THE AUTOMATIC GENERATION OF CLASSES, OBJECTS AND RULES AT THE INTERFACE BETWEEN KNOWLEDGE ACQUISITION TOOLS AND EXPERT SYSTEM SHELLS

Brian R Gaines and Alain T Rappaport

Knowledge Science Institute
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada T2N 1N4

Neuron Data
444 High Street
Palo Alto
CA94301, USA

Abstract: The knowledge acquisition bottleneck has led to the development of a variety of computer-based knowledge acquisition tools which are effective in capturing certain aspects of expert knowledge. However, the effective transfer of this knowledge to a performance engine to give an operational and well-structured expert system is not a trivial problem. This paper details our experience in automatically generating classes, objects and rules for an object-oriented expert system shell from entity-attribute grid data elicited by a knowledge acquisition system. Of particular interest is the way in which object-oriented meta-knowledge structures about rules and hypotheses are generated that model the appropriate control structures for the problem. We have found that the object-oriented approach allows very perspicuous expert system code to be generated that is easy to integrate with other modules and enhance as the system develops.

INTRODUCTION

Figure 1 illustrates the knowledge acquisition process for current systems (Boose & Gaines 1988, Gaines & Boose 1988). The system developer draws upon three major knowledge sources: knowledge encoded in *relevant media*, such as books, journals and videotapes; knowledge available by observing and modeling the *relevant domain*; and knowledge available from discourse with, and observation of, the *relevant community*. It is this last source that is most associated with expert systems as such, although all three sources play significant roles in the development of any practical system.

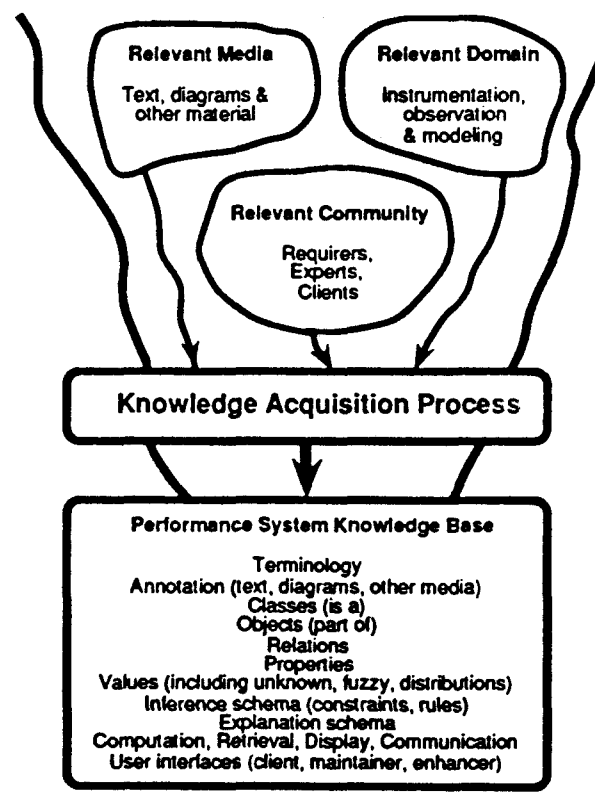

Fig.1 Knowledge acquisition

The content of the *performance system knowledge base* in Figure 1 illustrates the range of technical features which an adequate knowledge-based system shell may be expected to support:

- *Terminology* is often neglected as computationally significant. However, it is particularly important in knowledge-based systems where it is critically important that the terms used in questions, recommendations and explanations are understood by clients in the same way as they are by experts. Obtaining the terminology direct from experts without distortion is an important design consideration for knowledge acquisition systems.
- *Annotation* is used here as a term for non-computational knowledge such as text, diagrams, photographs, videos, and other media, that may be used to impart knowledge to people. Some of it may not be suitable for computer storage but not for inferential processing. Providing appropriate access to such knowledge is important in many knowledge-based systems, and collecting and organizing such knowledge is part of the knowledge acquisition process.

- *Classes, objects, relations, properties* and *values* form the abstract infrastructure of computational knowledge representation currently. Object-orientated knowledge-bases provide a formal conceptual and computational schema which subsumes and supports a wide variety of knowledge representation schema (Shriver & Wegner 1987). These primitives also correspond to natural cognitive operations of classifying, instantiating, relating, attributing and measuring. An important aspect of knowledge acquisition is to determine these cognitive primitives in a domain (Rappaport 1987).

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

- *Inference schema* are systems of constraints that express the relations between values in a knowledge base. These constraints may be thought of as rules expressing the relations and many of them may necessarily be expressed by explicit rules in the knowledge-base. However, the effective organization of the knowledge base in terms of generic classes and relationships should allow the majority of the constraints to be expressed implicitly in a natural way. This is the *framing problem* in knowledge representation for a domain, and an important aspect of knowledge acquisition is to determine a representation schema that minimizes the number of low level rules required.
- *Explanation schema* are forms of argument that generate satisfactory explanations of the inferences made from a knowledge base. Early expert systems generated explanations by displaying the chain of reasoning that was used from premises to conclusions. This approach is useful but does not necessarily 'explain' the conclusions in a way that an expert would, or in a way satisfying a client's need for knowledge. The acquisition of knowledge necessary for adequate explanation goes beyond that adequate for performance.
- *Computation, retrieval, display, communication* and other data processing activities play major roles in most knowledge-based systems. Modern expert system shells make provision for integration with data-processing, simulation, database, graphics and communications programs. The systems analysis for these needs to be integrated with the knowledge acquisition process.
- *User interfaces* are very important in knowledge-based systems as they are generally highly interactive and involve clients with little or no computer experience. The design of the interface in consultation with experts and clients is an important part of knowledge acquisition. Since most knowledge is fairly volatile, knowledge acquisition and updating the knowledge base are important aspects of system operation, and the interface to system maintainers and enhancers also needs careful attention.

TRANSLATING ENTITY-ATTRIBUTE CONCEPTUAL STRUCTURES TO AN O-O SHELL

In this paper we will focus upon the more formal structures in Figure 1 since these provide an inferential framework to which other material may be linked. Computer elicitation of entity-attribute, or *repertory*, grids has proved to be a powerful technique for acquiring the conceptual structures of an expert in a domain (Shaw & Gaines 1983, Boose 1984). It is an extensional approach in which individuals are asked to specify a set of entities in a domain, then make distinctions among them, naming the distinctions and classifying all the specified entities in terms of them. The extension of a distinction determined in this way is only an approximation to the underlying concept since critical entities may be missing in the classification. However, both manual and computer-based elicitation techniques attempt to prompt the individual for additional entities to discriminate between extensionally related distinctions (that is making the same, or similar, classifications). Knowledge acquisition is essentially a negotiation process leading to approximations to conceptual structures that are adequate for some practical purpose such as system development.

Figure 1 shows the grid representation of a set of data used by Cendrowska (1987) to demonstrate the inductive analysis of a set of ophthalmic data. The example is simple and the results well-defined which makes it a good test case. However, the requirements and results are sufficiently complex for the problem to be non-trivial and to demonstrate many major aspects of the object-oriented knowledge structures generated.

The grid in Figure 1 can be regarded as the definition of one class having 24 members (the cases) in terms of class attributes and the values of those attributes for each member. This class can itself be regarded as belonging to a superclass of datasets pertaining to the contact lens decision indexed by the expert whose knowledge is expressed. This is the basic class and object structure for the declarative knowledge involved.

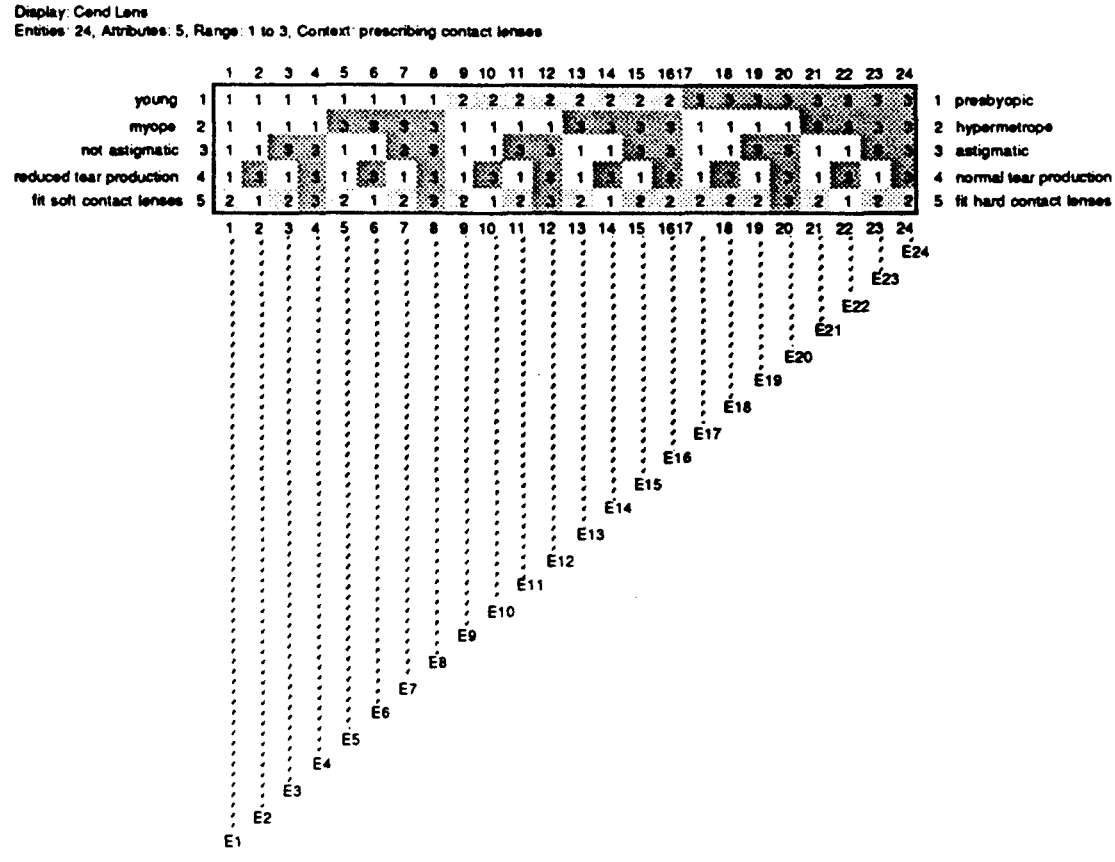

Fig.1 Contact lens decision data in entity-attribute grid form

Procedural knowledge can be derived from the dataset in Figure 1 by using inductive rule generation to link the descriptive attributes to the decision attribute. The resultant rule set (generated by the INDUCT algorithm, Gaines 1989) is shown in Figure 2. The first number after each rule is the percentage correct on the test data—it would be lower if the data were noisy. The second number is the statistical significance of the rule in terms of the percentage probability that it has arisen by chance.

prescription=hypermetrope & astigmatism=not astigmatic & tear production=normal -> lens recommendation=soft 100% 0.904%
age=young & astigmatism=not astigmatic & tear production=normal -> lens recommendation=soft 100% 4.34%
age=pre-presbyopic & astigmatism=not astigmatic & tear production=normal -> lens recommendation=soft 100% 4.34%

tear production=reduced -> lens recommendation=none 100% 0.355%
age=presbyopic & prescription=myope & astigmatism=not astigmatic -> lens recommendation=none 100% 39.1%
age=pre-presbyopic & prescription=hypermetrope & astigmatism=astigmatic -> lens recommendation=none 100% 39.1%
age=presbyopic & prescription=hypermetrope & astigmatism=astigmatic -> lens recommendation=none 100% 39.1%

prescription=myope & astigmatism=astigmatic & tear production=normal -> lens recommendation=hard 100% 0.463%
age=young & astigmatism=astigmatic & tear production=normal -> lens recommendation=hard 100% 2.78%

Fig.2 Decision rules derived from the dataset of Figure 1

Figure 3 shows the automatic translation produced of this rule into a knowledge structure for an object-oriented shell. The first line defines the rule to be itself an object in the class of rules pertaining to the lens recommendation. This meta structure enables the use of rules within a context to be controlled using the natural data structures of the shell. The first two clauses in the rule also refer to meta-knowledge:

- The first clause tests the "use" field of the object "Cendrowska" (automatically defined as a member of the class of experts) to check whether the client wishes this expert's knowledge to influence the decision.
- The second clause tests whether a decision has already been made about the relevant hypothesis. This is used to implement default reasoning in which exceptional cases are tested first.

The final line of the knowledge structure sets up the control structures for the default reasoning. The "inference category" attribute of the rule object is set to a value determined firstly by the probability that the rule is correct and secondly by the highest cost attribute tested (the cost of testing attributes is entered by the expert as meta-information). The overall result is that rules representing exceptions to the usual case are tested first, and, subject to this priority, rules testing expensive attributes are tested last, and only if necessary.

The net result of these structures is that the dataset of Figure 1 generates, without further intervention, a load-and-go expert system that performs exactly as required, testing the attributes in an appropriate order and not asking about the expensive attribute (tear-production) unless necessary. Additionally, the entire process is robust to noisy data and irrelevant attributes generating an operationally correct system even from very corrupt databases (Gaines 1989).

```
(@OBJECT=       Ind_6   (@CLASSES=    lens_recommendation))
(@RULE= Ind_6
  (@LHS=
        (Yes    (Cendrowska.use))
        (IsNot  (<people>.lens_recommendation)  (KNOWN))
        (Is     (<people>.age)  ("pre_presbyopic"))
        (Is     (<people>.prescription)   ("hypermetrope"))
        (Is     (<people>.astigmatism)    ("astigmatic"))
  )
  (@HYPO=       Ind_6.hypothesis)
  (@RHS=
        (Let    (<people>.lens_recommendation)  ("none"))
  )
)
(@SLOT= Ind_6.hypothesis        @INFCAT=        10099;)
```

Fig.3 Translation of a rule into a knowledge structure for an object-oriented shell

CONCLUSIONS

In this short paper it has been possible to give only a brief overview and some technical highlights of the application of an object-oriented approach to the rapid prototyping of knowledge-based systems. The most important lessons to be drawn from our studies to date are:

- Class/property/object/value structures provide a very natural encoding of knowledge acquired through the commonly used entity-attribute grid elicitation approach to knowledge acquisition
- Meta-knowledge about control structures and sources of knowledge is also naturally encoded in the same way and gives flexible and precise control of inferencing
- Default reasoning is simple to implement through such control structures
- The modular encapsulation of the knowledge structures enables this approach to scale upwards to large-scale system development—each "knowledge island" can be structured using a rapid prototyping approach, and then the islands can be linked through the object-oriented control structures.

REFERENCES

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

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

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**, Cornell University, Ithaca.

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

Rappaport, A. (1987b) Cognitive primitives. Boose, J.H. & Gaines, B.R. (Eds) **Proceedings of the Second AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop.** pp.15-0-15-13. Banff

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.

Shriver, B. & Wegner, P. (1987). **Research Directions in Object-Oriented Programming. Cambridge,** Masachusetts: MIT Press.