendrowska's (1987) contact lens problem represented in the KDraw knowledge editing tool already described. What is significant about this diagram is that it illustrates the way in which empirical induction techniques can be used to generate conceptual structures for a domain. The left hand and right hand sides of a rule each defines a concept. With default rules in particular these give rise to a natural subsumption ordering. Thus the distinctions, concepts and entailments shown directly entered in Figure 8 can also be indirectly generated through induction as shown in Figure 13.

Figure 15 shows the knowledge base of Figure 14 compiled automatically into a KL-ONE-style knowledge structure for export to a knowledge-based system shell such as KRS (Gaines, 1991a), Babylon (Christaller, di Primio & Voss, 1989) or NEXPERT (Rappaport, 1987). The knowledge structures produced by direct editing such as those of Figure 9 can be exported in a similar way in a variety of formats.

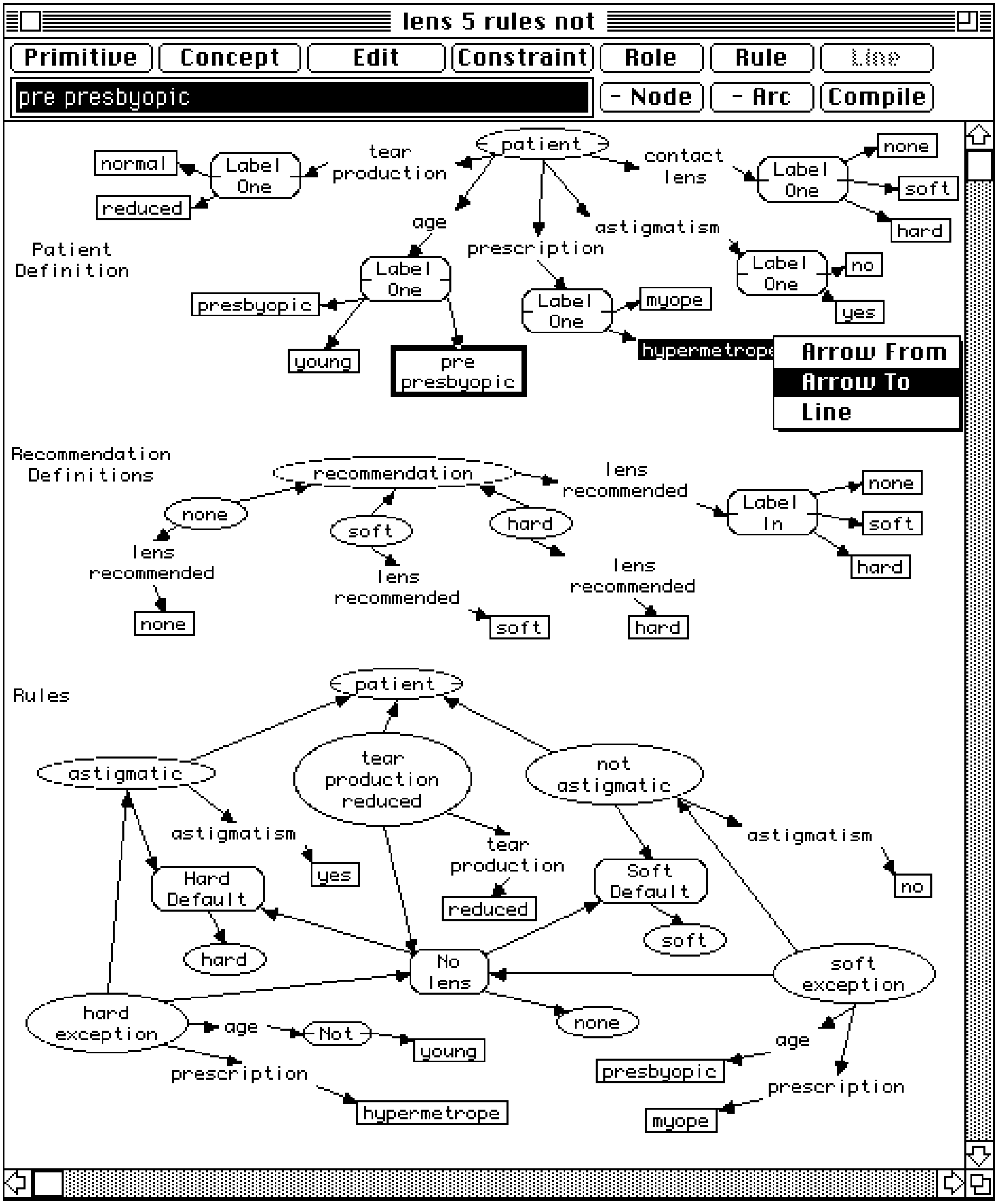

Fig.14 Induced knowledge structures

```
Primitive(patient,
  (All tear production, (Label One normal, reduced))
  (All astigmatism, (Label One yes, no))
  (All contact lens, (Label One none, hard, soft))
  (All age, (Label One young, presbyopic, pre presbyopic))
  (All prescription, (Label One myope, hypermetrope))
)
Concept(recommendation,
  (All lens recommended, (Label In none, hard, soft))
)
Concept(none, recommendation,
  (All lens recommended, (Include none))
)
Concept(soft, recommendation,
  (All lens recommended, (Include soft))
)
Concept(hard, recommendation,
  (All lens recommended, (Include hard))
)
Concept(astigmatic, patient,
  (All astigmatism, (Include yes))
  (Then Hard Default)
)
Concept(not astigmatic, patient,
  (All astigmatism, (Include no))
  (Then Soft Default)
)
Concept(tear production reduced, patient,
  (All tear production, (Include reduced))
  (Then No lens)
)
Concept(hard exception, astigmatic,
  (All prescription, (Include hypermetrope))
  (All age, (Not young))
  (Then No lens)
)
Concept(soft exception, not astigmatic,
  (All prescription, (Include myope))
  (All age, (Include presbyopic))
  (Then No lens)
)
If(Hard Default, hard)
If(Soft Default, soft)
If(No lens, none)
Unless(No lens, hard, soft)
```

Fig.15 Resultant knowledge base compiled

2.5 Comparing knowledge modeling techniques

Figures 8, 10 and 13 indicate strong similarities between the outcomes of direct knowledge elicitation, repertory grid elicitation and conceptual induction. This is as it should be since all three techniques are building a complete knowledge base. However, what is not apparent is the relative efficiencies of the different methodologies—that is, how large a database is required to generate the knowledge required to solve a problem? Figure 16 shows the results of one study to investigate the relationship between empirical induction and expertise transfer as knowledge acquisition methodologies (Gaines, 1989). Cendrowska's contact lens data was subjected to random distortion with known statistics to generate large datasets with a certain number of irrelevant binary attributes and a certain percentage of incorrect decisions. Induct was then run on the dataset with 5,000 items, 4,999 items, and so on, until the dataset failed to generate rules giving correct performance. This was done ten times for different datasets of the same type to give estimates of the mean and standard deviation of the size of dataset required to generate correct performance for different forms and levels of distortion.

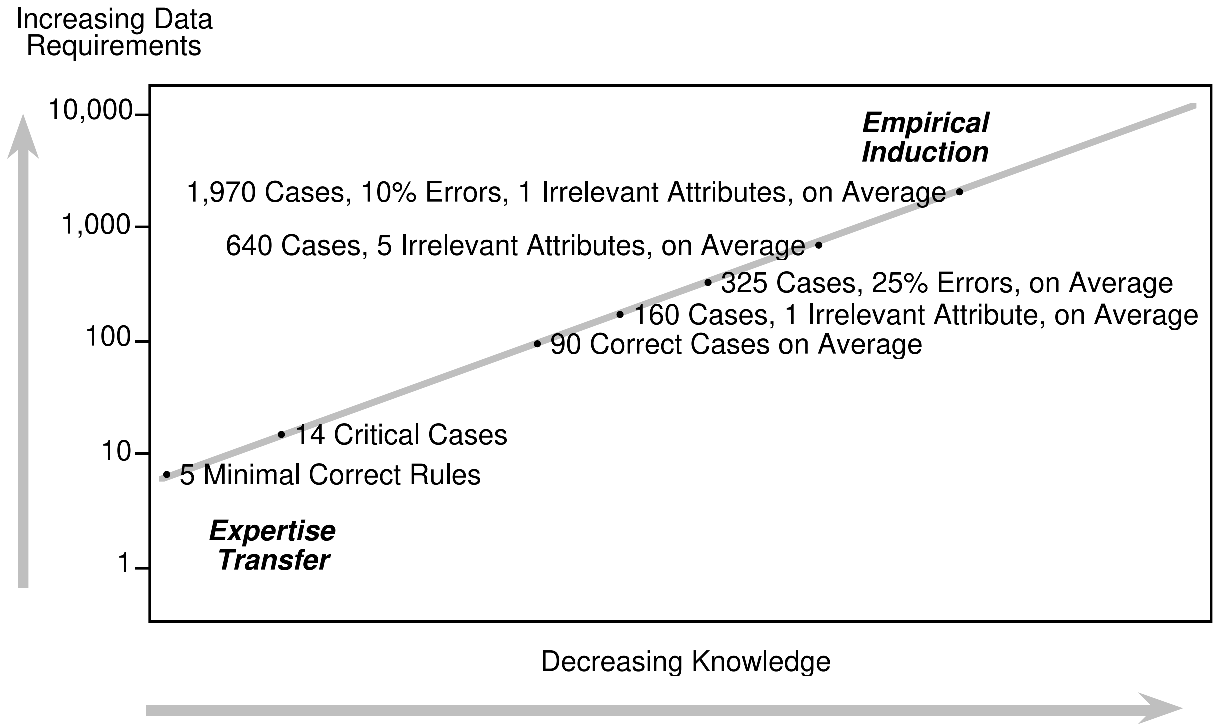

Fig.16 Data/knowledge tradeoff relating expertise transfer and empirical induction

The results shown in Figure 16 indicate the very wide range of the tradeoff between data and knowledge: from direct entry of the minimal knowledge structure of 5 default rules; through entry of 14 critical cases; through an average of 90 randomly selected correct cases; to 325 cases with 25% errors; 640 with 5 irrelevant binary noisy attributes; to 1,970 when a single irrelevant attribute interacts with a 10% error rate.

The moral from Figure 16 is not that expertise transfer is better than empirical induction, although the direct entry of overt knowledge is clearly highly ergonomic if it is available. It is rather that all three techniques described above are capable of producing equivalent quality knowledge, and there is a continuum between them in which knowledge is traded for data.

3 AN INTEGRATED SYSTEM ARCHITECTURE

The previous section has analyzed the major types of current knowledge acquisition techniques and tools within a common framework to show their similarities and differences with a view to designing an integrated system in which they operative effectively together. The techniques described join together in the multi-stage knowledge acquisition architecture shown in Figure 17 which can be seen as a detailed implementation of the process shown in Figure 1:

• Stage ⓪ consists of developing an informal knowledge base from interviews, protocols and media.
• Stage ① consists of using the informal knowledge to elicit the major coherent sub-domains that together encompass the significant phenomena in the overall domain.
• Stage ② consists of using the repertory grid methodology in each sub-domain to elicit relevant attributes and critical cases.
• Stage ③ consists of using the conceptual induction methodology to derive concepts and rules from the grids.
• Stage ④ consists of linking the sub-domains together, generally by specifying as a constraint in one domain that some role in that domain has as value an individual in another.
• Stage ⑤ consists of testing the overall knowledge base and iterating back through any of the earlier stages in order to develop and refine it further.

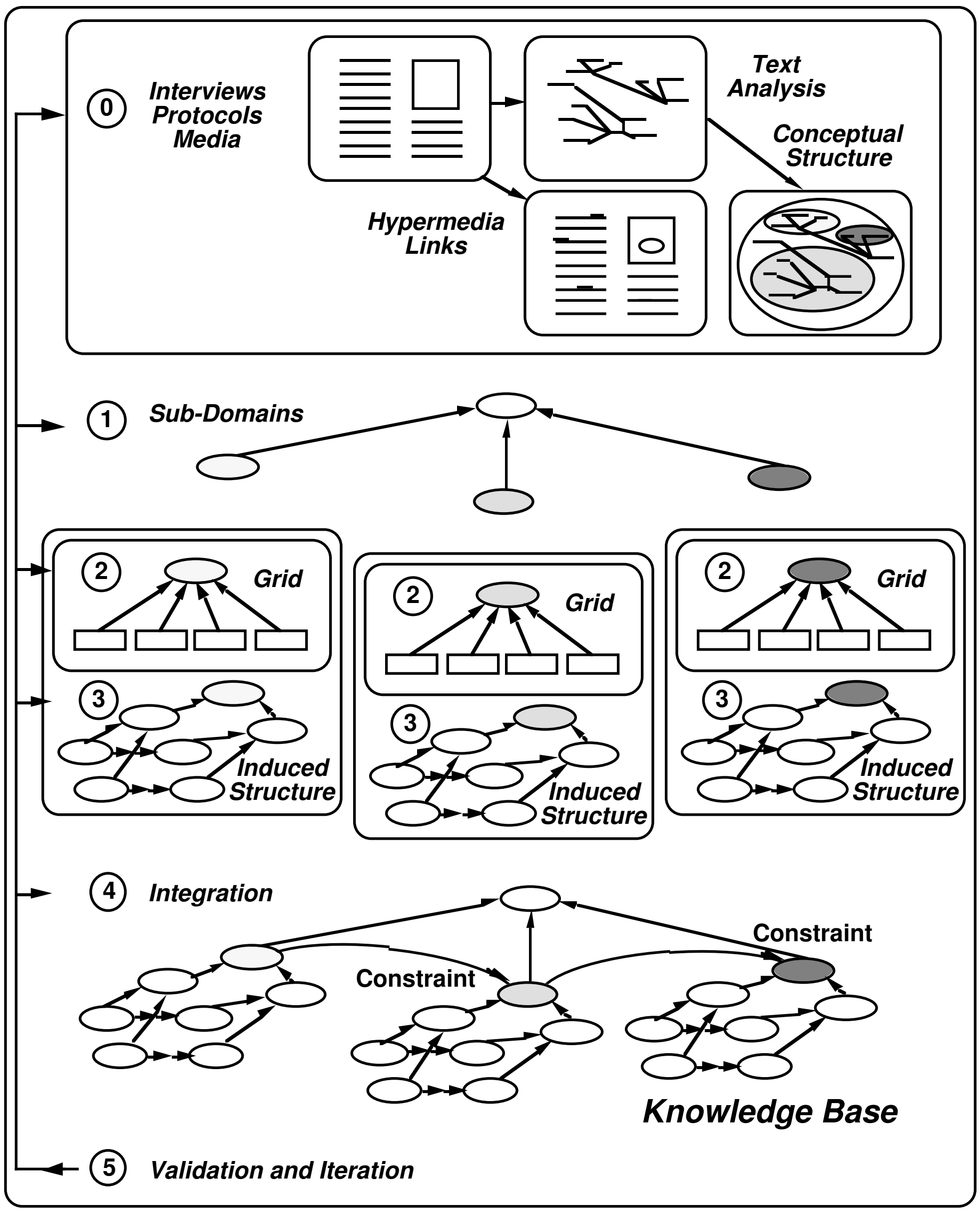

Fig.17 Integration of knowledge acquisition techniques

The architecture of Figure 16 is intended to be illustrative of the way the various knowledge acquisition techniques described naturally combine, rather than a rigid framework for an integrated tool. For example, some sub-domain structures may be developed by direct editing, others by induction from databases, others by text analysis, and others by tools not yet defined or developed. The point is that the techniques and tools that we have available now come together naturally in this framework, and it also provides an open architecture for incorporating further techniques and tools.

Figure 18 combines and extends Figures 4, 8, 10 and 13 to show the modules necessary to support the architecture of Figure 17. Nearly all the modules shown have been described already. At the upper left, text and picture entry and analysis are shown as prompting the expert and developing a hypermedia base that also has links to the computational knowledge structures.

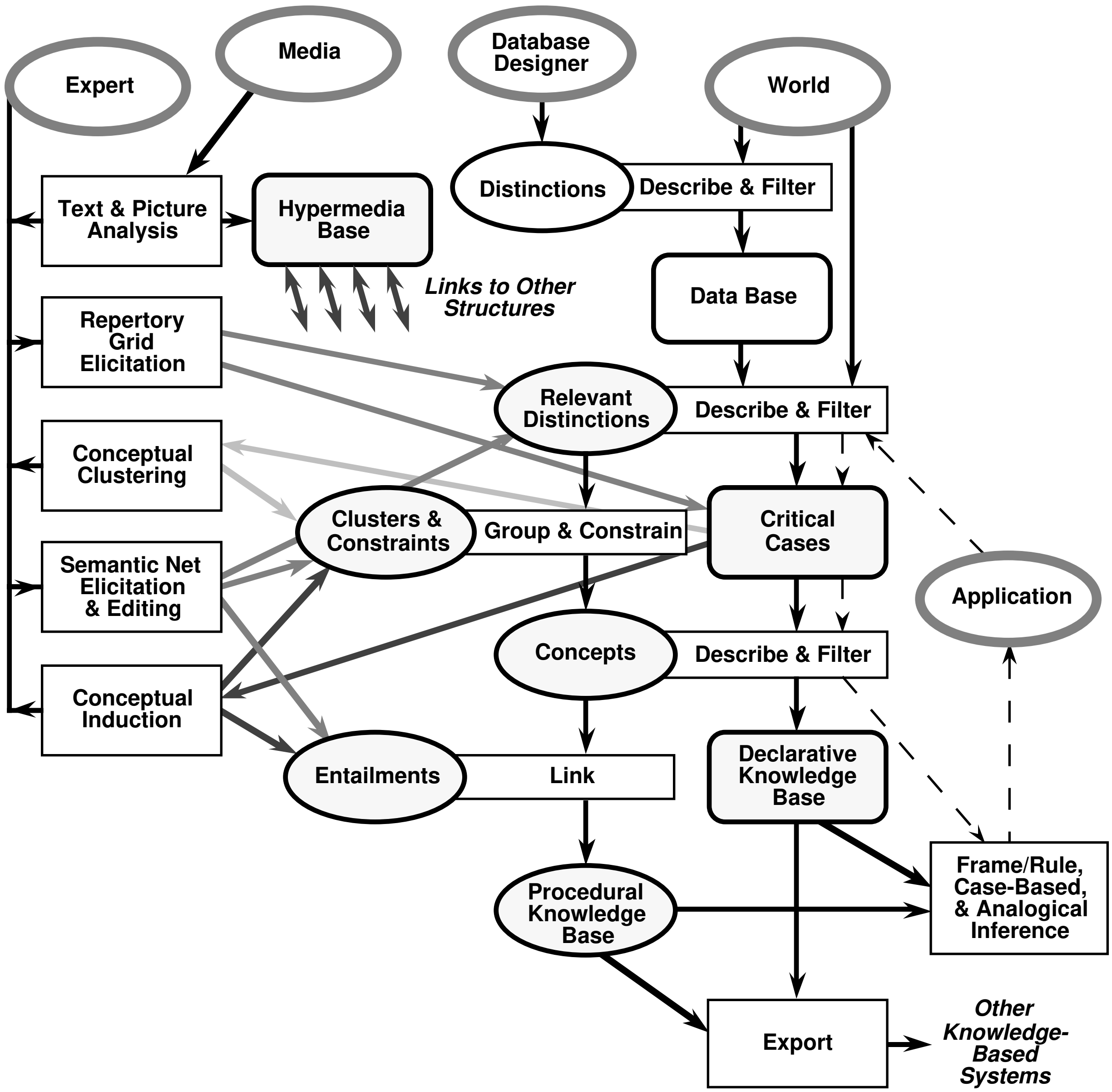

Fig.18 Integrated knowledge acquisition system architecture

At the lower right of Figure 18 the knowledge base has been separated into a procedural one based on the rules and a declarative one based on the cases, and two additional inference paradigms have been added. The center one combines frame/rule inference with *case-based* inference, as, for example, in PROTOS (Bareiss, 1989) The lower one provides *analogical* inference (Leishman, 1989) based on the cases in the declarative knowledge base, as, for example, in AQUINAS' performance module. These mixed paradigms based on both abstract knowledge and concrete cases are seen as the target inference modes. Even though inference is integrated, at the bottom right of Figure 18 an export module is shown since transfer of knowledge bases to specialist performance systems is still seen as an important requirement. We need to integrate performance tools with knowledge acquisition tools for testing and validation, and for long-term support and updating. However, there will continue to be major requirements for stand-alone knowledge acquisition tools and stand-alone performance tools optimized for their primary tasks.

4 CONCLUSIONS

An architecture for knowledge acquisition systems has been proposed that is based upon the integration of existing methodologies, techniques and tools developed within the knowledge acquisition, machine learning, expert systems, hypermedia and knowledge representation research communities. Existing tools have been analyzed within a common framework to show that their integration can be achieved in a natural and principled fashion. A detailed architecture for integrated knowledge acquisition systems has been proposed, much of whose implementation has been illustrated in the examples given in the paper.

While the emphasis of this paper has been on the development of integrated architectures and methodologies, we hope also that the analysis of different approaches to knowledge acquisition within a common framework will be of value to those using individual tools also. Knowledge engineering and knowledge modeling are complex processes requiring major effort and the utilization of a wide variety of sources and knowledge elicitation and analysis techniques. The approaches that have been developed in the past decade are not competetive alternatives, but rather complementary methodologies and tools that need to be used together appropriately.

Our emphasis on integrated architectures should not be taken as advocacy of monolithic tools attempting to encompass all approaches. Much of our own research has been based on the heterogeneous integration of tools that were not originally designed to work together. How the required integration is achieved is an implementation decision, and raises concerns about user interface uniformity and consistency, communication protocols, and knowledge interchange formats which we have addressed elsewhere (Gaines, 1990; Gaines, Linster and Shaw, 1992; Gaines, Rappaport and Shaw, 1992).

This paper has focused on tool integration and knowledge structure integration and has not discussed knowledge level integration through the development of various kinds of generic ontologies. The knowledge level requires a paper of its own for proper treatment and is largely orthogonal to the issues discussed in this paper. For example, the KADS knowledge engineering methodology (Akkermans, van Harmelen, Schreiber & Wielinga, 1992) can be supported through the integrated architecture described, as can approaches based on role-limiting methods (McDermott, J. (1988) and generic tasks (Chandrasekaran, 1988). It is not necessary to tailor the tools to the methodologies provided the integrated architecture provides effective support for modular libraries of knowledge structures that can be used in the initital stages of knowledge acquisition.

ACKNOWLEDGMENTS

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 the research described in this paper, in particular John Boose, Jeff Bradshaw, Bill Clancey, Debbie Leishman, Marc Linster, Alain Rappaport, Doug Skuce, Brian Woodward and other colleagues at the Knowledge Acquisition Workshops world-wide.

REFERENCES

Abu-Hakima & Oppacher, F. (1990) Improving explanations in knowledge-based systems: RATIONALE. **Knowledge Acquisition 2** (4), 301-343.

Akkermans, H., van Harmelen, F., Schreiber, G. & Wielinga, R. (1992) A formalisation of knowledge-level models for knowledge acquisition. **International Journal of Intelligent Systems,** to appear.

Bareiss, R. (1989). **Exemplar-Based Knowledge Acquisition**. New York: Academic Press.

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

Boose, J.H. (1986). **Expertise Transfer for Expert Systems**. Amsterdam: Elsevier.

Boose, J H. (1989) A survey of knowledge acquisition techniques and tools. **Knowledge Acquisition 1** (1), 39-58 (March).

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.

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

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

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

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

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

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.

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.

Gaines, B.R. (1988) Knowledge acquisition systems for rapid prototyping of expert systems. **INFOR, 26**(4), 256-285 (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. **Proceedings of 6th International Workshop on Machine Learning**, pp.156-159. San Mateo, California: Morgan Kaufmann (June).

Gaines, B.R. (1990). Knowledge support systems. **Knowledge-Based Systems 3**(4) 192-203 (December).

Gaines, B.R. (1991a) Empirical investigation of knowledge representation servers: design issues and applications experience with KRS. **AAAI Spring Symposium: Implemented Knowledge Representation and Reasoning Systems**. pp. 87-101. **SIGART Bulletin 2**(3), 45-56.

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

Gaines, B.R. (1991c). An interactive visual language for term subsumption visual languages. **IJCAI'91: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence**. pp.817-823. San Mateo, California: Morgan Kaufmann.
Gaines, B.R. & Boose, J.H., Eds. (1988). **Knowledge Acquisition for Knowledge-Based Systems**. London: Academic Press.
Gaines, B.R. & Linster, M. (1990). Integrating a knowledge acquisition tool, an expert system shell and a hypermedia system. **International Journal of Expert Systems Research and Applications 3**(2) 105-129.
Gaines, B.R., Linster, M. & Shaw, M.L.G. (1992). An integrated knowledge support system. **Proceedings of FGCS'92: International Conference on Fifth Generation Computer Systems**. Tokyo: ICOT, to appear.
Gaines, B.R., Rappaport, A. & Shaw, M.L.G. (1992). Combining paradigms in knowledge engineering. **Data and Knowledge Engineering** to appear.
Gaines, B.R. & Shaw, M.L.G. (1987). Knowledge support systems. **ACM MCC-University Research Symposium.** Austin, Texas: MCC. 47-66.
Gaines, B.R. & Shaw, M.L.G. (1989) Comparing the conceptual systems of experts. **Proceedings of the Eleventh International Joint Conference on Artificial Intelligence.** pp.633-638. Los Angeles, California: Morgan Kaufmann (August).
Gomez, F. & Segami, C. (1990) Knowledge acquisition from natural language for expert systems based on classification problem-solving methods. **Knowledge Acquisition 2** (2), 107-128.
Graesser A.C. & Clark, L.F. (1985). **Structures and Procedures of Implicit Knowledge.** New Jersey: Ablex.
Hausser, R. (1987). **Computation of Language.** Berlin: Springer.
Hughes, S. (1986). Question classification in rule-based systems. **Research and Development in Expert Systems III**. pp. 123-131. Cambridge, UK: Cambridge University Press.
Jones, W.P. (1990) Bringing corporate knowledge into focus with CAMEO. **Knowledge Acquisition 2** (3), 207-239.
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.
Leishman, D. (1989). Analogy as a constrained partial correspondence over conceptual graphs. **Proceedings of KR'89: First International Conference on Principles of Knowledge Representation and Reasoning.** pp.223-234. San Mateo, California: Morgan Kaufmann (May).
Marcus, S. (1987). Taking backtracking with a grain of SALT. **International Journal of Man-Machine Studies, 26**(4), 383-398.
Marcus, S. (Ed.) (1988) **Automating Knowledge Acquisition for Expert Systems**. New York: Kluwer.
McDermott, J. (1988) Preliminary steps toward a taxonomy of problem solving methods. Marcus, S. (Ed) **Automating Knowledge Acquisition for Expert Systems,** Kluwer: Boston.
Michalski, R.S. & Chilausky, R.L. (1980). Knowledge acquisition by encoding expert rules versus computer induction from examples—A case study involving soyabean pathology. **International Journal of Man-Machine Studies 12**, 63-87.
Morik, K. (1987) Sloppy modeling. Morik, K. Ed. **Knowledge Representation and Organization in Machine Learning**. pp.107-134. Berlin: Springer-Verlag.
Monarch, I. & Nirenburg, S. (1987) The role of ontology in concept acquisition for knowledge-based systems. **Proceedings of the First European Workshop on Knowledge Acquisition for Knowledge-Based Systems (EKAW'87)**. Reading University, UK (September).
Motta, E., Eisenstadt, M., Pitman, K. & West, M. (1988) Support for knowledge acquisition in the knowledge engineer's assistant (KEATS). **Expert Systems 5**(1), 6-28.
Motta, E., Rajan, T., Domingue, J. & Eisenstadt, M. (1991) Methodological foundations of KEATS, the Knowledge Engineer's Assistant. **Knowledge Acquisition 3** (1) 21-47.
Quinlan, J.R. (1987) Simplifying decision trees. **International Journal of Man-Machine Studies 27** (3), 221-234 (September).

Rantanen, J.A. (1990) Hypermedia in knowledge acquisition and specification of user interface for KBS: an approach and a case study. **Knowledge Acquisition 2** (3), 259-278.

Rappaport, A. (1987) Multiple-problem subspaces in the knowledge design process. **International Journal of Man-Machine Studies 26**(4), 435-452.

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.

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

Shaw, M.L.G. & Gaines, B.R. (1989). Comparing conceptual structures: consensus, conflict, correspondence and contrast. **Knowledge Acquisition 1**(4), 341-363 (December).

Skuce, D., Shenkang, W. & Beauville, Y. (1989) A generic Knowledge Acquisition Environment for conceptual and ontological analysis. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Fourth AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop**. pp.31-1-31-20. Banff (October).

Swartout, W. R. (1986). Knowledge needed for expert system explanation. **Future Computing Systems, 1**(2), 99-114.

Voß, A., Karbach, W., Drouven, U., Lorek, D. & Schuckey, R. (1990) Operationalization of a synthetic problem. **ESPRIT Basic Research Project P3178 REFLECT Task I.2.1 Report** (July).

Woodward, B. (1990) Knowledge engineering at the front-end: defining the domain. **Knowledge Acquisition 2**(1), 73-94 (March).