Knowledge Acquisition (1989) **1**, 185–214

Recent progress in AQUINAS: a knowledge acquisition workbench

John H. Boose, David B. Shema and Jeffrey M. Bradshaw
Boeing Advanced Technology Center, Boeing Computer Services, P.O. Box 24346, Seattle, WA 98124, USA

(Based on a paper presented at the 3rd AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, November, 1988)

Acquiring knowledge from a human expert is a major problem when building a knowledge-based system. AQUINAS, an expanded version of the expertise transfer system (ETS), is a knowledge acquisition workbench that combines ideas from psychology and knowledge-based systems research to support knowledge acquisition tasks. AQUINAS interviews experts directly and helps them organize, analyse, test, and refine their knowledge bases. Expertise from multiple experts or other knowledge sources can be represented and used separately or combined, giving consensus and dissenting opinions among groups of experts. Results from user consultations are derived from information propagated through hierarchies. ETS and AQUINAS have assisted in building knowledge-based systems for several years at The Boeing Company.

1. Introduction

This paper describes recent progress on AQUINAS in the areas of knowledge base performance measurement, knowledge base maintenance, interacting trait constraints, consultation graphics, and eliciting strategic and procedural knowledge. Experiments show how AQUINAS can automatically improve knowledge bases and even suggest new problem-solving information. Forms of interactive and automatic machine learning employed by AQUINAS are also discussed.

2. Knowledge-based systems and rapid prototyping

Knowledge-based systems provide environments where preliminary knowledge bases from human experts may be rapidly prototyped in a matter of days or weeks. Experts, end-users, and project managers may easily examine the system's behavior and suggest changes that can be rapidly implemented. The knowledge, often stored in English-like rules, is readily available for examination, discussion, and modification by all project participants. The rapid prototyping capability of these environments is one reason why they have been successful. One of the first of these environments, EMYCIN (Buchanan & Shortliffe, 1985) containing the knowledge transfer tool TEIRESIAS (Davis & Lenat, 1982), is still one of the best examples of such a rapid prototyping environment. TEIRESIAS helped knowledge engineers and experts enter rules and parameters, checked consistency, debugged consultations, and even suggested new rules. The interface and rule language could be used by the expert with a relatively small amount of training.

1042-8143/89/020185 + 30$03.00/0 © 1989 Academic Press Limited

The major hurdle when building these systems is that of acquiring and modeling the human's expertise. A knowledge engineer—someone familiar with the knowledge-based system environment or "shell"—interviews the expert, models the knowledge, embeds it in the system, and reviews the system's behavior with the expert. The knowledge engineer changes the system based on the suggestions of the expert. For complex systems, cycles of revision and review can take from six to twenty-four months before the system exhibits expertise when using manual interviewing methods.

Automated knowledge acquisition tools are being developed to help cut down the revise-and-review cycle time and increase the reliability and maintainability of knowledge bases. These tools interview experts directly and help them refine, structure, and test their knowledge. Expertise from many experts may be rapidly combined together and used. Because models can be built rapidly and easily revised, AQUINAS can be effectively used as a general-purpose decision aid as well as a tool for knowledge engineering.

3. AQUINAS system description

This section discusses AQUINAS' knowledge representation and basis in personal construct theory. Knowledge analysis and refinement tools, and the system's rapid consultation prototyping capability will also be shown.

AQUINAS, an expanded version of the expertise transfer system (ETS; Boose, 1984, 1985, 1986*a*), is a knowledge acquisition workbench that combines ideas from psychology and knowledge-based systems research to support knowledge acquisition tasks. These tasks include eliciting distinctions, decomposing problems, combining uncertain information, incremental testing, integration of data types, automatic expansion and refinement of the knowledge base, use of multiple sources of knowledge, use of constraints during inference, and providing process guidance (Boose & Bradshaw, 1988). AQUINAS interviews experts and helps them analyse, test, and refine the knowledge base. Expertise from multiple experts or other knowledge sources can be represented and used separately or combined. Results from user consultations are derived from information propagated through hierarchies. AQUINAS delivers knowledge internally or by creating knowledge bases for several different expert system shells. Help is given to the expert by a dialog manager that embodies knowledge acquisition heuristics.

Using AQUINAS, rapid prototypes of knowledge-based systems can be built in as little as 1 hr, even when the expert has little understanding of knowledge-based systems or has no prior training in the use of the tool. The interviewing methods in AQUINAS are derived from George Kelly's Personal Construct Theory and related work (Kelly, 1955; Shaw & Gaines, 1987; Boose, 1988). Kelly's methods and theory provide a rich framework for modeling the qualitative and quantitative distinctions inherent in an expert's problem-solving knowledge.

AQUINAS tools mentioned in this paper are explained more fully elsewhere (Boose, 1988; Boose & Bradshaw, 1988; Bradshaw & Boose, 1989; Kitto & Boose, 1988*a*; Shema & Boose, 1988). AQUINAS is written in Interlisp and runs on the Xerox family of Lisp machines. Subsets of AQUINAS also run in an Interlisp

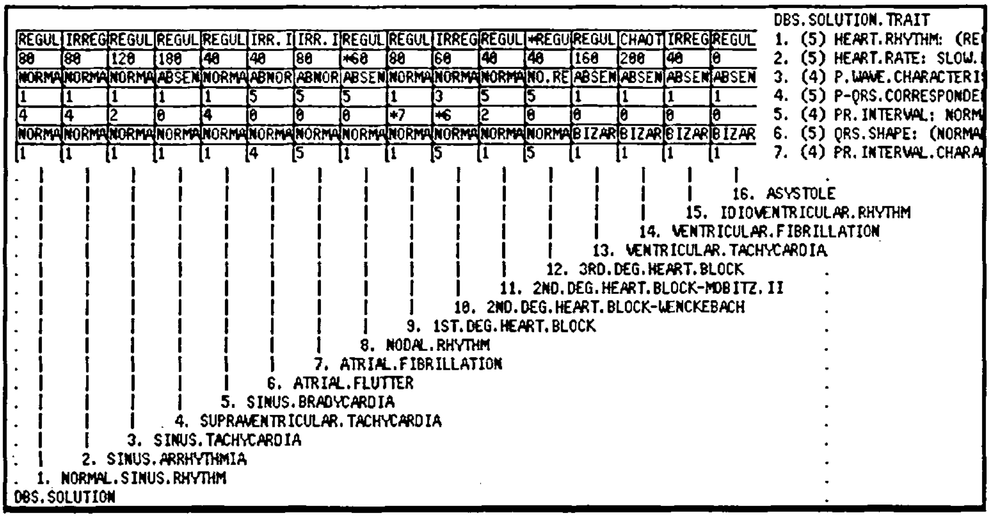


FIG. 1. Repertory grid for the heart dysrhythmias portion of the medical aid advisor.

version on the DEC Vax, SUN III and IV, and a micro computer "C/UNIX"-based portable version that runs on a variety of platforms.

3.1. KNOWLEDGE REPRESENTED IN REPERTORY GRIDS

The expert enters and refines knowledge in the form of repertory grids. In a repertory grid, problem solutions—*elements*—are elicited and placed across the grid as column labels, and traits of these solutions—*constructs*—are listed alongside the rows of the grid (Fig. 1). Traits are first elicited by presenting groups of solutions and asking the expert to discriminate among them. Following this, the expert gives each solution a rating showing where it falls on the trait scale, and gives the relative importance of each trait. Below are excerpts from an expert building a repertory grid for a heart dysrhythmias portion of a medical aid advisor.

First, the expert partitions the medical aid advisor into several subproblems, or *cases*. Then, for the HEART.DYSRHYTHMIAS case, the expert enters a list of possible problems (termed *solutions* in AQUINAS; the goal of this part of the system is to diagnose the problem):

:

Please enter a list of items for expert DBS that are potential solutions for HEART.DYSRHYTHMIAS. Enter them one to a line. When you're done, enter a RETURN
NEW SOLUTION**<u>NORMAL SINUS RHYTHM</u>
NEW SOLUTION**<u>SINUS ARRHYTHMIA</u>
NEW SOLUTION**<u>SINUS TACHYCARDIA</u>
NEW SOLUTION**<u>SUPRAVENTRICULAR TACHYCARDIA</u>
NEW SOLUTION**<u>SINUS BRADYCARDIA</u>
NEW SOLUTION**<u>ATRIAL FLUTTER</u>

NEW SOLUTION**<u>**ATRIAL FIBRILLATION**</u>**
NEW SOLUTION**<u>**NODAL RHYTHM**</u>**
NEW SOLUTION**<u>**1ST DEG HEART BLOCK**</u>**
NEW SOLUTION**<u>**2ND DEG HEART BLOCK-WENCKEBACH**</u>**
NEW SOLUTION**<u>**2ND DEG HEART BLOCK-MOBITZ II**</u>**
NEW SOLUTION**<u>**3RD DEG HEART BLOCK**</u>**
NEW SOLUTION**<u>**VENTRICULAR TACHYCARDIA**</u>**
NEW SOLUTION**<u>**VENTRICULAR FIBRILLATION**</u>**
NEW SOLUTION**<u>**IDIOVENTRICULAR RHYTHM**</u>**
NEW SOLUTION**<u>**ASYSTOLE**</u>**

AQUINAS asks the expert for traits that help distinguish between sets of solutions.

Think of an important new trait that two of NORMAL.SINUS.RHYTHM, SINUS.ARRHYTHMIA, and SINUS.TACHYCARDIA share, but that the other one does not. What is that trait?
NEW TRAIT (EXTREME)**<u>**REGULAR RHYTHM**</u>**

What is that trait's opposite as it applies in this case?
NEW TRAIT (OPPOSITE)**<u>**IRREGULAR RHYTHM**</u>**

What is the name of a scale or concept that describes REGULAR.RHYTHM/IRREGULAR.RHYTHM?
NEW TRAIT (CONCEPT)**<u>**HEART RHYTHM**</u>**

Think of an important new characteristic that two of SINUS.ARRHYTHMIA, SINUS.TACHYCARDIA, and SUPRAVENTRICULAR.TACHYCARDIA shae, but that the other one does not. What is that characteristic?
NEW TRAIT (EXTREME)**<u>**SLOW HEART RATE**</u>**

What is that characteristic's opposite as it applies in this case?
NEW TRAIT (OPPOSITE)**<u>**FAST HEART RATE**</u>**

What is the name of a scale or concept that describes SLOW.HEART.RATE/FAST.HEART.RATE?
NEW TRAIT (CONCEPT)**<u>**HEART RATE**</u>**

Think of an important new attribute that two of SINUS.TACHYCARDIA, SUPRAVENTRICULAR.TACHYCARDIA, and SINUS.BRADYCARDIA share, but that the other one does not. What is that attribute?
NEW TRAIT (EXTREME)**<u>**NORMAL P WAVE**</u>**

What is that attribute's opposite as it applies in this case?
NEW TRAIT (OPPOSITE)**<u>**ABNORMAL P WAVE**</u>**

What is the name of a scale or concept that describes NORMAL.P.WAVE/ABNORMAL.P.WAVE?
NEW TRAIT (CONCEPT)**<u>**P WAVE CHARACTERISTIC**</u>**

:

The expert then rates each solution on each trait. By default, AQUINAS scales each trait with ordinal rating values from 1 to 5. The expert can change the rating value type for a particular trait to nominal, interval, or ratio, and change rating value ranges and intervals. AQUINAS can assist with this in certain situations, and even suggest the kinds of changes that should be made.

```
Rate these solutions on the concept HEART.RHYTHM using a scale of 1 to 5,
where 1 means more like REGULAR.RHYTHM and 5 means more like
IRREGULAR.RHYTHM. If neither one seems to apply, enter NONE. If both seem to
apply, enter BOTH. If you would like help estimating values, enter
USE.BARCHART. If you wish to change the type of the trait, enter CHANGE.
REGULAR.RHYTHM(1) IRREGULAR.RHYTHM(5)
NORMAL.SINUS.RHYTHM** 1
SINUS.ARRHYTHMIA** 1
SINUS.TACHYCARDIA** 1
SUPRAVENTRICULAR.TACHYCARDIA** 1
SINUS.BRADYCARDIA** 1
ATRIAL.FLUTTER** 2
ATRIAL.FIBRILLATION** 5
NODAL.RHYTHM** 1
1ST.DEG.HEART.BLOCK** 1
2ND.DEG.HEART.BLOCK-WENCKEBACH** 4
2ND.DEG.HEART.BLOCK-MOBITZ.II** 1
3RD.DEG.HEART.BLOCK** 1 CF.6 5 CF .4
VENTRICULAR.TACHYCARDIA** 1
VENTRICULAR.FIBRILLATION** 5
IDIOVENTRICULAR.RHYTHM** 5
ASYSTOLE** 1

  :
```

Note that the expert entered a *rating distribution* for 3RD.DEGREE.HEART.BLOCK. More complex distributions can also be entered graphically. The expert then rates the importance of each trait's contribution to finding a solution.

```
Please rate the relative importance of HEART.RHYTHM
(REGULAR.RHYTHM/IRREGULAR.RHYTHM) on a scale from 5 to 0, where 5
means more important, and 0 means less important.
HEART.RHYTHM** 5

Please rate the relative importance of HEART.RATE (SLOW
HEART.RATE/FAST.HEART RATE) on a scale from 5 to 0, where 5 means more
important, and 0 means less important.
HEART.RATE** 5

Please rate the relative importance of P.WAVE.CHARACTERISTIC
(NORMAL.P.WAVE/ABNORMAL.P.WAVE) on a scale from 5 to 0, where 5 means
more important, and 0 means less important.
P.WAVE.CHARACTERISTIC** 4
```

3.2. REPERTORY GRID ANALYSIS

After an initial grid is constructed, AQUINAS helps the expert refine and expand the knowledge base by invoking a variety of analysis tools. *Similarities* between solutions and traits are analysed to help the expert refine useful distinctions and eliminate those that are inconsequential or redundant. For example, NORMAL.SINUS.RHYTHM and 1ST.DEG.HEART.BLOCK are found to be highly matched, and the expert enters a new trait, PR.INTERVAL, to help further distinguish between them. After the next row is added, AQUINAS asks the expert to rate all the solutions on the new trait.

> The two solutions **NORMAL.SINUS.RHYTHM** and **1ST.DEG.HEART.BLOCK** are matched at the **95%** level. Can you think of some new trait that would distinguish between them?
> AQU** YES
>
> What is the name of that trait?
> NEW TRAIT (CONCEPT)** PR.INTERVAL
>
> :

Inductive implications between trait values can show the expert higher levels of abstraction suggested by patterns of ratings in a repertory grid (Fig. 2). If the expert disagrees with an implication, AQUINAS helps guide the expert in making changes that will weaken inappropriate implications. This is one way in which interactive learning takes place in AQUINAS.

Rating scale types for a particular trait may be changed from the original default (ordinal scale with ranges from one to five) to other types (nominal, interval, ratio scales with user-defined ranges) based on the degree of precision required, cost of eliciting specific knowledge, and representation convenience. AQUINAS helps the expert change rating scale types in several ways. Here, AQUINAS notices that one trait has only extreme rating values (all ratings are either 1 to 5), and suggests that the rating scale type be changed from ordinal to nominal. The expert agrees, and AQUINAS automatically maps the numerical values to symbolic nominal values.

> P.WAVE.CHARACTERISTIC has only extreme values. Should its trait type be changed from ORDINAL to NOMINAL?
> AQU** YES

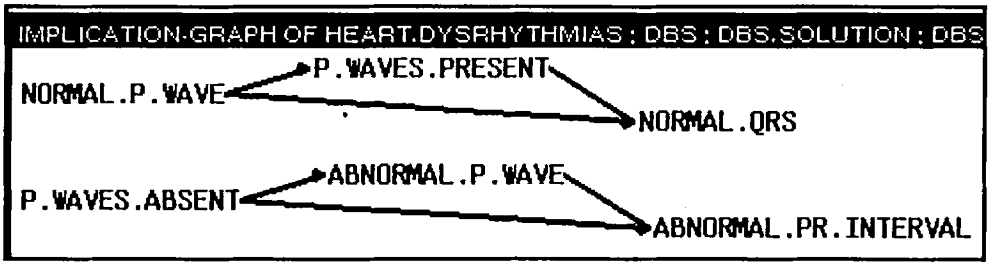


FIG. 2. Implication analysis performed on initial repertory grid.

Use the default method for mapping old to new values of P.WAVE.CHARACTERISTIC?
AQU** <u>YES</u>

Mapping for NORMAL.SINUS.RHYTHM from 1 to: NORMAL.P.WAVE 1.00 DEFAULT MAPPING
Mapping for SINUS.ARRHYTHMIA from 1 to: NORMAL.P.WAVE 1.00 DEFAULT MAPPING

:

Nominal, ordinal, and interval trait scales were illustrated in the grid in Fig. 1.

3.3. HIERARCHICAL REPERTORY GRIDS

In an important extension to Kelly's methods, AQUINAS allows experts to arrange repertory grid knowledge in hierarchies. Hierarchical tools in AQUINAS help the expert build, edit, and analyse knowledge in hierarchies. These hierarchies allow the

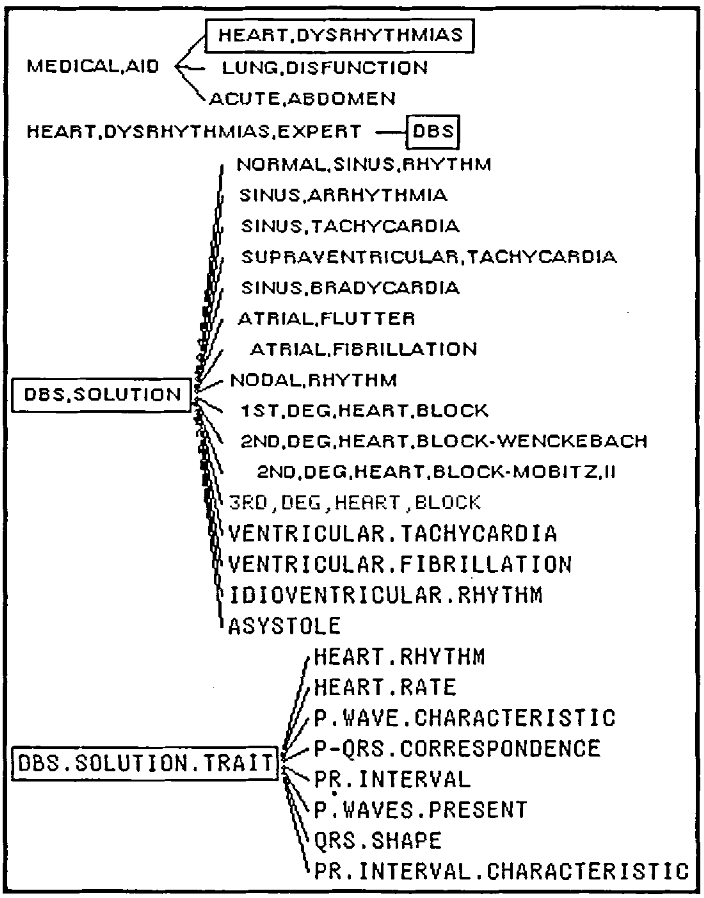


FIG. 3. Portions of four hierarchies for the medical aid advisor. Three cases are shown, as well as expert DBS's solution and trait hierarchies. So far, DBS is the only one who has entered information about heart dysrhythmias.

expert to break up complex problems into pieces of convenient size and similar levels of abstraction. Hierarchies in AQUINAS are organized around *solutions, traits, knowledge sources* (i.e. experts, databases, sensors), and *cases* (Figs 3 and 4).

Nodes in the four hierarchies combine to form repertory grids. In the most simple case, the children of a node in a solution hierarchy supply the solutions along the top of a grid; the children of a node in a trait hierarchy supply the traits down the side of a grid (Fig. 4). Rating values within the grid are judgments about the solutions with respect to each trait (Fig. 5).

Strategies for helping the expert build and refine hierarchies in AQUINAS include *laddering* (eliciting specializations and generalizations through dialogs that pursue the implications of "how" and "why" questions), *cluster analysis* (experts are

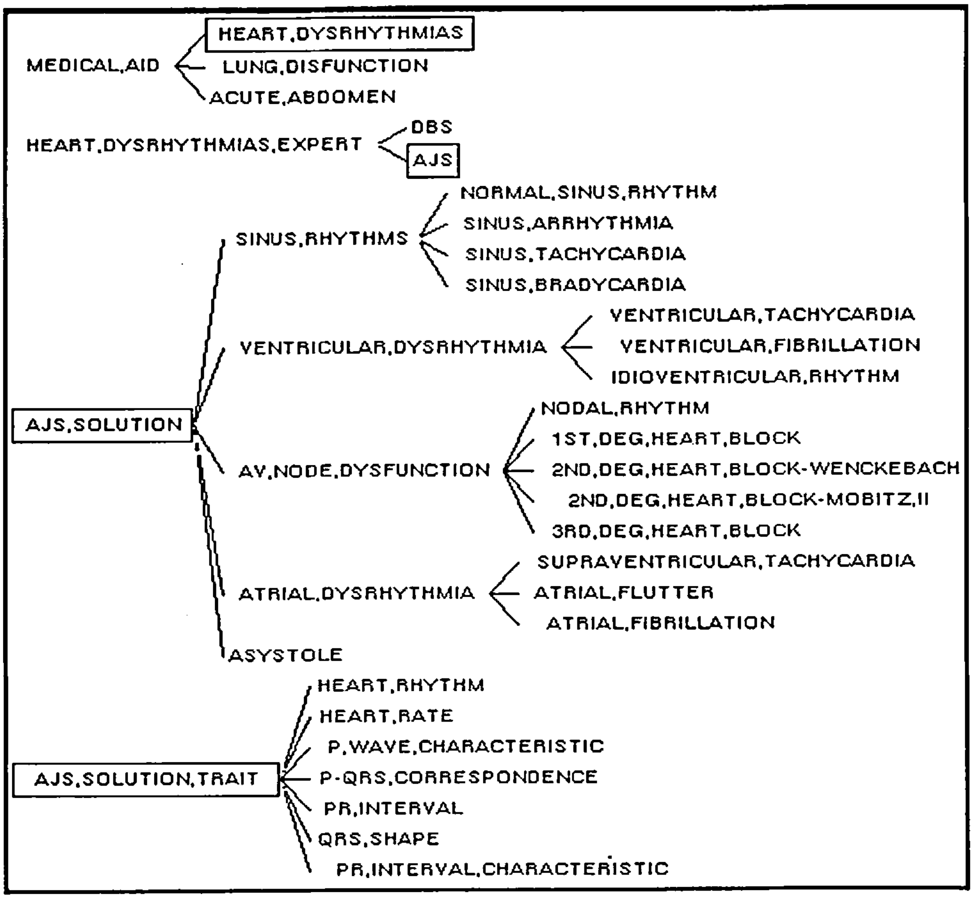


FIG. 4. Another expert, AJS, has entered knowledge for the same case. Tools in AQUINAS were used to help AJS structure solutions in a class hierarchy. The hierarchies are related from top to bottom: given the HEART.DYSRHYTHMIAS case, experts DBS and AJS exist (other experts could exist for other cases). Given expert AJS, we see his solution hierarchy (the top of the hierarchy is the class AJS.SOLUTION). Given that we are looking at the top-level grid in the hierarchy (the children of AJS.SOLUTION), the immediate children of AJS.SOLUTION.TRAIT apply. Different traits can apply to different class levels in the solution hierarchy; different solutions could come from different experts, and so on.

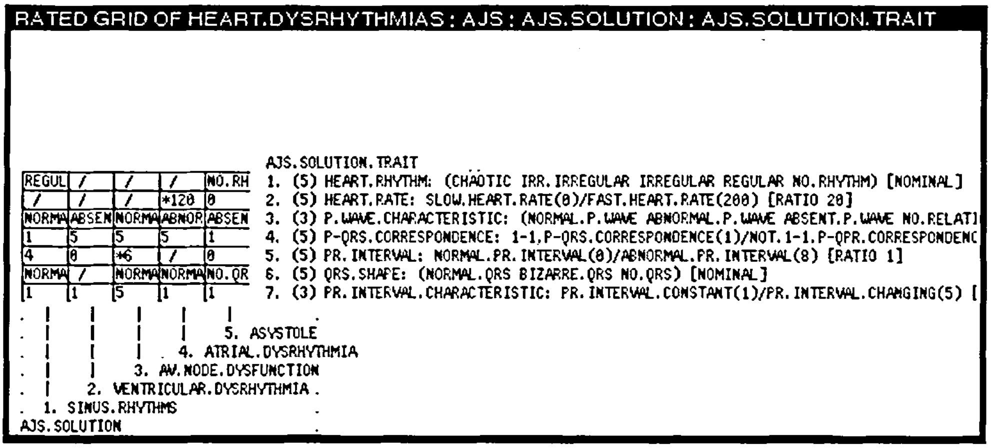


FIG. 5. AJS's top-level grid shows *solution classes* and traits. Separate grids exist for each class. During consultations AQUINAS will search the hierarchies to ask questions and infer ratings.

asked to label levels of abstraction represented by junctions in a cluster tree generated statistically from ratings in a grid; see Fig. 6), *implication pattern analysis* (certain patterns of implication may suggest levels of hierarchical subsumption), and *trait value examination* (certain combinations of trait values may suggest that hierarchical expansion is appropriate).

3.4. DECISIONS AND PERFORMANCE: CONSULTATIONS IN AQUINAS

Knowledge embedded in hierarchical repertory grids can be applied to specific problems by running consultations. During a consultation, AQUINAS asks the user to specify observations, preferences, or constraints associated with particular traits and solutions. Consultation results are displayed as a list of solutions, rank-ordered by strength of recommendation.

The model of problem-solving used in AQUINAS is that of multiple knowledge sources (experts) that work together in a common problem solving context (case) by selecting the best alternatives for each of a sequential set of decisions (solutions). Alternatives at each step are selected by combining relevant information about preferences (relativistic reasoning), constraints (absolute reasoning) and evidence (probabilistic reasoning). This paradigm follows one suggested by Clancey who suggested that many problems are solved by abstracting data, heuristically mapping higher level problem descriptions onto solution models, and then refining these models until specific solutions are found (Clancey, 1986). A variant of a maximum entropy approach is applied by AQUINAS's reasoning mechanism (Boose & Bradshaw, 1988; Bradshaw & Boose, 1989). ETS used a reasoning approach based on rules and certainty factors (Boose, 1986*a*).

Knowledge from multiple experts may be rapidly combined using AQUINAS. Users may receive dissenting as well as consensus opinions from groups of experts, thus getting a full range of possible solutions. Disagreement between the consensus and the dissenting opinion can be measured to derive a degree of conflict for a

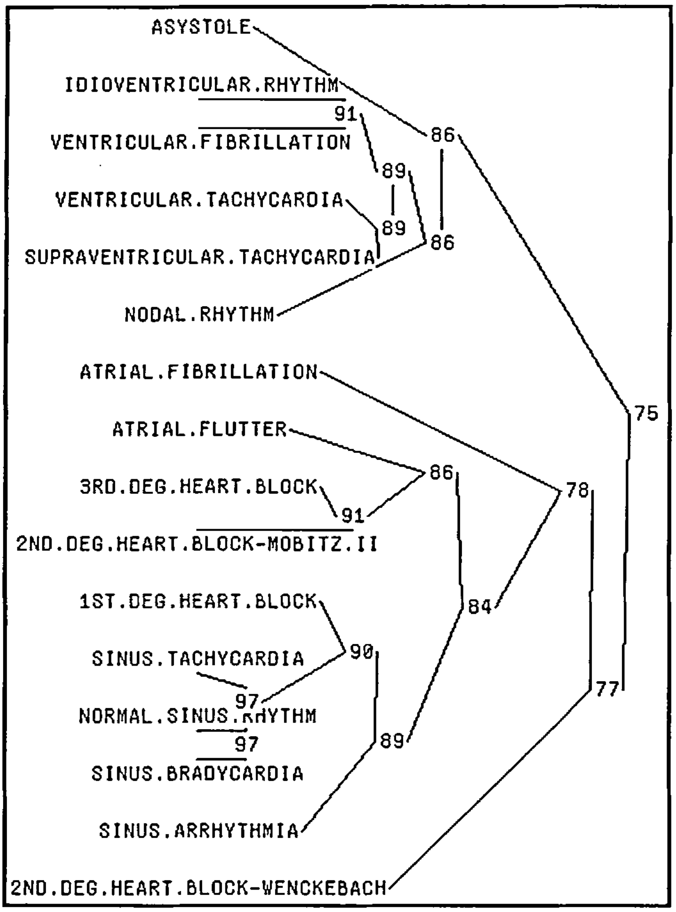


FIG. 6. A solution cluster analysis is produced from AJS's initial repertory grid. AJS is asked to form hierarchies by labeling junctions on the cluster. The clusters SINUS.RHYTHMS, VENTRICULAR.DYSRHYTHMIA, AV.NODE.DYSFUNCTION, and ATRIAL.DYSRHYTHMIA, shown in Figure 4, are formed.

particular consultation. The system can be used for cost-effective group data gathering (Boose, 1986*b*, 1988; Schuler, Russo, Boose & Bradshaw, 1989).

Following are portions of a transcript from an AQUINAS consultation using expertise from both DBS and AJS. The user specifies trait preferences, and AQUINAS combines the preferences together producing consensus and dissenting opinions:

:

Enter a brief description of this consultation
AQU** **56 YO MALE FOUND DOWN, UNKNOWN DOWNTIME, NO CPR @ ARRIVAL**

What solutions would you like to consider for consultation MEDICAL-AID-ALLEN
Enter them one to a line. If you wish every solution to be considered, enter ALL.

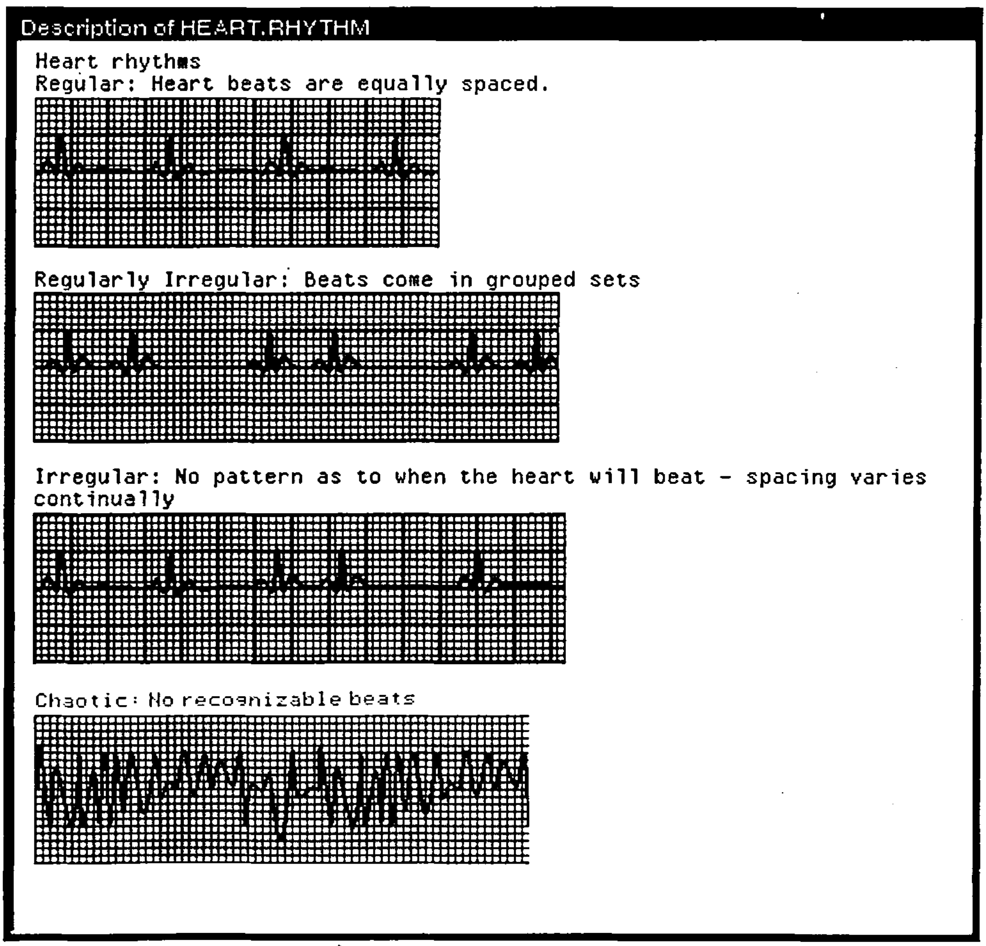
You may specify solution weights by appending selections with a number between 0 and 1 (e.g., 25). When done, enter RETURN.
AQU** ALL

The following experts know about the solutions; DBS AJS
Would you like to weight these experts?
AQU** NO

What kind of ordering should be used for questions (COST-BENEFIT SEQUENTIAL USER-SPECIFIED)?
AQU** SEQUENTIAL

:

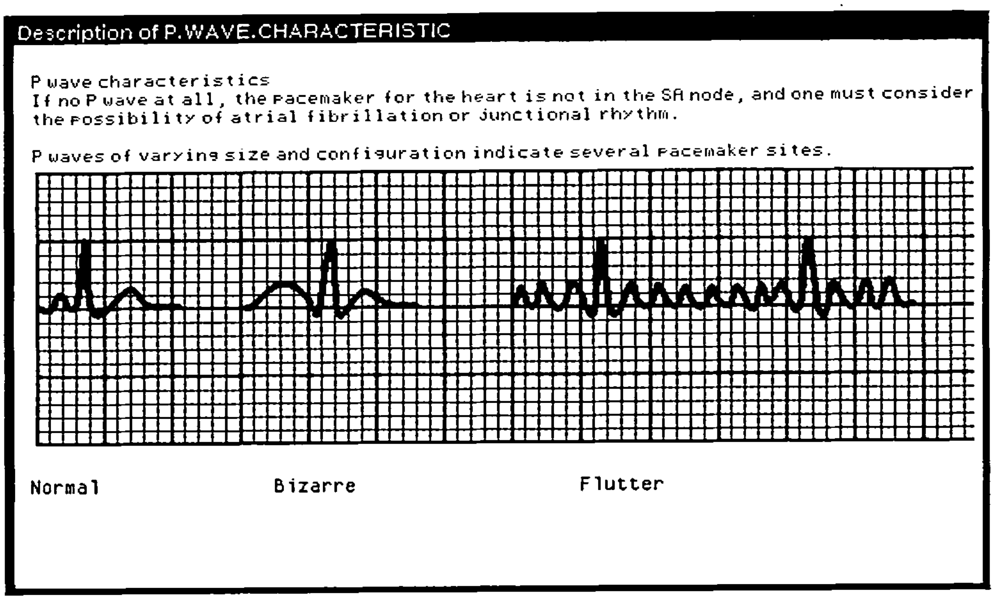
AQUINAS can display graphic information when asking questions. Here, the user answers a consultation question by comparing a scope reading with categories of heart rhythm.



Please indicate your preference, observation, or expectation for the attribute HEART.RHYTHM(AJS). You may also USE.BARCHART for entry.
(CHAOTIC IRR.IRREGULAR IRREGULAR REGULAR NO.RHYTHM) WT 5
AQU** <u>REGULAR</u>

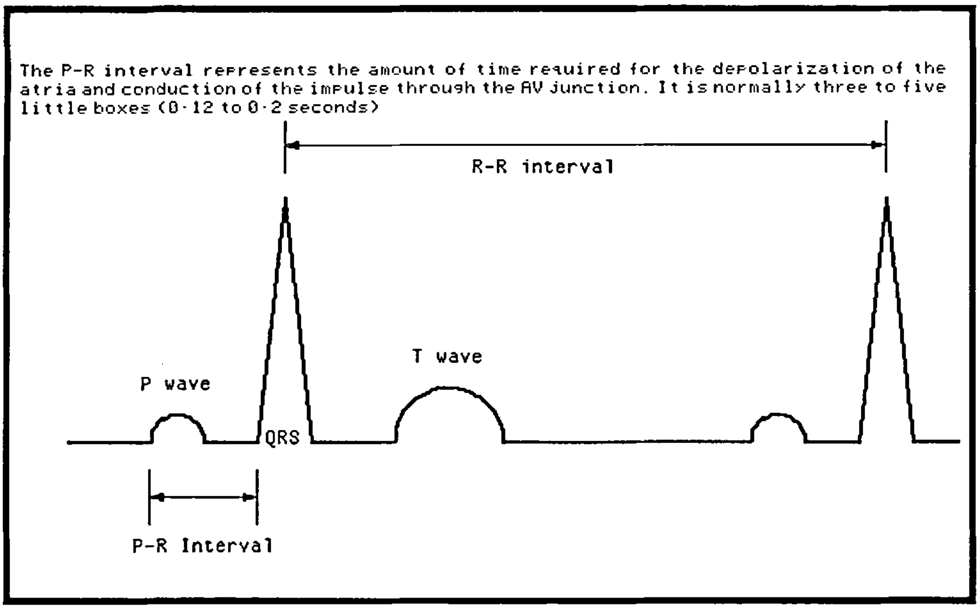
Please indicate your preference, observation, or expectation for the attribute HEART.RATE (AJS). The interval is 20. You may also USE.BARCHART for entry.
SLOW.HEART.RATE(0)/FAST.HEART.RATE(200) WT 5
AQU** <u>60</u>

Again, the user enters information by comparing a scope reading with graphic information displayed by AQUINAS. The expert or knowledge engineer enters this information by preparing a graphic file that is associated with a trait. Graphics can be created with a variety of paint programs and bitmap editors, or scanned and digitized. Six graphic displays are associated with this knowledge base; they took approximately two hours to enter.



Please indicate your preference observation, or expectation for the attribute P.WAVE.CHARACTERISTIC (AJS). You may also USE.BARCHART for entry.
(NORMAL.P.WAVE ABNORMAL.P.WAVE ABSENT.P.WAVE NO. RELATION) WT 3
AQU** <u>NORMAL.P.WAVE</u>

Please indicate your preference, observation, or expectation for the attribute P-QRS.CORRESPONDENCE (AJS). You may also USE.BARCHART for entry.
1-1.P-QRS.CORRESPONDENCE(1)/NOT.1-1.P-QPR.CORRESPONDENCE(5) WT 5
AQU** <u>1</u>



Please indicate your preference, observation, or expectation for the attribute PR.INTERVAL (AJS). The interval is 1 You may also USE.BARCHART for entry.
NORMAL.PR.INTERVAL(0)/ABNORMAL.PR.INTERVAL(8) WT 5
AQU** 4

:

After the observations are entered, AQUINAS displays the results of the consultation. Weighted contributions from each expert are shown. When only one expert contributes to a solution, the weight is shown at 1·00 (for example, SINUS.RHYTHMS). When more than one expert contributes to a solution each expert's contribution weight is shown. In SINUS.BRADYCARDIA for example DBS and AJS each contribute equally (0·5) so a weighted average of their contributions is computed as the total weight for the solution. If the user had weighted the experts differently, this would be taken into account when computing the average.

Test results for consultation
MEDICAL-AID-ALLEN

1 : SINUS.RHYTHMS (0·98)	(AJS 1·00 0·98)
2 : NORMAL.SINUS.RHYTHM (0.91)	(DBS 0·50 0·83)(AJS 0·50 0·99)
3 : SINUS.BRADYCARDIA (0.90)	(DBS 0·50 0·83)(AJS 0·50 0·98)
4 : SINUS.TACHYCARDIA (0.87)	(DBS 0·50 0·77)(AJS 0·50 0·96)
5 : SINUS.ARRHYTHMIA (0.83)	(DBS 0·50 0·83)(AJS 0·50 0·83)
6 : 1ST.DEG.HEART.BLOCK (0.73)	(DBS 0·50 0·78)(AJS 0·50 0·68)
7 : 2ND.DEG.HEART.BLOCK-MOBITZ.II (0.64)	(DBS 0·50 0·61)(AJS 0·50 0·68)
8 : AV.NODE.DYSFUNCTION (0.62)	(AJS 1·00 0·62)
9 : SUPRAVENTRICULAR.TACHYCARDIA (0.59)	(DBS 0·50 0·56)(AJS 0·50 0·62)

10 : 2ND.DEG.HEART.BLOCK-WENCKEBACH (0.57)	(DBS 0·50 0·61)(AJS 0·50 0·52)
11 : NODAL.RHYTHM (0.56)	(DBS 0·50 0·48)(AJS 0·50 0·64)
12 : ATRIAL.DYSRHYTHMIA (0.56)	(AJS 1·00 0·56)
13 : ATRIAL.FLUTTER (0.50)	(DBS 0·50 0·39)(AJS 0·50 0·60)
14 : 3RD.DEG.HEART.BLOCK (0.48)	(DBS 0·50 0·36)(AJS 0·50 0·59)
15 : VENTRICULAR.FIBRILLATION (0.46)	(DBS 0·50 0·55)(AJS 0·50 0·38)
16 : ASYSTOLE (0.45)	(DBS 0·50 0·45)(AJS 0·50 0·45)
17 : IDIOVENTRICULAR.RHYTHM (0·43)	(DBS 0·50 0·48)(AJS 0·50 0·38)
18 : ATRIAL.FIBRILLATION (0·40)	(DBS 0·50 0·36)(AJS 0·50 0·44)
19 : VENTRICULAR.TACHYCARDIA (0·40)	(DBS 0·50 0·42)(AJS 0·50 0·38)
20 : VENTRICULAR.DYSRHYTHMIA (0·38)	(AJS 1·00 0·38)

The simple weighted average is a comprehensible measure that avoids complex issues surrounding other consensus-type measures. We will experiment with these other techniques in future versions of AQUINAS.

Next, a dissenting opinion is presented. Correlation scores for all experts are listed. A score of 1·0 would represent perfect ordering agreement (lists in the same order); a score of −1·0 would represent perfect disagreement (lists in opposite order). A score of 0·0 would mean that the lists are ordered randomly with respect to each other.

Results from the expert with the "worst" score (the lowest) are presented side-by-side with the consensus opinion appearing above. Only those solutions that the dissenting expert knows about are listed in the comparative consensus. The order of consensus is preserved. In this case, both experts exhibited moderately strong agreement with the consensus.

Users can get an idea of the overall agreement or disagreement among experts by comparing their correlation scores. Users can also see the entire range of acceptable solutions by comparing the consensus and dissenting opinions. In the results below, the first several solutions are in substantially the same order. The user gains more confidence that these may be valid recommendations. If there are wide differences between experts, the user can employ other tools in AQUINAS to compare the appropriate repertory grids to see where the experts' opinions differed.

Dissenting opinion for consultation MEDICAL-AID-ALLEN

Correlation scores for all experts:
DBS 0·86
AJS 0·84

AJS has the most dissenting opinion (0·84 correlation score)

AJS	**/Consensus**
1 ; NORMAL.SINUS.RHYTHM 0·99	/SINUS.RHYTHMS 0·98
2 : SINUS.RHYTHMS 0·98	/NORMAL.SINUS.RHYTHM 0·91
3 : SINUS.BRADYCARDIA 0·98	/SINUS.BRADYCARDIA 0·90
4 : SINUS.TACHYCARDIA 0·96	/SINUS.TACHYCARDIA 0·87
5 : SINUS.ARRHYTHMIA 0·83	/SINUS.ARRHYTHMIA 0·83
6 : 1ST.DEG.HEART.BLOCK 0·68	/1ST.DEG.HEART.BLOCK 0·73
7 : 2ND.DEG.HEART.BLOCK-MOBITZ.II 0·68	/2ND.DEG.HEART.BLOCK-MOBITZ.II.0·64
8 : NODAL.RHYTHM 0·64	/AV.NODE DYSFUNCTION 0·62
9 : SUPRAVENTRICULAR.TACHYCARDIA 0·62	/SUPRAVENTRICULAR.TACHYCARDIA 0·59

10 : AV.NODE.DYSFUNCTION 0·62	/2ND.DEG.HEART.BLOCK-WENCKEBACH 0·57
11 : ATRIAL.FLUTTER 0·60	/NODAL.RHYTHM 0·56
12 : 3RD.DEG.HEART.BLOCK 0·59	/ATRIAL.DYSRHYTHMIA 0·56
13 : ATRIAL.DYSRHYTHMIA 0·56	/ATRIAL.FLUTTER 0·50
14 : 2ND.DEG.HEART.BLOCK-WENCKEBACH 0·52	/3RD.DEG.HEART.BLOCK 0·48
15 : ASYSTOLE 0·45	/VENTRICULAR.FIBRILLATION 0·46
16 : ATRIAL.FIBRILLATION 0·44	/ASYSTOLE 0·45
17 : VENTRICULAR.DYSRHYTHMIA 0·38	/IDIOVENTRICULAR.RHYTHM 0·43
18 : VENTRICULAR.TACHYCARDIA 0·38	/ATRIAL.FIBRILLATION 0·40
19 : IDIOVENTRICULAR.RHYTHM 0·38	/VENTRICULAR.TACHYCARDIA 0·40
20 : VENTRICULAR.FIBRILLATION 0·38	/VENTRICULAR.DYSRHYTHMIA 0·38

3.5. CONSTRAINTS

In AQUINAS, *constraints* are used to represent interactions between traits. Constraints may be specified either by the expert or by the end user. Use of constraints allows the expert to define relationships between traits and allows users to express necessary requirements or observations for solutions in addition to desirable or uncertain ones.

Experts can enter relationships that constrain traits given the values of other traits. These types of constraints can limit the solution set, prune questions, and affect question ordering during consultations. For example, the following response triggered an expert-supplied constraint. If the heart rate is 40 or below, certain rhythms won't occur. If this happens during a consultation, these rhythms are pruned from the solution candidates (their scores are set to zero and not changed). The expert also entered the constraint that the rhythm asystole is eliminated for all heart rates greater than zero. In another example, the expert entered a constraint that prunes traits during consultations. In effect, this stops certain questions from being asked. For instance, if the P.WAVE. CHARACTERISTIC is ABSENT.P.WAVE, then questions about PR.INTERVAL and PR.INTERVAL.CHARACTERISTIC are not asked. These types of constraints entered by the expert can also affect question ordering.

The user can enter constraints during a consultation. Here, for example, the user constrains the solution set to those with heart rates greater than 120 by using the keyword ABSOLUTE. Any solutions associated with heart rates less than 120 will be pruned:

```
please indicate your preference, observation, or expectation for the attribute
HEART.RATE (DBS). The interval is 20 You may also USE.BARCHART for entry.
SLOW.HEART.RATE(0)/FAST.HEART.RATE(200) WT 5
AQU** 120>ABSOLUTE
```

The user can also enter constraints for single values:

```
Please indicate your preference, observation, or expectation for the attribute
HEART.RHYTHM (DBS). You may also USE.BARCHART for entry.
(REGULAR IRREGULAR IRR.IRREGULAR CHAOTIC) WT 5
AQU** REGULAR ABSOLUTE
```

The constraints established by the user, HEART RATE greater than 120 and a HEART.RHYTHM of regular, eliminate most of the solutions (all other solutions received scores of zero):

Test results for consultation MEDICAL-AID-EXAMPLE2

1 : SINUS.TACHCARDIA (0·99)
2 : SUPRAVENTRICULAR.TACHYCARDIA (0·83)
3 : VENTRICULAR.TACHYCARDIA (0·69)

The addition of constraints allows AQUINAS to represent virtually any type of information in repertory grids that could be represented using a rule-based scheme. Constraints will be valuable as we attempt to apply AQUINAS to configuration, design, and planning problems (see Discussion, page 210).

4. Debugging, maintenance, validation and verification

A general problem when modifying knowledge bases is that changes may degrade system performance. This is especially a problem when the knowledge base is large; it may be unclear how changing one item in a knowledge base containing thousands of items will affect overall system performance. A set of consultation review tools in AQUINAS helps alleviate this problem.

Tools for analyzing consultations to improve performance are an important part of the workbench (Shema & Boose, 1988). These tools may modify the knowledge base after checking that existing test cases still give the intended results. Synthesis of traits and test cases help with the validation of the knowledge base. Experiments are underway to evaluate the utility of these tools.

4.1. CASE STORAGE AND PLAYBACK

Information entered by the expert during a consultation may be stored for later playback. The recording and playback facilities test new changes to the knowledge base using old consultations. The description of the case, the values of the traits, and the expert's expectations of the ranking are recorded with each stored case.

Cases can be run in batch mode, where changes made to the knowledge base to improve the performance of one test case may be automatically tested on all the previous cases. The expert can see whether or not new changes result in overall improvement.

4.2. KNOWLEDGE BASE CHANGES

Changes that lead to knowledge base performance improvement (or degradation) take several forms:

Changing ratings in a grid. Judgments about the relationships between solutions and traits may be revised.

Changing a trait's weight. A trait's influence may be increased or decreased by changing its weight. Traits can be temporarily ignored by assigning them a weight of zero.

Adding or deleting a trait. Traits that affect the outcome may be added, deleted, or moved to different levels in the hierarchies. New traits are rated against existing solutions and assigned weights.

Adding or deleting a solution. The potential set of solutions that affect the outcome may be expanded or restricted. New solutions are rated against existing traits.

Changing the expectations of outcomes. The expert may revise the expectation of the system's behavior.

Re-organizing existing knowledge. Knowledge can be reorganized to change its affect on performance. For instance, knowledge can be regrouped under different cases in the knowledge bases hierarchies so that it is applied under different circumstances. Experts may also change their ideas about the range of applicability of portions of the knowledge base.

Review mechanisms fall into several categories—*manual, directed, semi-automated,* and *automated.*

4.3. MANUAL METHODS—INSPECT RESULTS AND REVISE

A variety of methods exist to manually inspect consultation results. The expert may review the results and change the knowledge base directly to try and improve the system's performance.

The rank ordering of the solutions may be displayed in list form (as above), tables, bar charts, or pie charts. Tracing of an ongoing consultation can be done graphically using bar charts to display the current stage of the ranking, or tracing information can be displayed as text. The amount of tracing can be controlled by the expert.

Dynamic bar charts show intermediate consultation results. The effects of each trait can be seen on the solution set. After watching the effect of a particular trait, the expert may manually adjust the repertory grid information to decrease or increase the effect of the trait on the solution set.

Final consultation results may be displayed as an ordered bar chart. This graphic representation helps the expert see the *intervals* between solutions, while textual lists help the expert see the *overall ordering.* Consultation results can also be displayed in normalized form in a pie chart. It is often easier to see the *relative strength* of recommendation for each solution in a pie chart (as opposed to the *absolute strength* of recommendation emphasized in a bar chart).

Each individual ratings' contribution to the solution may be emphasized by highlighting those grid elements which come closest to preference, observations, and constraints given during a consultation. The expert can quickly see the information that contributed to the particular test case results.

All of these lists, graphs, charts, and grids may be saved from consultation to consultation. The expert can compare previous results to current results and see if changes led to improvements. These displays can also be recreated from saved consultations.

4.4. DIRECTED METHODS—ELICIT RESULTS EXPECTATIONS, SCORE RESULTS, REVISE

The correlation between the expert's expectation of the solution ranking *vs* the outcome of a consultation may be calculated. This correlation is used as a measure of the accuracy of the knowledge base performance for a case. Solutions that are out of place can be used to elicit new information. The expert may partially or completely specify the expected outcome of a consultation.

4.4.1. Scoring results

Results are scored against expectations by computing a rank correlation measure and checking to see whether or not the result is statistically significant. Correlation scores show the expert how well the knowledge base performed, given the expectations. Correlation scores recorded over time for many test cases can show whether or not the knowledge base is improving as revisions are made.

4.4.2. New triads based on expectations

AQUINAS calculates the solution that is most out of place and asks the expert for a new trait based on where the solution *should* have appeared in the list. When this new trait is included in a subsequent consultation the out-of-place solution should be closer to the expected place in the list (Boose, 1986).

4.5. SEMI-AUTOMATED METHODS—SUGGESTING CHANGES

4.5.1. Solution results adjustment

Implementation of these methods is in progress. If the expert disagrees with the rank ordering of the solutions, he will be able to modify the bar chart displaying the results of the consultation. His new ordering will be used to find potential changes to ratings or trait weights that would lead to the new ranking. Changes to *repertory grid values* may be suggested that would help push target solutions up or down on the results list. Changes to *trait weights* may be suggested that would help push groups of solutions up or down on the results list. After information is modified the consultation will be replayed and the new and old results will be compared. This method of interaction is similar to a spread-sheet: the expert modifies the answer graphically and the data is used to calculate changes.

4.6. AUTOMATED METHODS—RATING AND TRAIT WEIGHT REVISION; SYNTHESIZING NEW TRAITS

Preliminary results are presented for methods that examine the knowledge in a repertory grid and automatically generate changes that lead to performance improvements.

4.6.1. Automatic rating and weight adjustment

Values of ratings and trait weights may be varied automatically to produce a new repertory grid. Batch test cases are run, and the results of the changes are measured against the expert's expectations. A new change is selected, and the test cases are re-run and scored.

Figure 7 shows the grid modification, test, score, and report cycle. The expert starts by specifying a set of test cases and expected outcomes (a rank-ordered solution set for each test case).

Exhaustively checking every possible combination of ratings and weights is not feasible for even medium-sized grids. For example, it took over 90 h to test a small

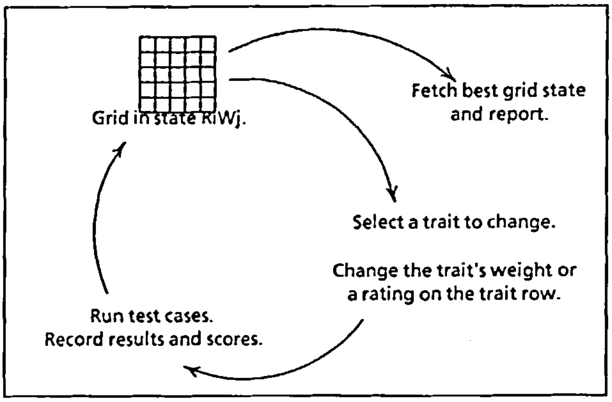


FIG. 7. Trait weights and associated ratings are changed automatically, batch test cases are run, and the results are scored against the expert's expectations for those cases and stored. Heuristics are used to select the next trait-weight-rating change. When finished, the best combination of trait weights and ratings is reported to the expert. "Best" is defined as the highest average correlation score over the test case set for that grid state. The expert can examine the proposed grid and direct AQUINAS to implement all or a portion of the suggested changes.

7×6 grid, sampling ratings (1, 3, and 5 on an ordinal scale from 1 to 5), and using six test cases. So, a hill-climbing heuristic is used to select a new change and decide when to stop work. The heuristic is based on:

—the original trait weight (traits with higher weights tend to be tried sooner);
—the original rating distribution spread in the row;
—the distance from the original grid state to the proposed new grid state (closer states are tried first since it is assumed that the grid from the expert is reasonably close to a final state); and
—previous results (if moving a rating or weight in a certain direction produces better results, try pushing it farther in that direction).

Starting with the expert's grid, one change to a rating or weight is made. The change is selected based on the distance of the proposed change from the original grid; "closer" changes are tried before more distant changes. If the change results in overall performance improvement, a change is made from this new state. If the change does not result in an overall improvement, a different change is tried from the old state. If no changes result in performance improvement, a "peak" has been reached. The grid is randomly changed periodically to keep from only finding local minima; if the random change produces an overall performance improvement, hill climbing proceeds from there. This process runs until a user-specified time limit is up.

When the process is over, several results are reported.

The grid that exhibited the best performance. The expert can examine the new grid to see whether or not the changes make sense. If the expert agrees, all or some of the changes are automatically implemented in the existing grid.

The "distance" between the best grid and the original grid. Trait ratings and weights in the new grid may be slightly different or significantly different from those in the original grid. When there are significant differences, it may suggest that a new trait should be used that exhibits the new ratings and weights (see the discussion of trait synthesis, below).

A sensitivity ranking for the traits. The overall effect each trait has on all the tested combinations is reported. Traits are ranked according to the greatest performance variance and reported. This information can help focus the expert's attention on traits that most affect performance accuracy.

Local minima. Using the hill-climbing approach described above, grids producing local minima are also reported. These grids may contain optimal interactions between a subset of traits.

A "completeness" score for the test case results. A score is given that shows the overall completeness of possible test case results rankings for the best (or any other) grid (for instance, there would be 25 possible outcomes for five possible solutions). This helps the expert see how well the test cases cover the set of solutions. This information can be used to generate new potential test cases (see below).

This technique has produced promising results. Minor sets of changes have yielded significant problem-solving improvement over small sets of test cases.

4.6.2. Analytic and synthetic changes

Using AQUINAS, the best case of knowledge base modification (from a maintenance point of view) is when adjustments made solely to trait values and weights result in improvements of (or at least, do not degrade) all existing cases. These are referred to as *analytic* changes, since no new traits or solutions are added, and a change can be made by analysing relationships between existing traits and solutions. Usually this is not the case; a given change may improve some case performance and degrade other case performance. When this happens, knowledge must be added and/or deleted. Adding new knowledge is referred to as a *synthetic* change.

4.6.3. Synthesizing new traits

Examination of the consultation results makes it possible to hypothesize new traits that would continue to keep the knowledge base correct for existing cases. A new trait is formed internally for a set of solutions; its values are systematically varied and combined with existing traits and tested against existing consultations. If the inclusion of this trait improves the overall scores of existing consultations, the expert is asked to try and name it, and it is included in the knowledge base. Such traits can also be used to start a *new* grid (Shema & Boose, 1988). A trait's original ratings could also be changed so much that the expert feels that the meaning of the trait has changed. Again, the expert would try to name the new trait. Trait change measurements are reported to the expert along with the best grid (see above).

Implementation of the following two methods is in progress.

4.6.4. Consultation synthesis

Examining the contents of previous consultations may suggest new test cases that are significantly different from those already present (that is, their rank orders are significantly different from any existing test cases). The expert could evaluate these synthesized test cases and try to see if they fit any real-world situations. The knowledge base validation effort could be assisted by the expanded coverage of these test cases.

4.6.5. Test case analysis

Comparisons of consultation results might highlight discrepancies in the expected results that the expert has given. Test cases may not be appropriate for the section of the knowledge base under consideration, or the expert may have made an error in his ranking. Validation of the test cases is often as important as the validation of the knowledge base.

5. Other AQUINAS features

5.1. STRATEGIC AND PROCEDURAL KNOWLEDGE

We have performed experiments with AQUINAS to elicit high level strategic knowledge. We took cases and listed them on the top of the grid, and then asked the expert, "What strategy might you use to solve two of CASE1, CASE2, and CASE3, but not the third?" Strategies were elicited as traits, and then the expert rated the applicability of each strategy to each problem case. This method tended to highlight the priorities of various procedures in solving the problem (Figs 8 and 9).

When this information is inverted (rows and columns are exchanged so that strategies become "solutions" and cases become "traits") the type of case can be used to select the appropriate strategy during consultations.

5.2. AQUINAS AND MACHINE LEARNING

Machine learning in AQUINAS takes place in two forms: *interactive* and *automatic.* Interactive forms include implication generation, analysis, and review. Automatic forms include strategies embedded in the inference engine and methods to automatically improve knowledge bases. Analysis and refinement tools in AQUINAS help the expert produce a grid that performs well. Learning mechanisms that start with such knowledge can be very effective. Combining more traditional machine learning techniques with interactive elicitation methods in the same tool

What strategy might you use to solve two of AUTO.ACCIDENT, MEDICAL.CALL, and MOUNTAIN.RESCUE, but not the third?
NEW STRATEGY** **FIND OUT LOCATION**

What is that strategy's opposite as it applies in this case?
NEW STRATEGY** **LOCATION NOT ESSENTIAL**

What is the name of a scale or concept that describes FIND.OUT.LOCATION / LOCATION.NOT.ESSENTIAL?
NEW STRATEGY (CONCEPT)** **LOCATION**

What strategy might you use to solve two of MEDICAL.CALL, MOUNTAIN.RESCUE, and MISSING.CHILD, but not the third?
NEW STRATEGY** **FIND OUT SEVERITY**

What is that strategy's opposite as it applies in this case?
NEW STRATEGY** **SEVERITY NOT ESSENTIAL**

What is the name of a scale or concept that describes FIND.OUT.SEVERITY / SEVERITY.NOT.ESSENTIAL?
NEW STRATEGY (CONCEPT)** **SEVERITY**
:

FIG. 8. Triadic comparison of cases is used to elicit strategies and procedures for a hot-line dispatch portion of the medical aid advisor.

```
              DBS.SOLUTION.TRAIT
|4|5|1|3|1|4   1. (5) LOCATION: FIND.OUT.LOCATION(1)/LOCATION.NOT.ESSENTIAL(5) [ORDINAL
|5|1|2|5|5|3   2. (5) SEVERITY: FIND.OUT.SEVERITY(1)/SEVERITY.NOT.ESSENTIAL(5) [ORDINAL]
|4|5|2|4|1|2   3. (5) ACCESSIBILITY: FIND.OUT.ACCESSIBILITY(1)/ACCESSIBILITY.NOT.IMPORTANT
|2|5|4|5|1|4   4. (5) NUMBER.OF.VICTIMS: FIND.OUT.NUMBER.OF.VICTIMS(1)/NUMBER.OF.VICTIM:
|1|4|3|5|5|2   5. (5) AMOUNT.OF.EQUIPMENT: FIND.OUT.AMOUNT.OF.EQUIPMENT.NEEDED(1)/AMI
.| | | | | |
.| | | | | 6. LOGGING.ACCIDENT
.| | | | 5. PLANE.CRASH
.| | | 4. MISSING.CHILD
.| | 3. MOUNTAIN.RESCUE
.| 2. MEDICAL.CALL .
. 1. AUTO.ACCIDENT .
DBS.SOLUTION       .
```

FIG. 9. Repertory grid of cases (columns), and strategies (rows) for a hot-line dispatch portion of the medical aid advisor. For this problem, strategies are *procedural priorities* for determining different types of information.

provides useful feedback to the user that triggers new insights and that can shed new light on machine learning development.

5.2.1. Interactive learning

Inductive implications between trait values are computed with an algorithm developed by Gaines based on fuzzy logic and information theory (Gaines & Shaw, 1987). A repertory grid is used as a set of examples or learning set. Trait values are viewed as logical predicates, solutions are the operands of the predicates, and ratings are fuzzy truth values. Implications are shown graphically (as in Fig. 2, above) or listed textually. The strength of each implication is shown, either by listing the score or by varying the thickness of the arrow on the graph. Implications show the expert relationships at a higher level of abstraction implied by a repertory grid. If the expert disagrees with an implication, AQUINAS helps the expert refine the grid. Frequently, the expert can think of an exception to the implication (a new solution) that disproves it. This solution is entered, rated, and the implication strength is reduced appropriately. Sometimes implications point out inconsistencies in the way the expert is using a trait. In such cases laddering is used to help the expert decompose inconsistent traits into consistent subtraits.

5.2.2. Automatic learning—reasoning

In a complex knowledge base the expert may not want to rate every possible cell in every possible repertory grid implied by the four hierarchies (every set of four nodes—one from each hierarchy—defines a unique grid cell). Experts tend to rate the leaf cells of hierarchies leaving cells at higher levels unrated. If the hierarchies are deep the expert may rate no more than 10 or 20% of the existing grid cells. However, AQUINAS' inference engine expects all cells to be rated so that solutions may be scored properly for each observation, preference, or constraint. AQUINAS has several mechanisms for "filling in" missing ratings as needed. Lower level ratings can be abstracted to higher levels (induction); ratings of parent values can be inherited down to child ratings (deduction); best guesses can be made by looking at siblings' ratings if they exist or by examining the functional similarity of traits (analogy); users can supply their own application-dependent derivation functions.

These mechanisms are used directly by the inference engine and to produce derived grids that show AQUINAS' best inferences for missing ratings.

5.2.3 Automatic learning—consultations and ID3

An ID3-like algorithm (Quinlan, 1988) may be optionally employed by the reasoning engine during consultations. The mechanism dynamically optimizes question ordering based on the ratio of expected information from answering the question to the cost of obtaining that information. For example, the user can instruct the inference engine to "only ask the three most important questions at this hierarchical level" or to only ask questions that score above a certain threshold of utility.

5.2.4. Automatic learning—automatic grid improvement and the new term problem

In general, learning systems cannot extend or modify their initial vocabulary to generate new descriptors (new terms) when needed. There are two types of new terms: terms resulting from the compilation or decompilation of existing terms and truly new terms that are orthogonal to the original ones. AQUINAS' clustering mechanism can help experts identify compilations of existing terms when experts are asked to label cluster junctions. AQUINAS' automatic grid improvement mechanism addresses part of the second problem. New, unlabeled traits (terms) are produced, and the expert is asked to name them (see discussion above). In this case, AQUINAS identifies the *need for* and *characteristics of* a new term that is guaranteed to improve the performance of the knowledge base and the expert is asked to supply the name of the term. For both types of new terms, allowing the expert to interact with clustering and automatic improvement mechanisms seems to be the key to effectiveness.

5.3. AUTOMATICALLY GENERATING PRODUCTION RULES FROM GRIDS

AQUINAS can transform repertory grids into sets of production rules. These rules can be automatically reformatted for several expert system building tools (KEE, KS-300/EMCYIN, LOOPS, M.1, OPS5, S.1, and others). Certainty factors for rules are generated based on EMYCIN's certainty factor calculus (Adams, 1985). Certainty factor values are based on the relative strength of the rating, the relative weight of the trait, and the overall size of the grid.

Experts find it easier to analyse and maintain knowledge in the more compact form of repertory grids rather than in the form of production rules. In ETS, use of production rules and an internal reasoning engine were the sole methods of reasoning. This method has been replaced in AQUINAS with a maximum entropy-based approach. Rule-based reasoning in AQUINAS can still be used to pre-test knowledge bases destined for other expert system building tools.

5.4. PROCESS GUIDANCE

A subsystem of AQUINAS called the dialog manager contains pragmatic heuristics to guide the expert through knowledge acquisition using AQUINAS. Its help is important in the use of AQUINAS, given the complexity of the AQUINAS environment and the many elicitation and analysis methods available to the expert. The dialog manager makes decisions about general classes of actions and then recommends one or more specific actions providing comments and explanation if desired. This knowledge is contained in rules within the dialog manager in AQUINAS. A session history is recorded so that temporal reasoning and learning may be performed (Kitto & Boose, 1987, 1988*a*,*b*). The dialog manager helps novices learn to use AQUINAS, resulting in more efficient prototyping. We are experimenting with additional mechanisms and representations to provide more sophisticated control and guidance during sessions with AQUINAS (Bradshaw, Boose, Covington & Russo, 1989).

5.5. STEPS IN BUILDING A RAPID PROTOTYPE WITH AQUINAS

A typical sequence of steps when using AQUINAS to build knowledge bases is:

(1) elicit cases and the initial grid (solutions, traits, and ratings):
(2) analyse and expand the initial, single grid;
(3) test the knowledge in the single grid by running consultations;
(4) build hirarchies (structured as solutions and traits in multiple grids) from the first grid if problem is complex;
(5) use several rating value types (transform ordinal ratings to nominal and interval ratings) to represent knowledge;
(6) test knowledge in hierarchies; add, analyse and test knowledge from other experts if warranted by the application;
(7) if multiple experts are used, analyse the similarities and differences among experts; conduct structured negotiation among the experts (Boose, 1986*b*);
(8) edit, analyse, and refine the knowledge base, building new cases;
(9) further expand and refine the knowledge base by successively testing the knowledge to see if the results agree with the expert's opinion.

6. AQUINAS application areas

AQUINAS is useful for *analysis* problems whose solutions may be comfortably enumerated, such as classification, interpretation, and diagnosis. It is not as useful for *synthesis* problems where unique solutions are built up from component parts. Such problems include configuration, planning, scheduling, and design. However, AQUINAS can often be used on the analytic components of larger synthesis problems (Bradshaw *et al.*, 1989).

6.1. DIMENSIONS OF AQUINAS USE

AQUINAS can be used in a number of ways to help facilitate decision-making. Using AQUINAS to help build a large, complex expert system is one such use. However, AQUINAS can also be used as a stand-alone one-shot personal decision aid, a teaching aid, a group data gathering and negotiation tool, a tool for exploring project feasibility, and so on.

AQUINAS is useful in a variety of situations because of the flexibility and ease of use of repertory grids and because of its ability to rapidly gather knowledge and build a prototype consultation system.

Dimensions of use of AQUINAS as a knowledge-based decision making aid are listed below.

Source and user. The source of the knowledge may be the same or different than the user of the knowledge.

Source quantity. One person or more than one person may contribute to the knowledge base.

User quantity. One person or more than one person may use the knowledge base.

Knowledge base longevity. Knowledge bases may be used once and "thrown away" after some problem insight has been achieved or they may be used repeatedly, at different rates (continuously, hourly, daily, weekly, monthly, and so on).

Knowledge base update frequency. Knowledge may be updated with varying rates of frequency. For example, a "frozen" knowledge base may be used for a long time before it is updated or a "living" knowledge base may be updated continuously (a large, commercial, time-share knowledge base with few updates; a marketing trends knowledge base with periodic updates; a stock market knowledge base with continuous updates).

Stand-alone. The tool can be used for stand-alone decision use or used with other software systems (expert system shell; spreadsheet; database management system).

Commercial knowledge base use. The knowledge base can be generated for commercial, external or internal, or personal use.

Consulting use. The decision maker can use the tool directly or a consultant can use the tool in support of the decision maker (decision consulting; education and training).

Empty tool or primed tool. The tool can be used as an empty general decision aid or primed with knowledge for specific applications. Knowledge templates or full or almost-full knowledge bases may be delivered to a customer. Some examples are knowledge for specific domains at varying levels of abstraction (general diagnosis, engine diagnosis, jet engine diagnosis), proprietary or commercially valuable knowledge, surveys (fill in the ratings and return the grid to the surveyors), or use of domain-tailored dialogs (such as the original use of repertory grids in psychotherapy where patients filled in specific roles based on supplied descriptions).

Process and product. These tools can be used in the process of making a knowledge base as well as actually delivering knowledge (feasibility analysis, education and training of experts in the knowledge engineering process).

ETS (based on single grids) and AQUINAS (handling multiple hierarchical grids) have been used to generate hundreds of small and medium sized knowledge-based systems in the following categories (Boose, 1988). The categories are derived using different combinations of values for the above dimensions.

(1) *decision aid for one-shot decisions*
A grid tool is used to help people gain insight while making decisions: What employee should I assign to this project? What stocks should I invest in? What car should I buy? How can I better represent my products to my customers?

(2) *Stand-alone knowledge-based system development and delivery tool*
A grid tool is used to help develop, deliver, and maintain a knowledge-based system in a cost-efficient manner.

(3) *Group decision aid*
A grid tool is used as an aid to facilitate rapid, documented group decisions: On what software and hardware environment should this project be implemented? What research should our company pursue? How should capital assets be assigned?
(4) *Large-scale on-line knowledge bases for a community of users*
A grid tool can be used to develop and maintain a large dial-in knowledge-and-databases for a large number of users (stock market advising, insurance claims information).
(5) *Feasibility analysis and project exploration tool*
A grid tool is used to help assess project feasibility: What project idea is technically feasible? What project ideas are possible in this general idea? Given a project idea, what kinds of knowledge will be involved? Does it make sense to invest resources in this project?
(6) *Expert system building tool (shell) front-end*
A grid tool can be used as a front-end to develop knowledge for another expert system shell (S.1, M.1, OPS5, etc.).
(7) *Teaching aid*
A grid tool can be used to help teach others (especially experts and beginning knowledge engineers) quickly and painlessly about the concepts involved in knowledge-based systems and knowledge engineering: What are production rules? What is a consultation system? How can an expert's knowledge be structured and modified? How can systems be tested and validated?
(8) *Situation insight*
A grid tool can be used to discover important or controlling factors in a situation; these factors can help the user understand, control, or change the situation (job satisfaction and motivation self-analysis, psychotherapy, counseling, relations with colleagues).

Large knowledge bases (containing thousands of judgments) are typically generated as the result of group decision applications. A 20×20 grid contains 400 ratings; when such grids are collected from multiple experts the resulting knowledge base contains many thousands of ratings. For example, a past AI tool advisor system included over four thousand ratings from four experts. Larger systems from single users tend to have more hierarchical structure and fewer ratings. For purposes of comparison, a rating can be thought of as the rough equivalent of a context-dependent rule in a rule-based system.

7. Discussion and future work

AQUINAS inherits the advantages of ETS: rapid prototyping and feasibility analysis, vocabulary, solution and trait elicitation, interactive testing and refinement during knowledge acquisition, implication discovery, conflict point identification, expert system shell production, and generation of expert enthusiasm (Boose, 1986*a*). It is much easier for users to learn knowledge-based system concepts by using AQUINAS than through reading books or attending classes (i.e. rules are automatically generated and used dynamically in consultations; new vocabulary is incrementally introduced; Kitto, 1989). AQUINAS is as easy for novice users to learn to use as was ETS; the dialog manager helps more sophisticated users explore advanced features.

The first expert took about 2 h to enter, analyse, and refine the knowledge for the heart dysrhythmias case shown here. Working from the first expert's solutions and traits (but not ratings or weights), the second expert took about an $1\frac{1}{2}$ hr to enter, analyse, and refine the knowledge for this case. A knowledge engineer took less than 2 hr to draw and enter the graphic displays.

Knowledge from multiple experts may be rapidly combined using AQUINAS. Users may receive dissenting as well as consensus opinions from groups of experts, thus getting a full range of possible solutions. Disagreement between the consensus and the dissenting opinion can be measured to derive a degree of conflict for a particular consultation. The system can be used for cost-effective group-data gathering (Boose, 1986*b*; Schuler *et al.*, 1989).

Elicitation, structuring, analysis, and testing of knowledge is based on specific cases. When knowledge in AQUINAS is updated, it is done so with respect to a specific case. Addition of new knowledge in this way that can be strictly controlled by the expert; the tendency for local changes to degrade other cases is thus curbed.

7. FUTURE DIRECTIONS

In the future we plan to extend AQUINAS in several areas:

7.1.1. Capturing explanations

Templates will be used to capture comments from experts when making judgments (ratings, weights, traits from triads). Comments will be examined to help maintain the knowledge base and will be played back during consultations to provide explanations for the end-user. Applicable work on classifying explanations has been reported by Schank (1986) and Kass & Leake (1987).

7.1.2. Further work in acquiring and analysing knowledge from multiple experts

In addition to AQUINAS' current capabilities to analyse and use knowledge from multiple experts, we plan to implement the following features:

—elicit knowledge in parallel (on-line) from several experts (Chang, 1986) in the same or different fields;
—analyse subsumption and overlap of experts' knowledge in multiple fields (similar to SocioGrids in KSS0, Shaw & Gaines, 1987);
—support on-line structured negotiation among experts to solve problems or make recommendations;
—discover the most important aspects of individual knowledge when compared to group knowledge (special cases);
—elicit and merge knowledge dynamically for use in on-line applications.

7.1.3. Knowledge acquisition for synthesis problems

Attempts are being made to use grid tools to elicit knowledge for synthesis problems (Bradshaw *et al.*, 1989). For example, we are experimenting with allowing the user to put arbitrary computations in the cells of repertory grids in AQUINAS (spreadsheet-like computations, database access calls, functions that sample sensors) or to call specialized tools for alternative generation and constraint satisfaction.

Eventually, we hope to blend together techniques from other tools (such as MDIS, Antonelli, 1983, and SALT, Marcus, 1985, 1987) to help elicit knowledge for aspects of design problems:

—acquisition of causal models and design constraints;
—selecting alternative design concepts based on competing criteria such as reliability, maintainability, cost, manufacturability;
—acquiring human expertise in validation and verification of designs;
—acquiring knowledge for design evaluation and change recommendation;
—acquiring knowledge for document generation and evaluation;
—capturing historical data in a "corporate memory" database to help solve future design problems.

Other knowledge acquisition tools are useful in producing knowledge-based systems more rapidly than manual interviewing methods. Knowledge acquisition tools may be characterized by the problem tasks for which they are designed to gather knowledge and the problem-solving method that uses the knowledge to solve a problem (Boose, 1988*b*). Particular knowledge acquisition tools can be viewed as linking tasks and methods.

AQUINAS attempts to capture knowledge for most types of analysis problems; other systems such as MOLE and MDIS derive their power from knowing more about the application domain.

We intend to build a knowledge acquisition environment that includes specific domain knowledge for specialized application areas and can acquire knowledge for synthetic problems, combining features from other knowledge acquisition tools. Development of the AQUINAS workbench will continue in an incremental fashion. Techniques will be continuously integrated and refined to build an increasingly more effective rapid knowledge acquisition environment.

Thanks to Kathleen Bradshaw, Brian Gaines, Cathy Kitto, Ted Kitzmiller, Art Nagai, Doug Schuler, Mildred Shar, Suzzanne Shema, and Lisle Tinglof-Boose, for their contributions and support. AQUINAS was developed at the Knowledge Systems Laboratory, advanced Technology Center, Boeing Computer Services in Seattle, Washington.

References

Adams, J. (1985). Probabilistic reasoning and certainty factors. In B. Buchanan & E. Shortliffe, Eds. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, MA: Addison-Wesley.

Antonelli, D. (1983). The application of artificial intelligence to a maintenance and diagnostic information system (MDIS). *Proceedings of the Joint Services Workshop on Artificial Intelligence in Maintenance,* Boulder, CO.

Boose, J. H. (1984). Personal construct theory and the transfer of human expertise. *Proceedings of the National Conference on Artificial Intelligence (AAAI-84),* Austin, Texas, pp. 27–33.

Boose, J. H. (1985). A knowledge acquisition program for expert systems based on personal construct psychology. *International Journal of Man–Machine Studies,* **23,** 495–525.

BOOSE, J. H. (1986*a*). *Expertise Transfer for Expert System Design*. New York: Elsevier.

BOOSE, J. H. (1986*b*). Rapid acquisition and combination of knowledge from multiple experts in the same domain. *Future Computing Systems Journal*, **1**, 191–216.

BOOSE, J. H. (1988). Uses of repertory grid-centred knowledge acquisition tools for knowledge-based systems, special issue on the 2nd Knowledge Acquisition for Knowledge-Based Systems Workshop, 1987. *International Journal of Man–Machine Studies*, **29**, 287–310.

BOOSE, J. H. & BRADSHAW, J. M. (1987). Expertise transfer and complex problems: using AQUINAS as a knowledge acquisition workbench for expert systems, special issue on the 1st Knowledge Acquisition for Knowledge-Based Systems Workshop, 1986, Part 1. *International Journal of Man–Machine Studies*, **26**, 3–28; also in J. H. BOOSE & B. R. GAINES, Eds (1988). *Knowledge-Based Systems: Knowledge Acquisition Tools for Expert Systems*, Vol. 2, pp. 39–64. New York: Academic Press.

BRADSHAW, J. M. & BOOSE, J. H. (1989), Decision analysis techniques for knowledge acquisition: combining information and preferences using AQUINAS, special issue on the 2nd Knowledge Acquisition for Knowledge-Based Systems Workshop, 1987, *International Journal of Man–Machine Studies*, in press.

BRADSHAW, J. M., BOOSE, J. H., COVINGTON, S. P., RUSSO, P. J. (1989). How to do with grids what people say you can't, special issue on the 3rd Knowledge Acquisition for Knowledge-Based Systems Workshop, 1988. *International Journal of Man–Machine Studies*, in press.

BUCHANAN, B. & SHORTLIFFE, E., Eds (1985). *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Reading, MA: Addison-Wesley.

CHANGE, E. (1986). Participant systems. *Future Computing Systems*, **2**, No. 1.

CLANCEY, W. (1986). Heuristic classification. In J. KOWALIK, Ed. *Knowledge-Based Problem-Solving*. New York: Prentice-Hall.

DAVIS, R. & LENAT, D. B. (1982). *Knowledge-Based Systems in Artificial Intelligence*. New York: McGraw-Hill.

GAINES, B. R. & SHAW, M. L. G. (1987). Induction of inference rules for expert systems. *Fuzzy Sets and Systems*, **8**, 315–328.

GAINES, B. R. & SHAW, M. L. G. (1981). New directions in the analysis and interactive elicitation of personal construct systems. In M. L. G. SHAW, Ed. *Recent Advances in Personal Construct Technology*, New York: Academic Press.

KASS, A. & LEAKE, D. (1987). Types of explanations. Yale Computer Science Report YALEU/CSD/RR No. 523.

KELLY, G. A. (1955). *The Psychology of Personal Constructs*. New York: Norton.

KITTO, C. M. & BOOSE, J. H. (1987). Heuristics for expertise transfer: the automatic management of complex knowledge acquisition dialogs, special issue on the 1st Knowledge Acquisition for Knowledge-Based Systems Workshop, 1986, Part 2, *International Journal of Man–Machine Studies*, **26**, 183–202; also in J. H. BOOSE & B. R. GAINES, Eds (1988). *Knowledge-Based Systems: Knowledge Acquisition Tools for Expert Systems*, vol. 2, pp. 175–194. New York: Academic Press.

KITTO, C. M. & BOOSE, J. H. (1988*a*). Heuristics for expertise transfer: the automatic management of complex knowledge acquisition dialogs. In J. H. BOOSE & B. R. GAINES, Eds. *Knowledge-Based Systems: Knowledge Acquisition Tools for Expert Systems*. Vol. 2. London: Academic Press.

KITTO, C. M. & BOOSE, J. H. (1988*b*). Selecting knowledge acquisition tools and strategies based on application characteristics, special issue on the 2nd Knowledge Acquisition for Knowledge-Based Systems Workshop, 1987. *International Journal of Man–Machine Studies*, in press.

KITTO, C. M. (1989). Progress in automating knowledge acquisition tools: how close are we to replacing the knowledge engineer?, special issue on the 3rd Knowledge Acquisition for Knowledge-Based Systems Workshop, 1988. *International Journal of Man–Machine Studies*, in press.

MARCUS, S., MCDERMOTT, J. & WANG, T. (1985). Knowledge acquisition for constructive systems. *Proceedings of the 9th Joint Conference on Artificial Intelligence,* Los Angeles, CA, pp. 637–639.

MARCUS, S. (1987). Taking backtracking with a grain of SALT, special issue on the 1st Knowledge Acquisition for Knowledge-Based Systems Workshop, 1986, Part 3, *International Journal of Man–Machine Studies,* **26,** 383–398; also in J. H. BOOSE & B. R. GAINES, Eds (1988). *Knowledge-Based Systems: Knowledge Acquisition Tools for Expert Systems,* vol. 2, pp. 211–226. New York: Academic Press.

QUINLAN, J. R. (1988). Simplifying decision trees. In B. R. GAINES, & J. H. BOOSE, Eds. *Knowledge-Based Systems: Knowledge Acquisition for Knowledge-Based Systems,* vol. 1. London: Academic Press.

SAATY, T. L. (1981). *The Analytic Hierarchy Process.* New York: McGraw-Hill.

SCHANK, R. C. (1986). *Explanation Patterns: Understanding Mechanically and Creatively.* Englewood Cliffs, NJ: Lawrence Erlbaum.

SCHULER, D. C., RUSSO, P. J., BOOSE, J. H. & BRADSHAW, J. M. (1989). Using personal construct techniques for collaborative evaluation, special issue on the 3rd AAAI-Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop, 1988. *International journal of Man–Machine Studies,* in press.

SHAW, M. L. G. & GAINES, B. R. (1987). Techniques for knowledge acquisition and transfer, special issue on the 1st Knowledge Acquisition for Knowledge-Based Systems Workshop, 1986, Part 5. *International Journal of Man–Machine Studies,* **27,** 251–280.

SHEMA, D. B. & BOOSE, J. H. (1988). Refining problem-solving knowledge in repertory grids using a consultation mechanism, special issue on the 2nd Knowledge Acquisition for Knowledge-Based Systems Workshop, 1987. *International Journal of Man–Machine Studies,* **29,** 447–460.