Eliciting Knowledge and Transferring it Effectively to a Knowledge-Based System

Brian R. Gaines and Mildred L. G. Shaw
Knowledge Science Institute
University of Calgary
Alberta, Canada T2N 1N4
gaines@cpsc.ucalgary.ca, mildred@cpsc.ucalgary.ca

Index Terms: Knowledge acquisition, expert systems, expertise transfer, empirical induction

Abstract

The knowledge acquisition bottleneck impeding the development of expert systems is being alleviated by the development of computer-based knowledge acquisition tools. These work directly with experts to elicit knowledge, and structure it appropriately to operate as a decision support tool within an expert system. However, the elicitation of expert knowledge and its effective transfer to a useful knowledge-based system is complex and involves a diversity of activities. This paper illustrates the complete development of a decision support system using knowledge acquisition tools. The example is simple enough to be completely analyzed but exhibits enough real-world characteristics to give significant insights into the processes and problems of knowledge engineering.

1 Introduction

Knowledge acquisition for expert system development has come to be termed *knowledge engineering*, following Feigenbaum's (1980) use of the term to describe the reduction of a large body of knowledge to a precise set of facts and rules. The term *knowledge engineer* has come to be used for the person responsible for such system development, and concise job descriptions for knowledge engineers have been given:

> *"Knowledge acquisition is a bottleneck in the construction of expert systems. The knowledge engineer's job is to act as a go-between to help an expert build a system. Since the knowledge engineer has far less knowledge of the domain than the expert, however, communication problems impede the process of transferring expertise into a program. The vocabulary initially used by the expert to talk about the domain with a novice is often inadequate for problem-solving; thus the knowledge engineer and expert must work together to extend and refine it. One of the most difficult aspects of the knowledge engineer's task is helping the expert to structure the domain knowledge, to identify and formalize the domain concepts." (Hayes-Roth, Waterman & Lenat, 1983)*

Thus, the basic model for knowledge engineering has been that the knowledge engineer mediates between the expert and knowledge base, eliciting knowledge from the expert, encoding it for the knowledge base, and refining it in collaboration with the expert to achieve acceptable performance. Figure 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:

- The knowledge engineer interviews the expert to elicit his or her knowledge;

- The knowledge engineer encodes the elicited knowledge for the knowledge base;
- The shell uses the knowledge base to make inferences about particular cases specified by clients;
- The clients use the shell's inferences to obtain advice about particular cases.

However, the knowledge engineer in the role of an intermediary between the expert and the knowledge-based systems may create as many problems as he or she solves (Gaines, 1987a). The computer itself is an excellent tool for helping the expert to structure the knowledge domain, and, in recent years research on knowledge acquisition has focused on the development of computer-based acquisition tools (Boose & Gaines, 1988; Gaines & Boose, 1988; Boose, 1989). Many such tools are designed to be used directly by the expert with the minimum of intervention by the knowledge engineer, and emphasize facilities for visualizing the domain concepts. The objective of systems such as PLANET (Shaw & Gaines, 1983, 1986, 1987a), ETS (Boose, 1984, 1986), MOLE (Eshelman, Ehret, McDermott & Tan, 1987), SALT (Marcus, 1987), KITTEN (Shaw & Gaines, 1987b), KNACK (Klinker Bentolila, Genetet, Grimes & McDermott, 1987), KRITON (Diederich, Ruhmann & May, 1987), OPAL (Musen, Fagan, Combs & Shortliffe, 1987), AQUINAS (Boose & Bradshaw, 1987) and KSS0 (Gaines, 1987b; Gaines & Shaw, 1987) is to expedite the process of acquiring knowledge and transferring it to knowledge-based systems.

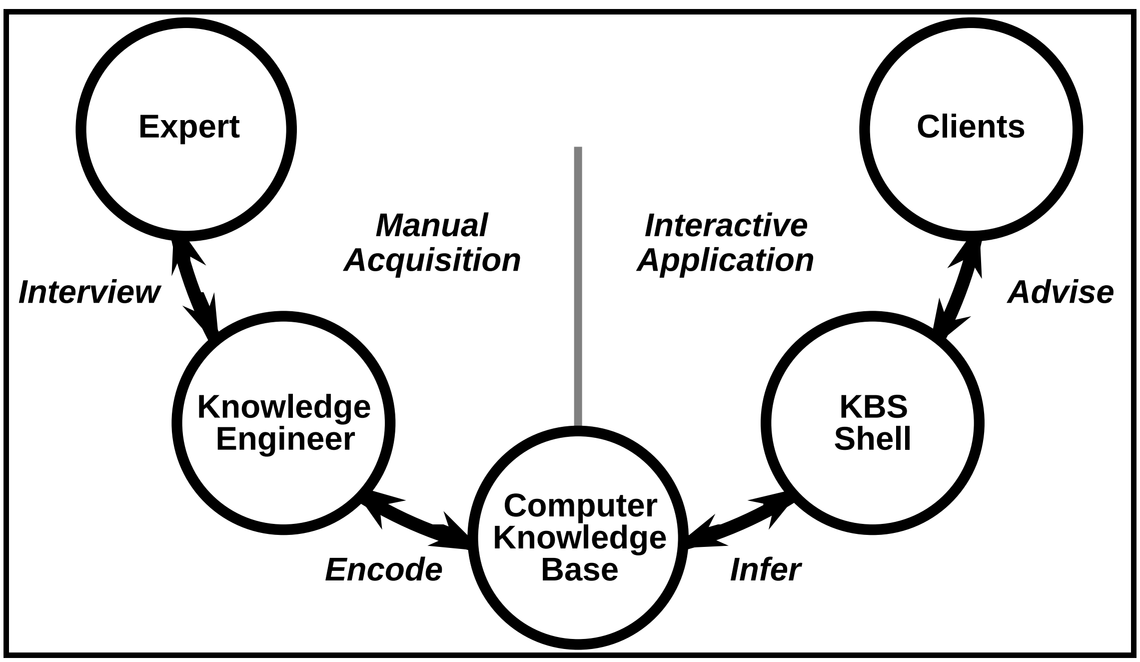


Figure 1 Basic knowledge engineering

Interactive knowledge acquisition and encoding tools can greatly reduce the need for the knowledge engineer to act as an intermediary but, in most applications, they leave a substantial role for the knowledge engineer. As shown in Figure 2, knowledge engineers have responsibility for:

- Advising the experts on the process of interactive knowledge elicitation;
- Managing the interactive knowledge acquisition tools, setting them up appropriately;

- Editing the unencoded knowledge base in collaboration with the experts;
- Managing the knowledge encoding tools, setting them up appropriately;
- Editing the encoded knowledge base in collaboration with the experts;
- Validating the application of the knowledge base in collaboration with the experts;
- Setting up the user interface in collaboration with the experts and clients;
- Training the clients in the effective use of the knowledge base in collaboration with the expert by developing operational and training procedures.

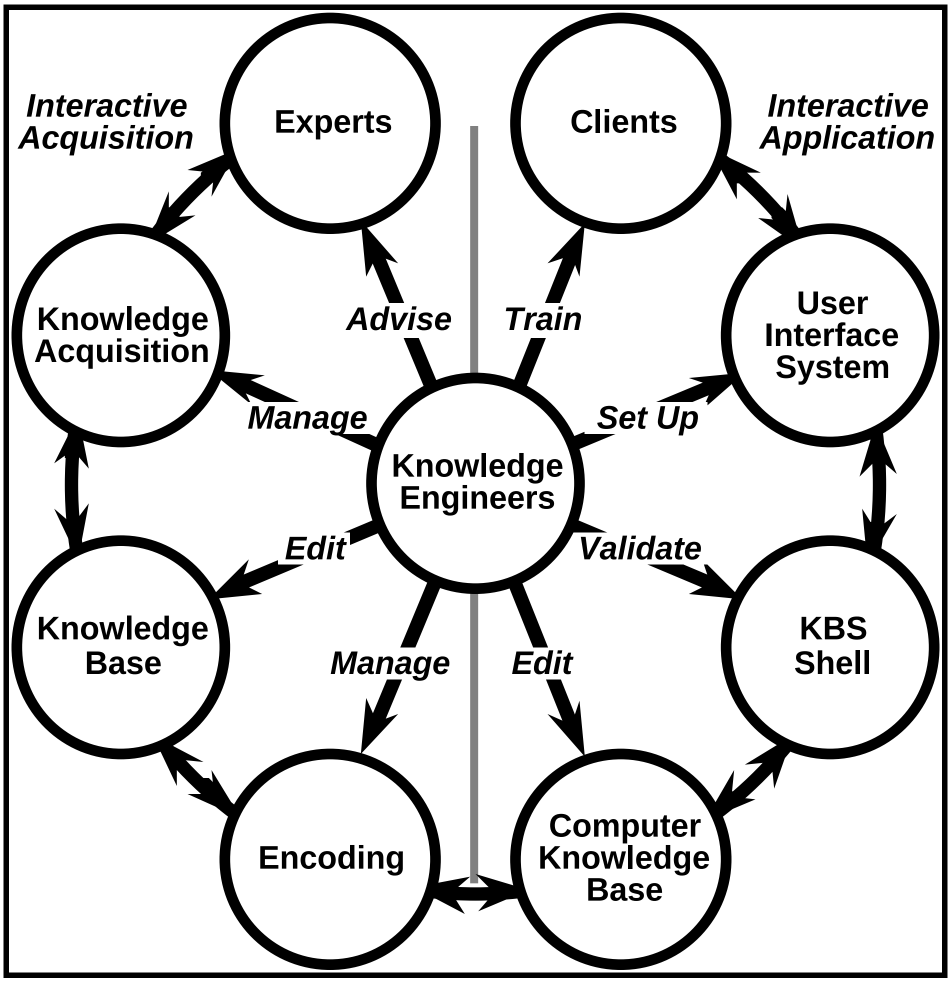


Figure 2 Knowledge engineering with interactive tools

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

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

Figure 2 specifies 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 specified 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 (Boose, 1987; Shaw & Woodward, 1987; Shaw & Gaines, 1988; Gaines & Shaw, 1989).

Validation is shown in Figure 2 as a global test of the shell in operation with the knowledge base, that is of overall inferential performance. However, validation may also be seen as a local feature of each step of the knowledge engineers' activities: the experts' proper use of the tools needs validation; the operation of the tools themselves needs validation; the resultant knowledge base needs validation; and so on. Attention to quality control through validating each step of the knowledge acquisition process is key to effective system development.

2 A Knowledge Acquisition Tool

How do interactive knowledge acquisition tools operate? Figure 3 shows the architecture of one system that integrates a wide range of knowledge acquisition methodologies. The system is written in Pascal and runs on the Apple Macintosh family of computers to provide a highly interactive and graphic knowledge acquisition environment. At the heart of the system is an object-oriented knowledge base in which knowledge is formally represented as a multiple-inheritance structure of classes, objects, properties, values, and relations. Such a structure generalizes the entity-attribute datasets used in several early knowledge acquisition systems and has proved both general and powerful in a variety of applications.

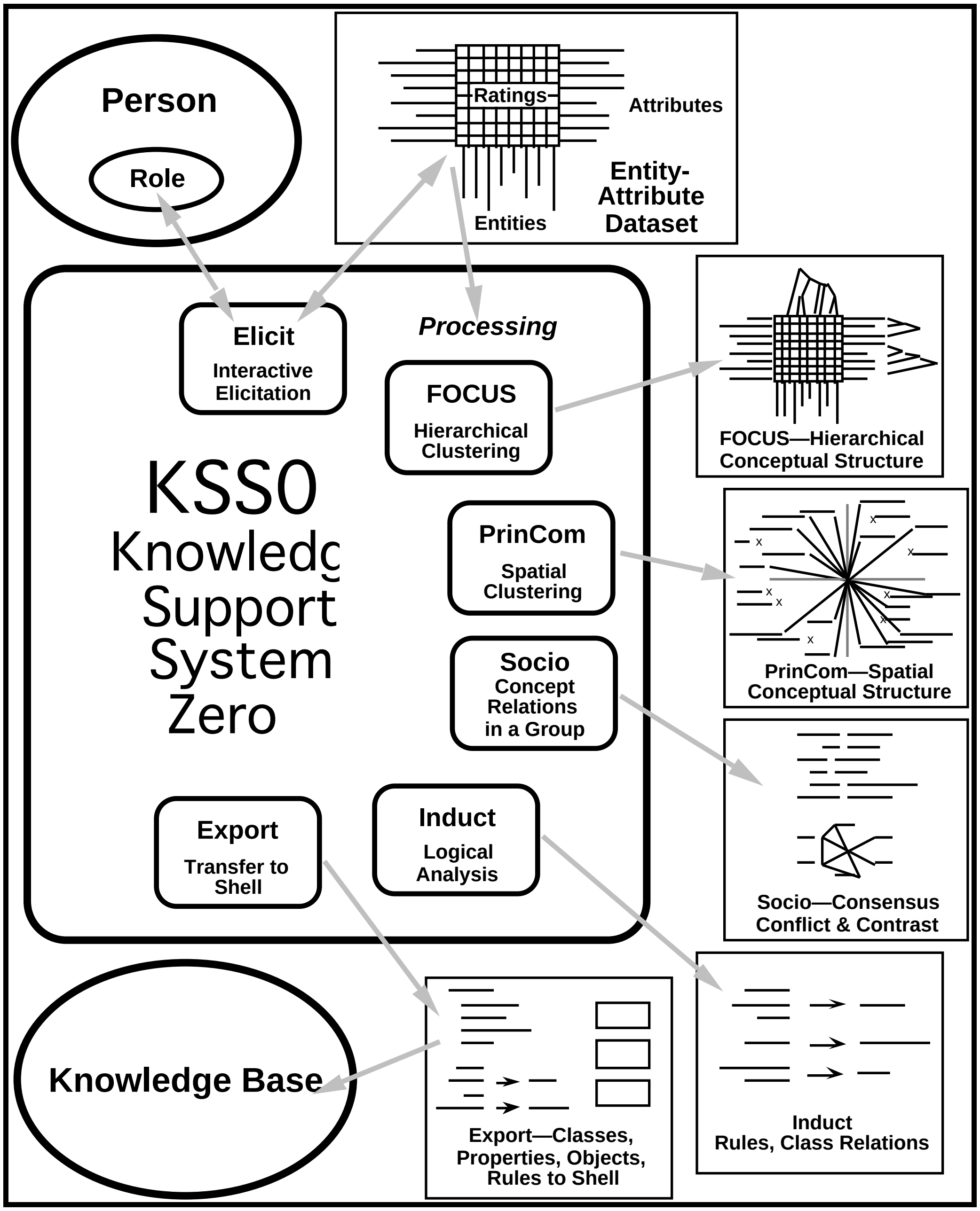


Figure 3 KSS0, an integrated knowledge acquisition system

The elicitation tools are based on Shaw's (1980, 1981) computer-based interviewing techniques extended through the use of graphical rather than numerical data entry. The visualization and direct manipulation of knowledge structures through the graphic and click-and-drag facilities of modern workstations is an important development at the user interface giving experts, knowledge engineers and clients improved access to the knowledge structures.

- *Interview* accepts specifications of cases within a sub-domain and provides an interactive graphical elicitation environment within which the experts can distinguish cases to derive their attributes. The resultant class is continuously analyzed to provide feedback prompting the expert to enter further cases and attributes.

The visualization tools consist of an interactive interface to represent the abstractions derived from those cases in terms of hierarchical clusters using Shaw's (1980) *FOCUS* algorithm, and relational diagrams such as a non-hierarchical conceptual maps derived through principal components analysis (Slater, 1976, 1977). The objectives are to validate the raw domain knowledge and suggest further structure at a higher level through interactive topological induction (Rappaport & Gaines, 1988):

- *FOCUS* hierarchically clusters cases and attributes within a sub-domain prompting the experts to add higher-level attributes structuring the domain.
- *PrinCom* spatially clusters cases and attributes within a sub-domain prompting the experts to add higher-level attributes structuring the domain.

The group comparison tools consist of an interactive interface to represent the relations between the terminologies and conceptual systems of different experts, or experts and clients. The objectives are to determine the consensus, conflict, correspondence and contrast between different conceptual systems (Shaw & Gaines, 1988):

- *Socio* compares the structures for the same sub-domain generated by different experts, or the same expert at different times or from varying perspectives.

The inductive part consists in the derivation of constraints within the conceptual structures through logical entailment analyses (Gaines and Shaw, 1980, 1986; Quinlan, 1987; Cendrowska, 1987; Gaines, 1989a). The objective is to suggest further structure at a higher level that translates into class inclusions or rules in the expert system shell:

- *Induct* induces logical entailments enabling the attributes of an case or the evaluations of a decision-making situation in a domain, to be derived from other attributes.

The generative part consists in the transformation of the knowledge analysis made by the previous tools into formalisms understandable by knowledge-based system shells such as *NEXPERT* (Rappaport, 1987a,b) and *Babylon* (Christaller, Primio & Voss, 1989):

- *Export* formats the specifications of sub-domains as classes, of cases as objects, of attributes as properties, and of entailments as methods, and transfers them to the performance tool.

The next section shows the use of the induction and export tools in KSS0 to create a computational knowledge base from entity-attribute datasets giving the required performance in an expert system shell. The final sections illustrates the use of the interactive interviewing and visualization tools in eliciting an adequate dataset.

3 A Sample Dataset

The dataset used is this paper is taken from a paper by Cendrowska (1988) on the inductive analysis of a set of ophthalmic data. The problem is to infer the type of lens to be prescribed by determining relevant attributes of the client. The example, shown in Figure 4, is simple and the results well-defined which makes it a good test case for the basic techniques. However, the requirements and results are sufficiently complex for the test case to be non-trivial and to demonstrate many major features of the knowledge acquisition tools and their application.

Case	Age	Prescription	Astigmatism	Tear Production	Lens
1	young	myope	not astigmatic	reduced	none
2	young	myope	not astigmatic	normal	soft
3	young	myope	astigmatic	reduced	none
4	young	myope	astigmatic	normal	hard
5	young	hypermetrope	not astigmatic	reduced	none
6	young	hypermetrope	not astigmatic	normal	soft
7	young	hypermetrope	astigmatic	reduced	none
8	young	hypermetrope	astigmatic	normal	hard
9	pre-presbyopic	myope	not astigmatic	reduced	none
10	pre-presbyopic	myope	not astigmatic	normal	soft
11	pre-presbyopic	myope	astigmatic	reduced	none
12	pre-presbyopic	myope	astigmatic	normal	hard
13	pre-presbyopic	hypermetrope	not astigmatic	reduced	none
14	pre-presbyopic	hypermetrope	not astigmatic	normal	soft
15	pre-presbyopic	hypermetrope	astigmatic	reduced	none
16	pre-presbyopic	hypermetrope	astigmatic	normal	none
17	presbyopic	myope	not astigmatic	reduced	none
18	presbyopic	myope	not astigmatic	normal	none
19	presbyopic	myope	astigmatic	reduced	none
20	presbyopic	myope	astigmatic	normal	hard
21	presbyopic	hypermetrope	not astigmatic	reduced	none
22	presbyopic	hypermetrope	not astigmatic	normal	soft
23	presbyopic	hypermetrope	astigmatic	reduced	none
24	presbyopic	hypermetrope	astigmatic	normal	none

Figure 4 Contact lens dataset

Cendrowska's paper argues against the utility of decision trees in developing expert systems, and for the use of modular rules. Her main arguments are that the use of trees produced by algorithms such as *ID3* and its extensions do not produce effective rule sets for expert systems because:

i) the rule sets are unnecessarily large and complex;

ii) the rule sets test conditions unnecessarily causing the expert system to ask redundant, and potentially expensive, questions.

An *ID3* decision tree analysis of the contact lens dataset in Figure 4 is shown in Figure 5.

- **reduced tear production:** ***none***
- **normal tear production**
  - **astigmatic**
    - **myope:** ***hard***
    - **hypermetrope**
      - **young:** ***hard***
      - **pre-presbyopic:** ***none***
      - **presbyopic:** ***none***
  - **not astigmatic**
    - **hypermetrope:** ***soft***
    - **myope**
      - **young:** ***soft***
      - **pre-presbyopic:** ***soft***
      - **presbyopic:** ***none***

Figure 5 ID3 analysis of dataset

The ID3 tree may be converted to a set of rules as shown in Figure 6 by tracing the paths to each decision. Note that every one of these rules tests for tear production. Cendrowska notes that this is an expensive and time-consuming test, and that it is unnecessary in a number of significant cases. That is, if the decision rules of Figure 6 were transferred to the knowledge base of an expert system, the system would request data about some patients that was, in practice, unnecessary for it to reach correct conclusions.

reduced tear production: ***none***
normal tear production & astigmatic & myope : ***hard***
normal tear production & astigmatic & hypermetrope & young: ***hard***
normal tear production & astigmatic & hypermetrope & pre-presbyopic: ***none***
normal tear production & astigmatic & hypermetrope & presbyopic: ***none***
normal tear production & not astigmatic & hypermetrope: ***soft***
normal tear production & not astigmatic & myope & young: ***soft***
normal tear production & not astigmatic & myope & pre-presbyopic: ***soft***
normal tear production & not astigmatic & myope & presbyopic: ***none***

Figure 6 Rules from ID3 analysis

Cendrowska proposes a new algorithm for rule induction called *Prism* which generates rules directly without going through decision trees. She uses the ophthalmic data to show that Prism solves this problem correctly, as shown in Figure 7, and satisfies her expressed requirements. That is, Figure 7 gives an alternative set of rules that are completely equivalent to those of Figure 6 but where the last three rules do not test for tear production. It should be noted that Quinlan (1987), the founder of the *ID3* line of inductive techniques, has shown that the problems of redundant tests in rule sets may be overcome by post-processing decision trees. In particular, his techniques cope with noisy data whereas *Prism* can be used only with error-free data.

myope & astigmatic & normal tear production: ***hard***
young & astigmatic & normal tear production: ***hard***

hypermetrope & not astigmatic & normal tear production: ***soft***
young & not astigmatic & normal tear production: ***soft***
pre-presbyopic & not astigmatic & normal tear production: ***soft***

reduced tear production: ***none***
presbyopic & myope & not astigmatic: ***none***
pre-presbyopic & hypermetrope & astigmatic: ***none***
presbyopic & hypermetrope & astigmatic: ***none***

Figure 7 Prism analysis of dataset

The *Induct* algorithm used in KSS0 is based on one by Gaines (1989a) which generates rules directly and filters them statistically to cope with data that contains errors. It is able to reproduce Cendrowska's analysis of the ophthalmic data exactly. In addition it is able to produce the same analysis with extremely noisy data. It will also process data effectively in which many features are unknown, and this may be used to allow experts to enter rules directly as generalized cases. That is, a rule may be treated as a case in which some of the values of the attributes are unknown because they could have any value.

It is not trivial to transfer the rules which Cendrowska takes as a solution to the ophthalmic problem to an expert system in such a way that it has the behavior she requires, that is, it should not ask irrelevant questions. This means in practice that the last three rules in Figure 7 should be tested first. The appropriate control strategies to ensure this have to be transferred to the expert system also, and the *Export* tool in KSS0 does this in a generally applicable way using appropriate control mechanisms in the shell.

It is also non-trivial to transfer the rules in such a way that they are applicable to a wide range of problem types that may be not be defined at the time the knowledge acquisition takes place. One wishes to transfer the knowledge (which includes some control structures) to the expert system in such a way that it forms a re-usable module, not generate application-specific code. The *Export* tool does this using appropriate classes, objects and pattern-matching rules in the shell.

4 Analyzing the Contact Lens Dataset

We will first assume that the dataset of Figure 4 already exists and the knowledge engineer wishes to analyze it and transfer the resulting knowledge structures to a shell. He or she may create a dataset by using the interactive elicitation facilities to enter the cases, attributes, values and control information such as the decision attribute.

Running the *Induct* tool with this dataset generates the rules shown in Figure 8 which are logically the same as those in Figure 7. The two numbers following each rule are the percentage of cases for which the rule is correct, and the percentage probability that the rule might arise by chance, respectively. For example, the first rule in Figure 8 is 100% correct and has a 0.904% probability of arising by chance.

```
Rule 1: prescription=hypermetrope & astigmatism=not astigmatic & tear production=normal -> lens
    recommendation=soft 100% 0.904%
Rule 2: age=young & astigmatism=not astigmatic & tear production=normal -> lens
    recommendation=soft 100% 4.34%
Rule 3: age=pre-presbyopic & astigmatism=not astigmatic & tear production=normal -> lens
    recommendation=soft 100% 4.34%

Rule 4: tear production=reduced -> lens recommendation=none 100% 0.355%
Rule 5: age=presbyopic & prescription=myope & astigmatism=not astigmatic -> lens
    recommendation=none 100% 39.1%
Rule 6: age=pre-presbyopic & prescription=hypermetrope & astigmatism=astigmatic -> lens
    recommendation=none 100% 39.1%
Rule 7: age=presbyopic & prescription=hypermetrope & astigmatism=astigmatic -> lens
    recommendation=none 100% 39.1%

Rule 8: prescription=myope & astigmatism=astigmatic & tear production=normal -> lens
    recommendation=hard 100% 0.463%
Rule 9: age=young & astigmatism=astigmatic & tear production=normal -> lens recommendation=hard
    100% 2.78%
```

Figure 8 *Induct* analysis of dataset in KSS0

The dataset of Figure 4 is without conflicts in that a given set of values of the attributes always corresponds to the same decision. Hence all the rules have 100% probability of being correct. As shown later, if conflicting data is entered, *Induct* may generate rules that are less than 100% correct. The required maximum probability that a rule might be due to chance is a parameter which can be set in the *Induct* dialog. It corresponds exactly to the normal “level of significance” of conventional statistics, for example, that a result “is significant at the 5% level.”

The adjustment of this parameter is particularly interesting because the datasets elicited from experts may be very much smaller than those required for statistical testing yet still be effective in terms of knowledge transfer. The essence of expertise transfer techniques is that the expert’s expertise makes available high quality data. That is, experts can often provide a minimal set of relevant attributes and a complete set of critical cases with correct decisions. In these circumstances the significance level can be set to accept rules which from a statistical point of view might well arise by chance. The statistical logic is that the data is correct and representative of many independent cases, and hence it could be entered in exactly the same form a number of times. This repetition of correct data would make the statistical significance of the results as great as required. Hence it is usual to start with the significance in *Induct* at 100%.

In practice, experts may enter incomplete sets of attributes or incorrect cases and some of the rules generated by *Induct* will be spurious. The statistical significance measure is then usually a very good indicator of the most suspect rules. The expert can inspect the rules and see if they make sense. The knowledge engineer can then set the significance threshold in *Induct* to exclude the spurious rules.

The contact lens data is complete and correct so that the rules in Figure 8 do not need to be tested for significance. Even so, the significance level indicates those rules which are most difficult to conclude from the data. For example the fifth through seventh rules have 39.1% probability of arising by chance if the dataset was a random sample. This is because they correspond to three critical exceptions to a much simpler rule set: If the tear production is normal then if the client is astigmatic fit hard lenses or if not astigmatic fit soft. The three exceptions to this simple rule set are exhibited only once each in the lens dataset and hence the rules to cover them are not validated very well. This is what one might expect in that exceptions by definition are infrequent events.

5 Transferring the Contact Lens Dataset

Running the Export tool with the contact lens dataset generates a knowledge base which may be loaded and run in shells such as *NEXPERT* (Rappaport, 1987a,b) and Babylon (Christaller, Primio & Voss, 1989) as a complete expert system. Figure 9 shows three class definitions for the *NEXPERT* shell which classify the rules produced by *Induct* as a class concerned with determining the value of the lens recommendation attribute, which is a sub-class of those concerned with the domain of contact lens prescription, which is a subclass of the general class of Rules. The class of rules has one property, a Boolean hypothesis, which is inherited by its subclasses. This class structure makes the rules and their hypotheses themselves knowledge structures subject to the addition of further properties as required for more complex control structures. The objective is for the tool to generate model code, not just output that will run, but knowledge that is well-structured for further development and embedding in larger systems.

```
(@CLASS=	Rules	(@SUBCLASSES=	contact_lens_prescription)
	(@PROPERTIES=
		hypothesis	@TYPE=Boolean;
	)
)

(@CLASS=	contact_lens_prescription
	(@SUBCLASSES=
		lens_recommendation
	)
)

(@CLASS=	lens_recommendation)
```

Figure 9 Rule class and subclass definitions from contact lens dataset

Figure 10 shows the top level control rule generated. It uses the class of Rules defined in Figure 9 in a pattern-matching clause that tests whether the hypothesis of any rule is true. The global hypothesis of this rule is also put on the list of suggestions in the shell so that this rule may be triggered very simply.

```
(@OBJECT=	Global_Hypothesis)
(@RULE=	Global
	(@LHS=
		(Yes	(<Rules>.hypothesis))
	)
	(@HYPO=	Global_Hypothesis)
)

(@GLOBALS=	@SUGLIST=	Global_Hypothesis;)
```

Figure 10 The top level control rule

Figure 11 shows the class definition that transfers the primary knowledge structure about the class of cases defined by the dataset. It is followed by meta-slot definitions, of which one is shown, giving the prompts that the shell should use in requesting the values of attributes. These are followed by the instantiation of one object in the class that may be used as a test case.

```
(@CLASS=	people
	(@PROPERTIES=
		age	@TYPE=String;
		prescription	@TYPE=String;
		astigmatism	@TYPE=String;
		tear_production	@TYPE=String;
		lens_recommendation	@TYPE=String;
	)
)

(@SLOT=	people.age	@PROMPT="Double click on the age from the list below
	that applies to @SELF";)

(@OBJECT=	person	(@CLASSES=	people))
```

Figure 11 Case class definition, prompting and instantiation

Figure 12 shows the first of the rules of Figure 7 generated by *Induct* translated into *NEXPERT* knowledge base format. The first line instantiates a rule object for the rule. The left hand side tests the premise of the rule with an added test to determine whether the value of the attribute to be set on the right hand side of the rule is already known.

```
(@OBJECT=     Ind_1   (@CLASSES=    lens_recommendation))
(@RULE=       Ind_1
        (@LHS=
              (IsNot  (<people>.lens_recommendation)        (KNOWN))
              (Is     (<people>.prescription)         ("hypermetrope"))
              (Is     (<people>.astigmatism)          ("not_astigmatic"))
              (Is     (<people>.tear_production)      ("normal"))
        )
        (@HYPO=       Ind_1.hypothesis)
        (@RHS=
              (Let    (<people>.lens_recommendation)        ("soft"))
        )
)
(@SLOT=       Ind_1.hypothesis        @INFCAT=      10098;)
```

Figure 12 Induced rule transferred to *NEXPERT*

The meta-slot INFCAT (inference category) of the hypothesis for the rule is set in the final line to 10098. This is generated from the probability of the rule being correct times 100 plus the lowest priority of the attributes tested in the left hand side of the rule (98 for the expensive to test attribute, tear production—rest are set to 99). When executing the top level control rule defined in Figure 8, the shell prioritizes the pattern-matching based on the value of this meta-slot, testing hypotheses with the highest values first. Thus rules with the greatest probability of being correct are tested first, and among those of equal probability the ones that require the values of only higher priority attributes are tested first.

This priority system, together with the test for the value of an attribute being known, ensures that values of attributes are set by the rules most likely to be correct and that the values of low priority attributes are not asked unnecessarily. In particular, this strategy implements default reasoning since default rules are usually of high occurrence but with some probability of being incorrect, while exception rules are of lower occurrence but higher probability of being correct.

6 Consulting the Contact Lens Dataset

Figure 13 shows the resulting object and rule overviews when the knowledge base described in the previous section is loaded into the shell. The top class and object in the object overview is that defined in Figure 9, the class "people" and the instance of it, "person." The two objects below it are the global and setup hypotheses defined in Figure 10. The lowest class, two sub-classes, and nine instances are the rule classes and rules defined in Figures 8 and 9. The top rule in the rule overview is the top level control rule defined in Figure 10 and the nine rules below it are the ones induced for the contact lens dataset shown in Figure 8 whose definition is exemplified in Figure 11.

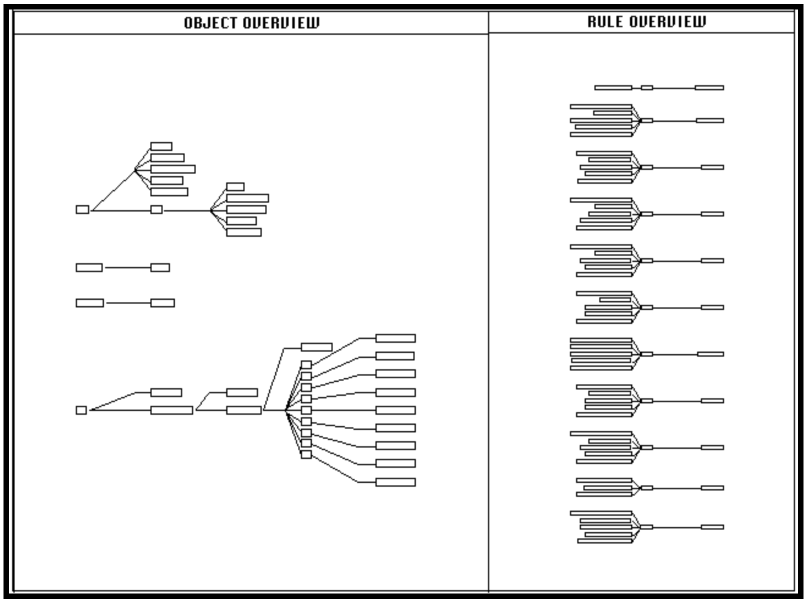


Figure 13 Object and rule overviews in NEXPERT

Figure 14 illustrates a consultation with the knowledge base started against a background that shows the object window open at the top right with the values of the instance “person” shown, and the rule window open at the bottom with the top level rule and the first rule shown. Note that everything on this screen has been generated by the knowledge acquisition and expert system tools from the original dataset, including prompt strings, options, test case and rules. In essence the dataset has been compiled in two stages by the two tools to become an expert system. The whole process is automatic and involves no manual editing, although both expert and knowledge engineer can interact to control and modify the compilation as they see fit. The contact lens example is an exceptionally perfect dataset that requires no intervention.

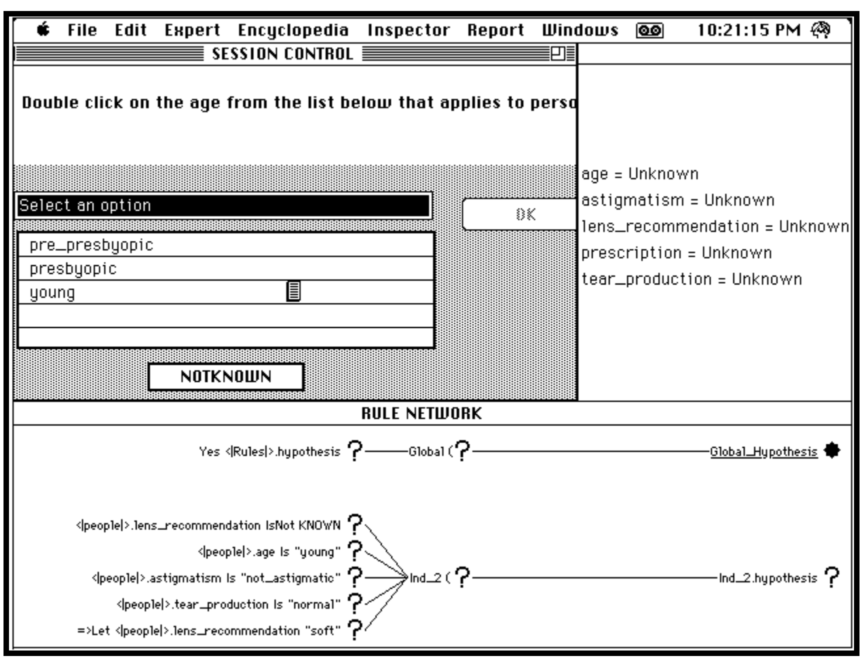


Figure 14 Start of consultation

Figure 15 shows the continuation of the consultation begun in Figure 14. The client clicked “young” in answer to the first question, “myope” in answer to a second question about the prescription, and is now being asked about the astigmatism. As can be seen in the rule window at the bottom, it is actually the first rule that is now being tested.

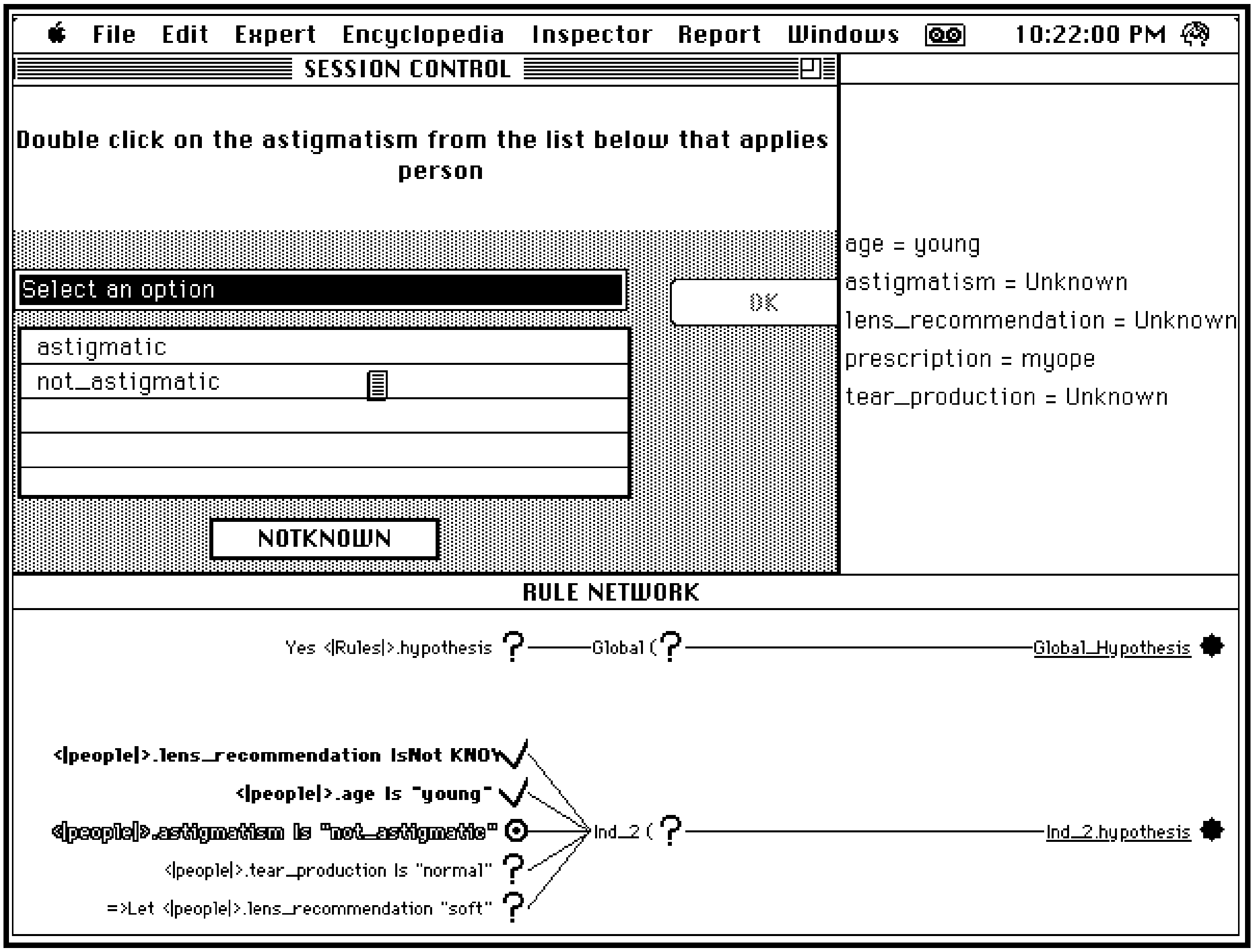
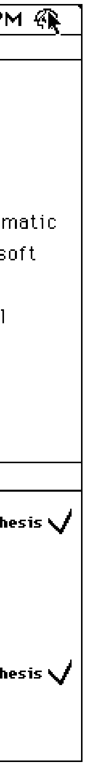
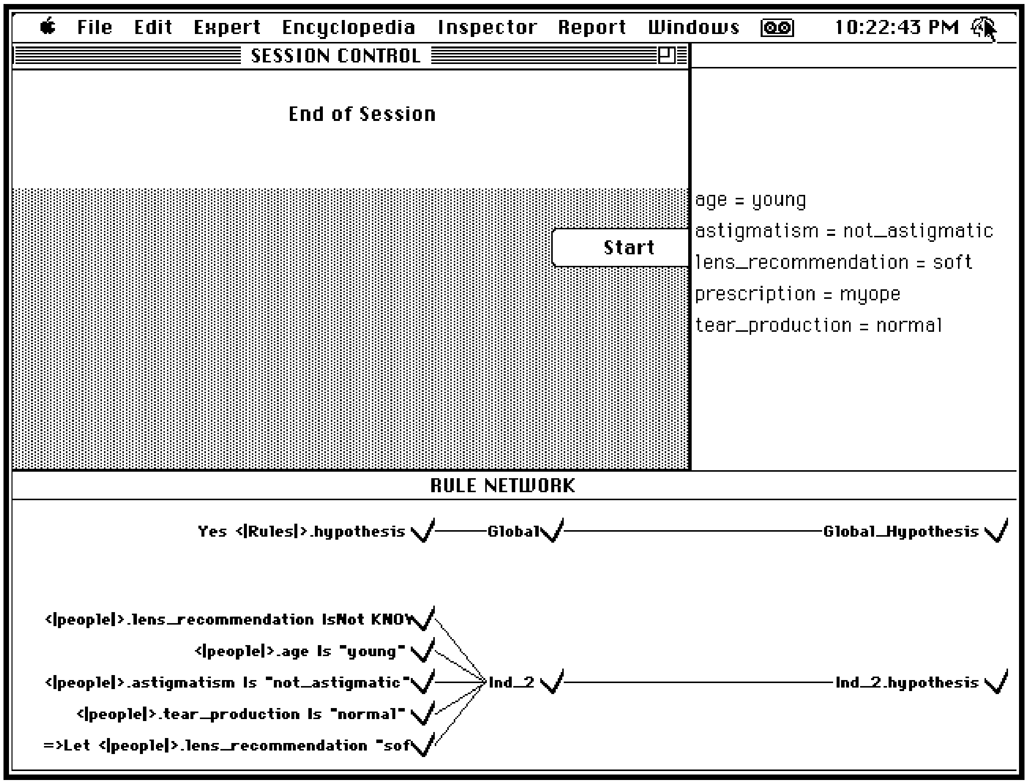


Figure 15 Part way through a consultation

Figure 16 shows the completion of the consultation when the answers “not astigmatic” and “normal” tear production have been given. The rule shown has been successful and has set the value of lens recommendation to “soft.” When this consultation is run with the rules shown rule 5 in Figure 8 is tested first since it is the first one with the highest inference category of 10099. It asks for the age and proceeds only if is “presbyopic.” If not, then rule 6 will be tested and, if the age is “pre-presbyopic”, it will asks for the prescription. Note that if any of rules 5, 6 or 7 succeed then the value of the lens recommendation will be set to “none” without the value of tear production being requested.



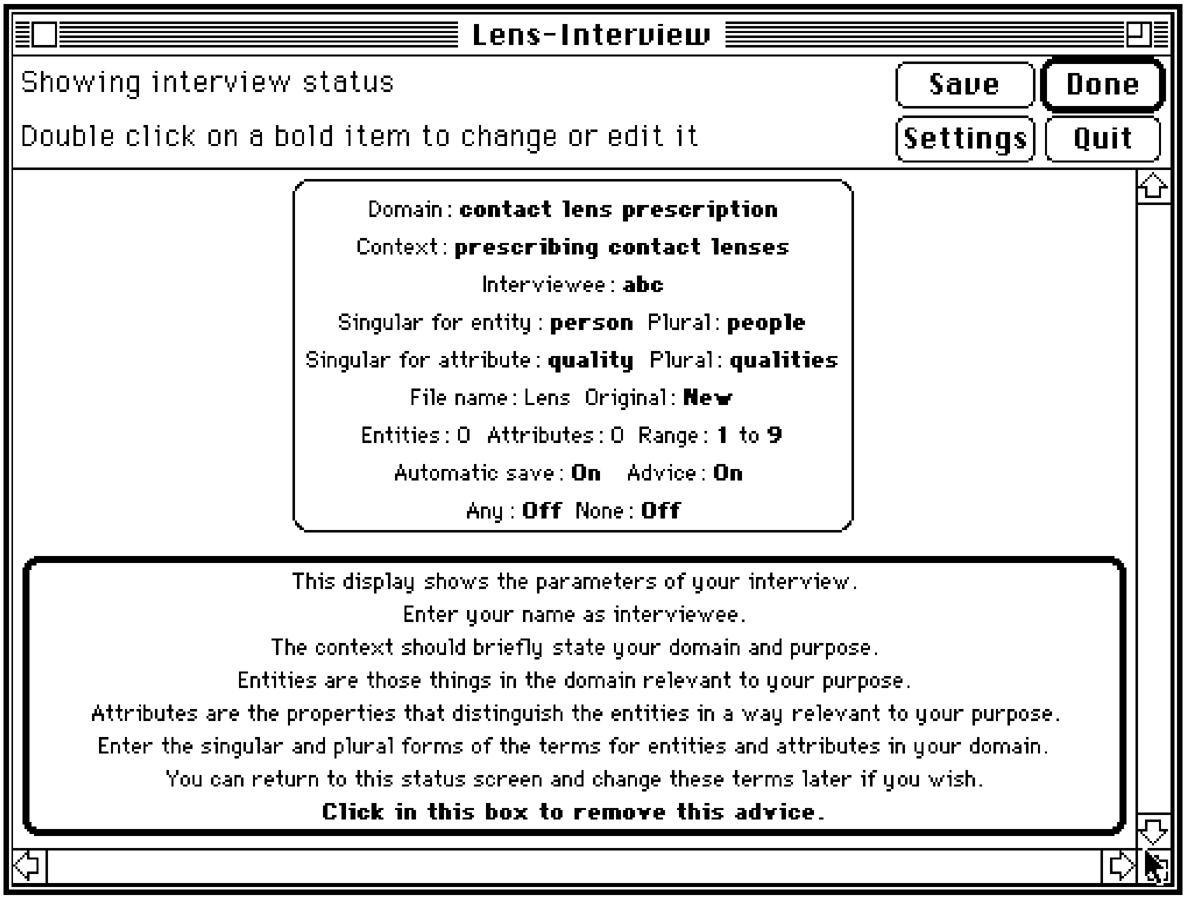


16 Conclusions of consultation

7 Eliciting a Dataset for the Contact Lens Problem

The discussion up to this point has been concerned with analyzing the contact lens dataset, transferring it to an expert system shell and running a consultation. However, the *Interview* facilities in the knowledge acquisition tool are designed to elicit the dataset at a stage when the expert may not be able to state the relevant attributes and a complete and consistent set of critical cases as in Figure 4.

Assume that the expert has not already worked out the table of Figure 1 and that the acquisition tool is being used to help him or her to describe the relevant attributes and critical cases. He or she chooses "New Data", specifies a dataset name of "Lens", moves to the Interview status screen and specifies the domain, context and names for cases as shown in Figure 17. The expert leaves the default rating scale range of 1 through 9, allowing intermediate or "fuzzy" attributes to be entered.



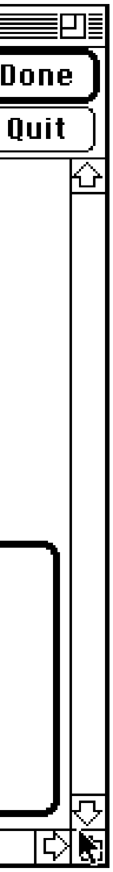


Figure 17 Status screen for contact lens knowledge elicitation

He or she then clicks on “Done”, moves to the *Interview* case selection screen, clicks “Done” again and moves to the *Interview* case entry screen, entering some initial cases as shown in Figure 18.

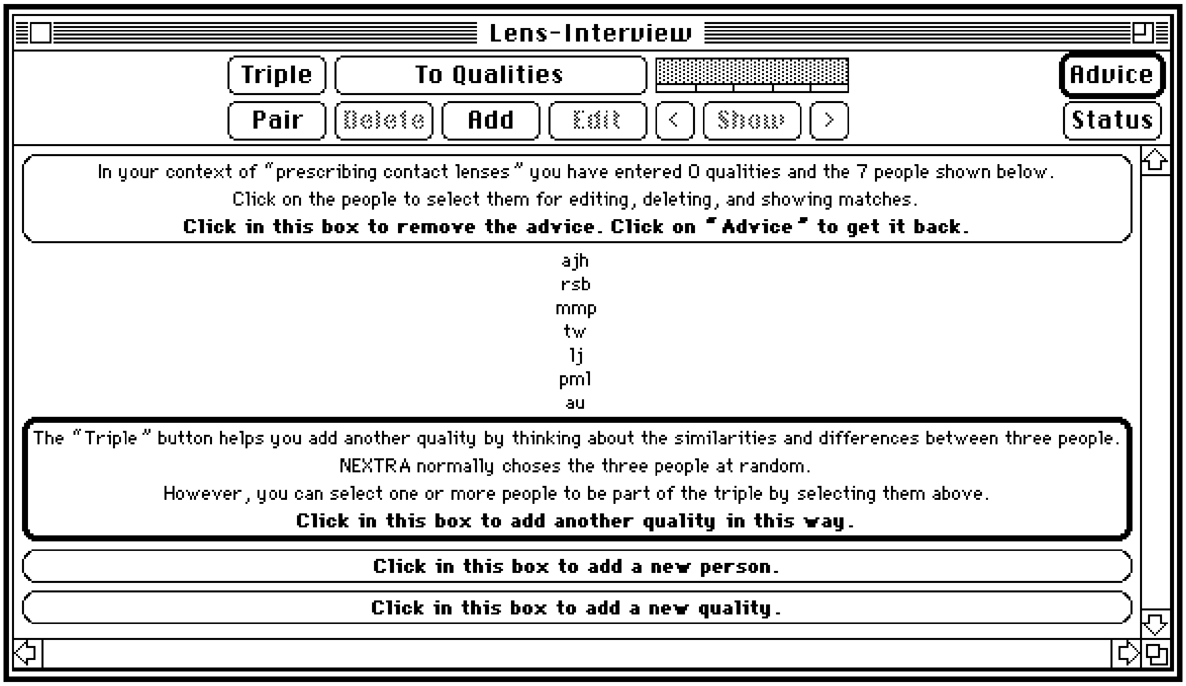
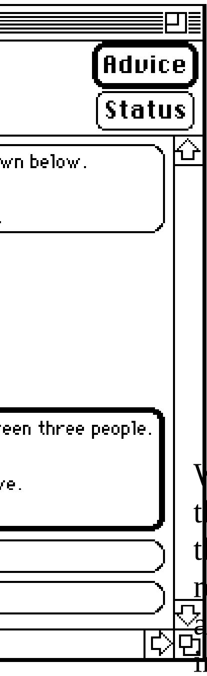
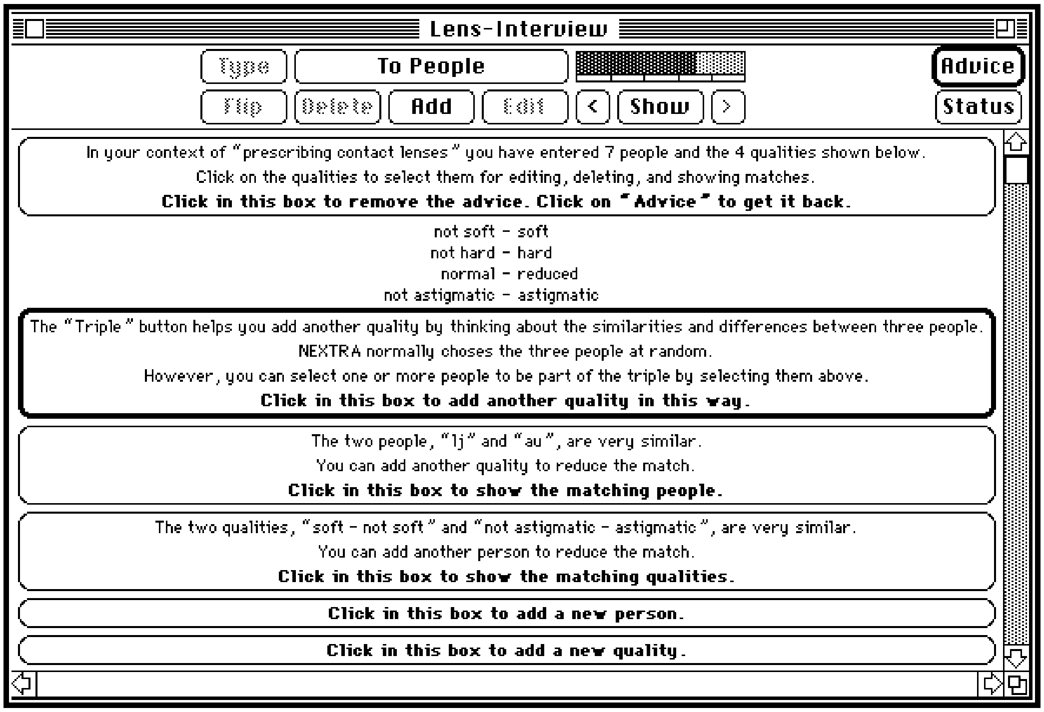


Figure 18 Initial clients entry



What initial cases does it make sense for the expert to choose? He or she should think in terms of those which represent critical features of the problem. For example, in the contact lens problem, the fact that clients with reduced tear production should not have lenses fitted should be represented, as should the fact that astigmatic clients in general have hard lens and non-astigmatic clients soft lens. The clients entered in Figure 18 correspond to 1, 2, 3, 4, 6, 8 and 10 in Figure 4.

The expert may next choose to add some initial attributes that are clearly relevant to the problem. Figure 19 shows the attributes "not soft—soft", "not hard—hard", "normal—reduced" (tear production), and "not astigmatic—astigmatic", added as initial attributes.



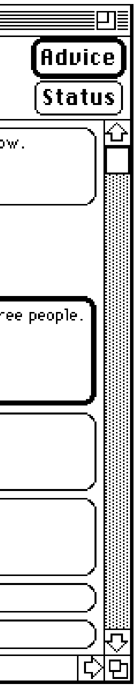


Figure 19 Initial qualities entry

Note that the expert has not assumed that the decisions to fit soft or hard contact lenses are mutually exclusive. Entering them as separate decisions allows for the possibility that either may be suitable for some types of patient. This example will show that this makes no difference to the behavior or performance of the knowledge acquisition tool or resulting expert system In fact, if the original dataset of Figure 1 is entered with two separate output attributes as is being done now, it will result in exactly the same behavior in the shell.

We know that the attributes entered are not sufficient to solve the problem. How does the acquisition tool help the expert to realize this? The most important capability is that of having interactive analysis at any time during an interview. The expert can see his or her dataset as a knowledge base and come to understand what is missing. Figure 20 shows the tool's hierarchical cluster analysis of the initial dataset with 6 cases and 3 attributes. It can be seen that "astigmatic" is clustered with "hard" and "not astigmatic" with "soft". It is apparent from the case clusters that groups of clients are not being distinguished. Note from the data in Figure 20 that the expert has chosen to use only the two extreme values of the rating scale, for example, in the top row the value 1 means "hard" is selected and the value 9 means "not hard" is selected. He or she could also have expressed doubt about some cases by using intermediate values.

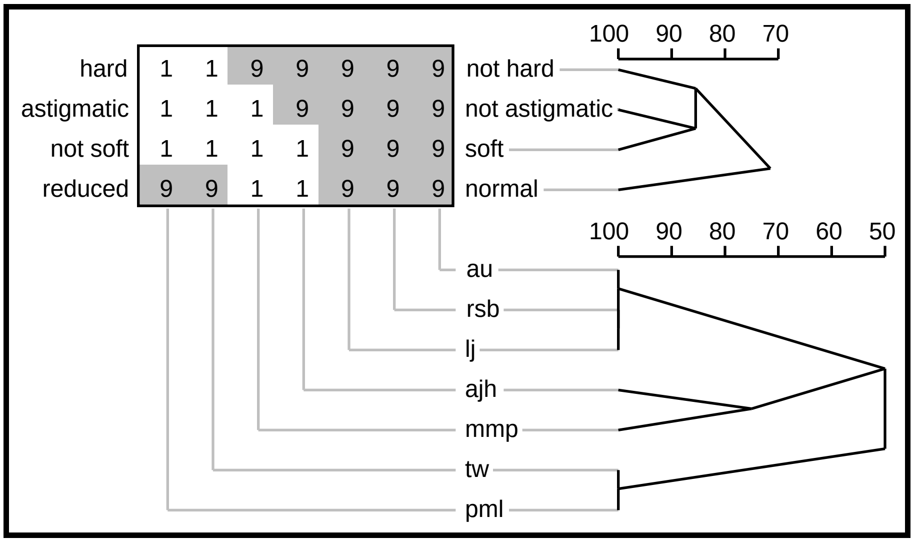


Figure 20 *FOCUS* analysis of initial data

Figure 21 shows the tool's alternative spatial analysis of the initial dataset. It is apparent that neither type of lens should be fitted if tear production is "reduced", and that "astigmatic" goes with "hard" and "not astigmatic" with "soft". Again the lack of distinction between groups of client is apparent. These two forms of analysis are closely related as will be apparent from checking that cases and attributes that are close together in one are also close together in the other. The hierarchical analysis has the advantage that the relation between the raw data and the clusters is apparent. The spatial analysis has the advantage that the relation between the cases and attributes is explicit.

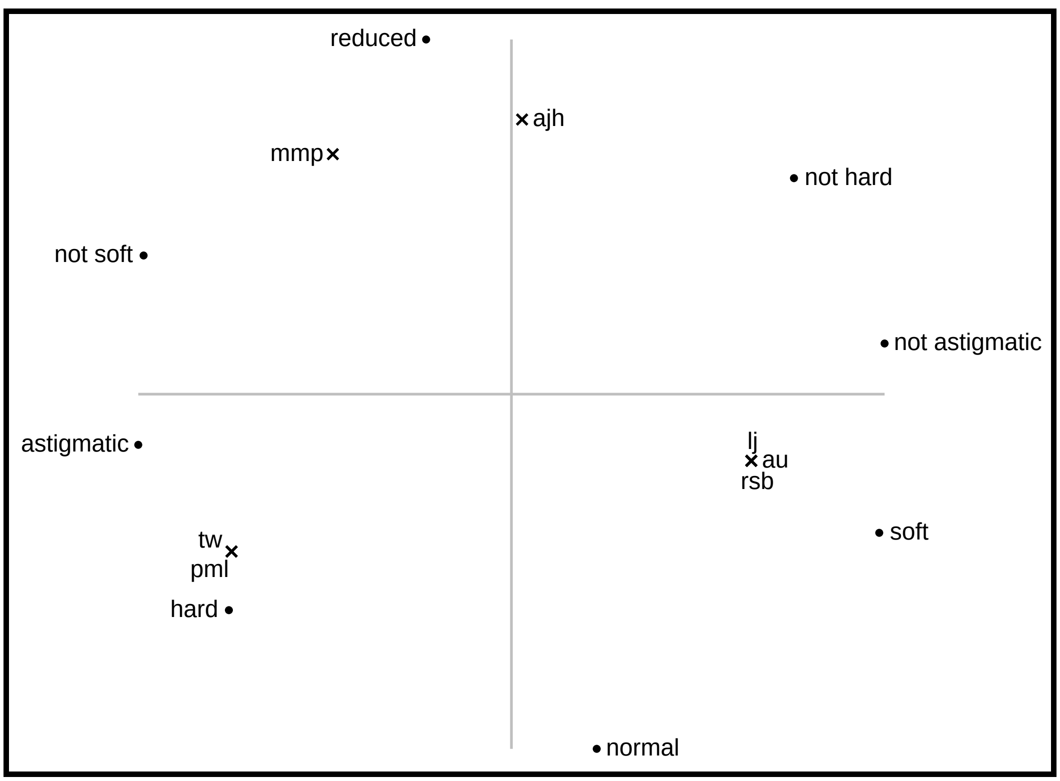


Figure 21 *PrinCom* analysis of initial data

To obtain a logical analysis of the entailments between attributes the expert or knowledge engineer has first to specify that the type of the first two attributes is “Output”, not “Input”, through the *Type* screen shown in Figure 22.

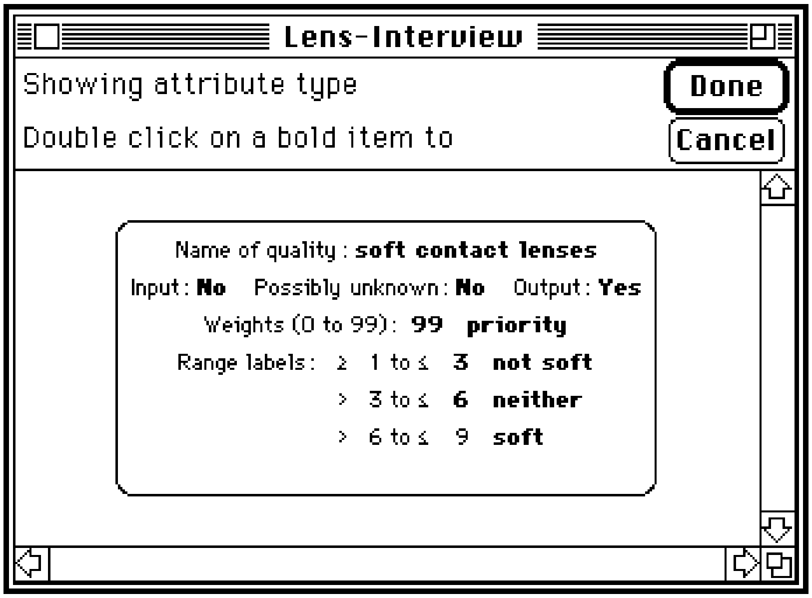


Figure 22 Setting the type of each quality

Running the *Induct* logical entailment analyis program then produces the rules shown in Figure 23. These conclude that soft lenses should be fitted when tear production is normal and the astigmatism is not astigmatic, and that hard lenses should be fitted when tear production is normal and the astigmatism is astigmatic. Note that this is already the simple default rule set for the contact lens situation already discussed. The exception rules are missing because the expert has not yet entered clients that should not have lenses fitted for other reasons.

astigmatism=astigmatic -> soft contact lenses=not soft 100% 18.7%
tear production=reduced -> soft contact lenses=not soft 100% 32.7%

tear production=normal & astigmatism=not astigmatic -> soft contact lenses=soft 100% 7.87%

astigmatism=not astigmatic -> hard contact lenses=not hard 100% 26.0%
tear production=reduced -> hard contact lenses=not hard 100% 51.0%

tear production=normal & astigmatism=astigmatic -> hard contact lenses=hard 100% 8.16%

Figure 23 *Induct* analysis of initial data

How does the expert proceed to add more attributes and cases? Several possibilities are open. He or she may continue the elicitation using triples or breaking case and attribute matches, or may think about the graphic representation of the knowledge base, or the rules produced so far. It does not really matter what prompts the expert to remember that age and prescription are also relevant variables and that there are some special cases that should be entered.

Note that it also does not matter if the expert enters a duplicate case. It will affect the result if the expert enters an incorrect decision. If this conflicts with an existing case it will be noticed quickly. Otherwise, it will be apparent through studying the graphic representations or rules.

Figure 24 shows a display of the dataset when the expert has entered another 7 cases (corresponding to clients 12, 13, 16, 18, 20, 22 and 24 in Figure 4) and the age and prescription attributes.

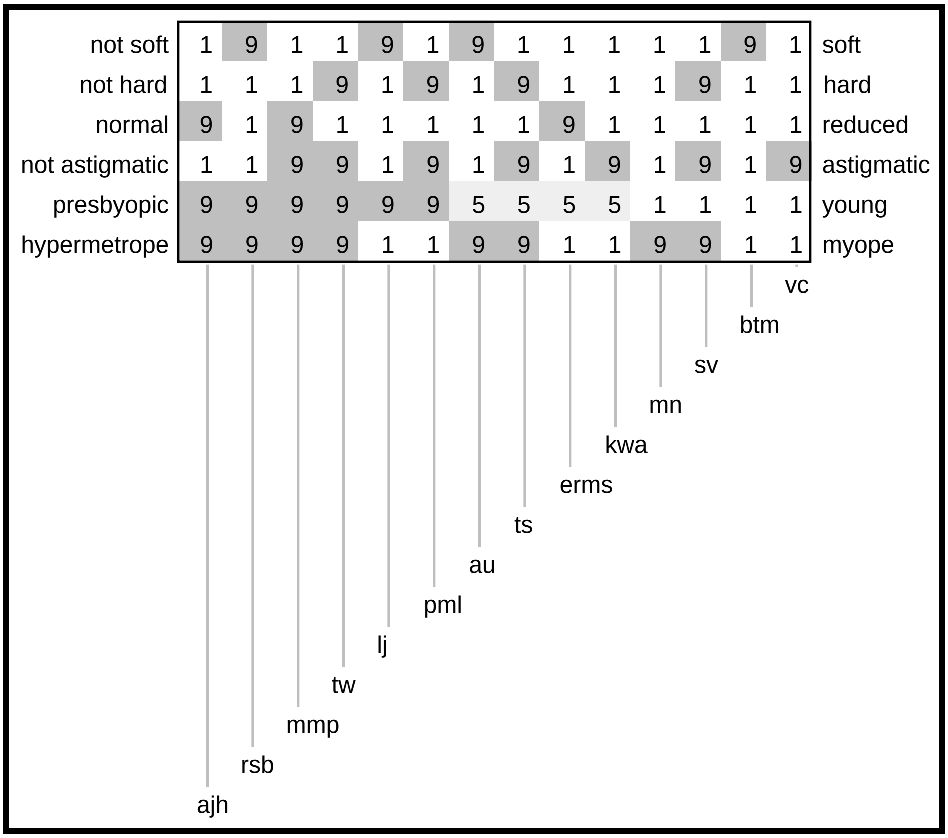


Figure 24 *Display* of final dataset

Figure 25 shows a spatial analysis of this dataset.

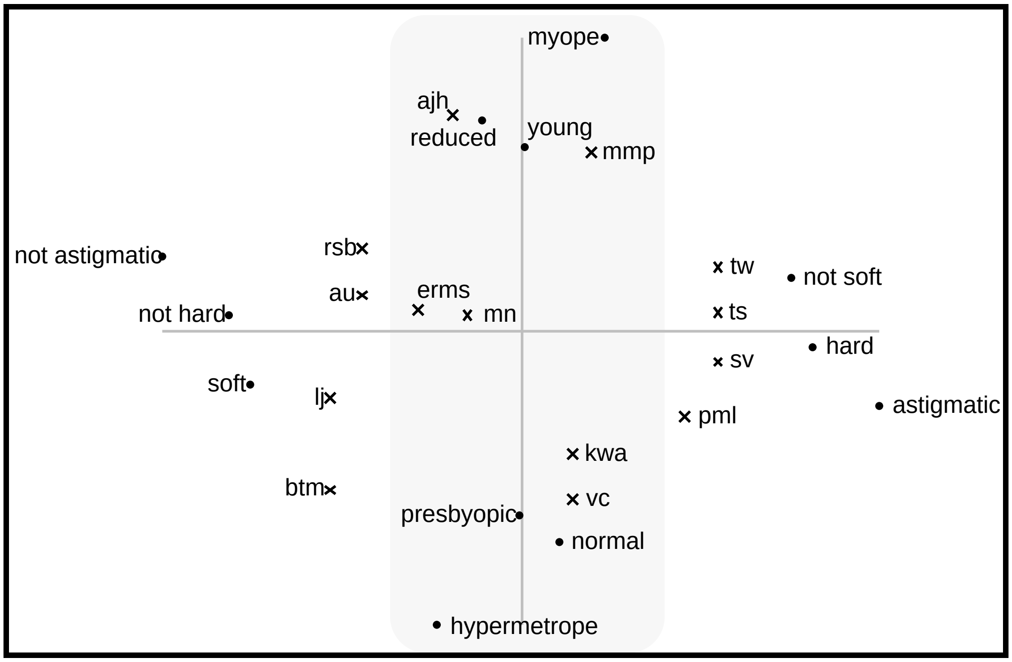


Figure 25 *PrinCom* analysis of final dataset

The area behind the clients for whom neither form of lens are recommended has been shaded so that the structure of the knowledge base stands out. Clients rsb, au, lj and btm are standard cases for soft lenses, tw, ts, sv and pml are standard cases for hard. Clients ajh, mmp and erms are standard cases for no lenses because their tear production is reduced. Client mn is the anomalous case where soft lenses might otherwise be fitted. Clients kwa and xc are the anomalous cases where hard lenses might otherwise be fitted.

Figure 26 shows the rules produced by an *Induct* analysis of the dataset.

```
tear production=reduced -> soft contact lenses=not soft 100% 36.4%
age=presbyopic & prescription=myope -> soft contact lenses=not soft 100% 51.0%

tear production=normal & astigmatism=not astigmatic & age=young -> soft contact
    lenses=soft 100% 8.16%
tear production=normal & astigmatism=not astigmatic & age=pre-presbyopic -> soft
    contact lenses=soft 100% 28.6%
astigmatism=not astigmatic & age=presbyopic & prescription=hypermetrope -> soft
    contact lenses=soft 100% 28.6%

astigmatism=not astigmatic -> hard contact lenses=not hard 100% 9.49%
tear production=reduced -> hard contact lenses=not hard 100% 36.4%
age=presbyopic & prescription=hypermetrope -> hard contact lenses=not hard 100%
    51.0%
age=pre-presbyopic & prescription=hypermetrope -> hard contact lenses=not hard
    100% 51.0%

tear production=normal & astigmatism=astigmatic & prescription=myope -> hard
    contact lenses=hard 100% 2.33%
astigmatism=astigmatic & age=young & prescription=hypermetrope -> hard contact
    lenses=hard 100% 28.6%
```

Figure 26 *Induct* analysis of final dataset

This rule set is logically identical to those of Figures 7 and 8. If the priority of the tear production attribute is set to 98 as before using the *Type* screen then exporting this dataset to the expert system shell produces the same correct behavior as before in consultations. Note that the lack of effect of the differences between the original example of Figure 4, and the elicited data of Figure 24—the use of a 1 to 9 rating scale, of two decision attributes and of a subset of cases. The knowledge elicitation is robust against such differences.

Suppose the expert does not realize that the knowledge base is adequate at this stage and adds further cases? Provided the correct lens prescription is given according to Cendrowska's original model, adding additional cases will have no effect on the rules produced by *Induct*. It may vary the cluster outputs since the additional cases will weight the analyses differently, but the broad structures apparent in these analyses will remain the same. If the expert adds further attributes in principle the *Induct* analysis will not be affected. However, it is possible for an arbitrary attribute, such as hair color, to correlate highly with a significant attribute or the decision attributes. On a small dataset of cases this may result in the arbitrary attribute appearing to be significant to *Induct*. Increasing the number of cases should get rid of spurious correlations. However, it is probably of much greater importance that the expert notices a spurious attribute appearing in the rules and decides to remove it. The acquisition tool is primarily designed to support the transfer of expertise by interviewing experts. It also has powerful techniques for statistically sound empirical induction. However, the number of cases required to be entered increases very rapidly with the proportion of incorrect decisions and irrelevant attributes, and it is better for the expert to clean up the data than to rely on statistics applied to poor quality data.

Figure 27 illustrates this by showing the trade-off between data and knowledge derived from empirical studies using the data of Figure 4 artificially corrupted with noise and irrelevant attributes (Gaines 1989a,b). Using default logic the knowledge in the data can be captured in 5 rules. It can also be entered as 14 critical cases as illustrated above. If a random collection of cases selected from those of Figure 1 is used as a dataset then on average 90 cases are required. If 1 irrelevant random attribute is added then on average 160 cases are required. If 25% of the decisions in the data are incorrect then on average 325 cases are required. If 5 irrelevant random attributes are added then on average 640 cases are required. If 1 irrelevant random attribute is added and 10% of the decisions in the data are incorrect then on average 1,970 cases are required. These results show the significance of using the expert's expertise effectively, compared with attempting to regenerate that expertise through gathering large amounts of data.

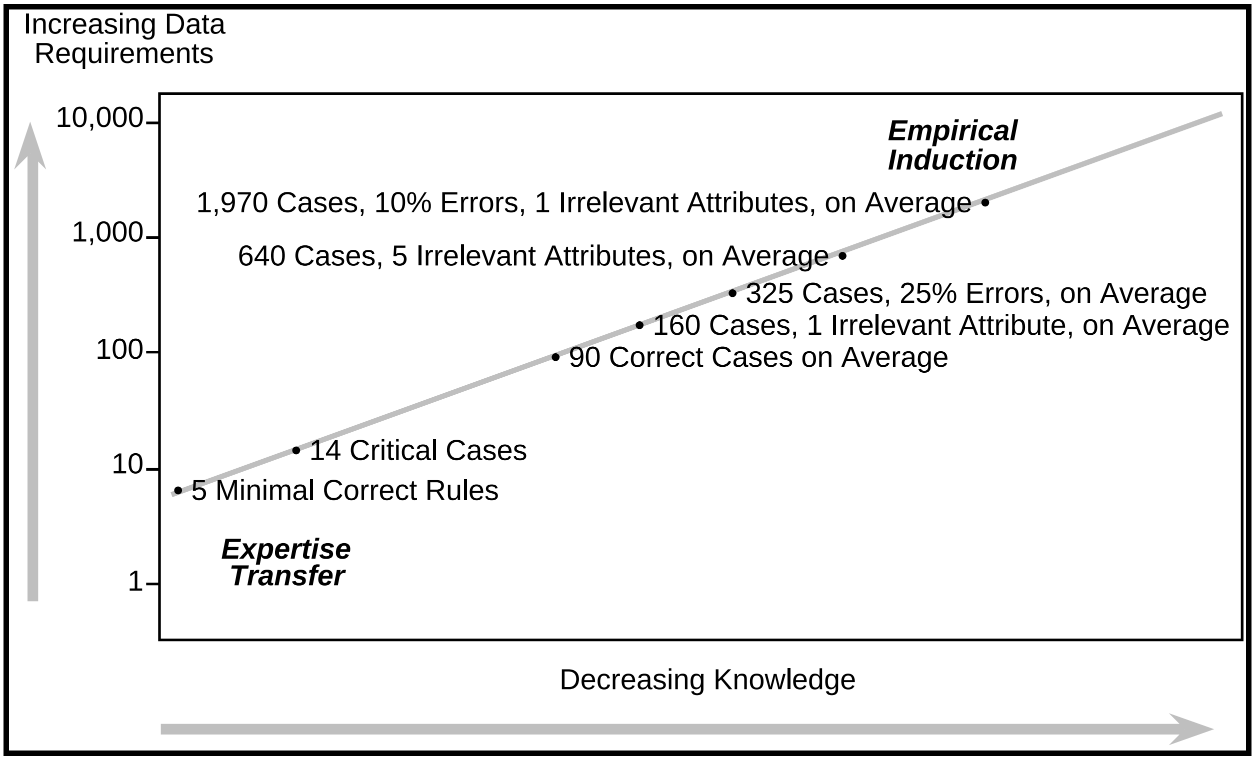


Figure 27 The trade-off between data and knowledge

8 Conclusions

The examples given in this paper demonstrate how a dataset representing an expert's conceptual structure for a problem domain can be elicited in a knowledge acquisition tool and converted automatically to a knowledge base that performs as expert system solving the problem in an expert system shell. In particular, examples have been given to show the significance of effective software engineering in transforming the elicited knowledge into data structures for the shell.

In large scale system development knowledge structures for different sub-domains have to be developed to support a complete system. Many of them will have the decision-making form analyzed in detail in this paper. The acquisition tool can be used to develop and validate knowledge structures for each of these sub-domains. The modular and well-structured knowledge bases produced can then be combined through appropriate control structures.

There is no royal road to expert system development. Understanding and communicating expertise are not easy tasks. Knowledge acquisition tools are designed to support and guide the expert and knowledge engineer in that understanding and communication. Any real system development involves exploration, false starts, inadequate datasets, and so on. The expert has to learn the skills of expertise transfer through trial and error. A good tool provides an interesting, friendly and supportive environment for that learning experience. Once that learning has converged upon a solution, the tool can also provide formally correct and well-structured mechanisms for transferring the solution to a knowledge base driving an expert system shell.

Acknowledgements

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada. We are grateful to many colleagues at the knowledge acquisition workshops for discussions which have highlighted many of the issues raised in this paper. We are particularly grateful to John Boose, Jeff Bradshaw and Brian Woodward for many fruitful discussions of knowledge acquisition systems, and to Alain Rappaport for such discussions and collaboration in the development of the knowledge base transfer facilities.

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. (1986) **Expertise Transfer for Expert System Design**. New York: Elsevier.

Boose, J. H. (1987) Rapid Acquisition and Combination of Knowledge from Multiple Experts in the Same Domain. **Future Computing Systems 1**(2) 191-216

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

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

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**(1), 3-28 (January).

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

Christaller, T., di Primio, F. & Voss, A. (1989). **Die KI-Werbank Babylon: Eine Offene und Portable Entwicklungsumgebung für Expertensysteme**. 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 (January).

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 (January).

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

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). 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. (1989a) 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. (1989b). Extracting knowledge from data. **Proceedings of the AAAI Workshop on Knowledge Discovery in Databases**. pp.109-116. Detroit (August).

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

Gaines, B.R. & Shaw, M.L.G. (1980). New directions in the analysis and interactive elicitation of personal construct systems. **International Journal of Man-Machine Studies 13**(1), 81-116 (July).

Gaines, B.R. & Shaw, M.L.G. (1986). Induction of inference rules for expert systems. **Fuzzy Sets and Systems**, 8(3), 315-328 (April).

Gaines, B.R. & Shaw, M.L.G. (1987). Knowledge support systems. **Proceedings of ACM MCC-University Research Symposium**. Austin, Texas: MCC. pp.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. San Mateo, California: Morgan Kaufmann (August).

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

Klinker, G., Bentolila, J., Genetet, S., Grimes, M. & McDermott, J. (1986) KNACK—report-driven knowledge acquisition. **International Journal of Man-Machine Studies 26**(1), 65-79 (January)

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

Musen, M.A., Fagan, L.M., Combs, D.M. & Shortliffe, E.H. (1987) Use of a domain model to drive an interactive knowledge editing tool. **International Journal of Man-Machine Studies 26**(1), 105-121 (January).

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

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

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 (October).

Rappaport, A. & Gaines, B.R. (1988). Integration of acquisition and performance systems. **Proceedings of the Third AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop**. pp.25-1-25-20. Banff (November).

Shaw, M.L.G. (1980). **On Becoming a Personal Scientist**. London: Academic Press.

Shaw, M.L.G., Ed. (1981) **Recent Advances in Personal Construct Technology**. London: Academic Press.

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. (1986). Interactive elicitation of knowledge from experts. **Future Computing Systems, 1**(2), 151-190.

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

Shaw, M.L.G. & Gaines, B.R. (1987b) 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 consensus, correspondence, conflict 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).

Slater, P., Ed. (1976). **Explorations of Intrapersonal Space: Volume 1**. London: John Wiley.

Slater, P., Ed. (1977). **Dimensions of Intrapersonal Space: Volume 2**. London: John Wiley.