M. Henrion
R.D. Shachter
L.N. Kanal
J.F. Lemmer
Editors

Uncertainty in Artificial Intelligence 5

Machine Intelligence and Pattern Recognition Series, volume 10

North-Holland

Uncertainty in Artificial Intelligence 5
M. Henrion, R.D. Shachter, L.N. Kanal, and J.F. Lemmer (Editors)
© Elsevier Science Publishers B.V. (North-Holland), 1990

Knowledge Acquisition Techniques for Intelligent Decision Systems: Integrating *Axotl* and *Aquinas* in *DDUCKS*

Jeffrey M. Bradshaw, Stanley P. Covington, Peter J. Russo, and John H. Boose
Advanced Technology Center, Boeing Computer Services
P.O. Box 24346, M/S 7L-64, Seattle, Washington 98124 USA (206) 865-3422;
jbrad@atc.boeing.com

ABSTRACT

The effective application of current decision tree and influence diagram software requires a relatively high level of sophistication in the theory and practice of decision analysis. Research on intelligent decision systems aims to lower the cost and amount of training required to use these methods through the use of knowledge-based systems; however, application prototypes implemented to date have required time-consuming and tedious hand-crafting of knowledge bases. This paper describes the development of *DDUCKS*, an "open architecture" problem-modeling environment that integrates components from *Axotl*, a knowledge-based decision analysis workbench, with those of *Aquinas*, a knowledge acquisition workbench based on personal construct theory. The knowledge base tools in *Axotl* can be configured with knowledge to provide guidance and help in formulating, evaluating, and refining decision models represented in influence diagrams. Knowledge acquisition tools in *DDUCKS* will allow the knowledge to be efficiently modeled, more easily maintained, and thoroughly tested.

1. INTRODUCTION

1.1. Progress in Automated Decision Analysis

The most promising approach to dealing with decision complexity in a consistent, general-purpose manner is decision analysis (Howard & Matheson, 1984). In the past few years, researchers and developers have made important theoretical advances and have implemented several successful systems for automated support of the decision analysis process (review by Horvitz, Breese, & Henrion, 1988; Neapolitan, 1990). Although a thorough discussion is beyond the scope of this paper, we wish to review three of the important developments that have led to the current state of the art. Following this, we will explain why we think that the development of automated knowledge acquisition tools is crucial to the future of efforts to deliver decision analysis methodology to a wider spectrum of decision makers and domains.

Decision tree software. The development of decision tree software (Figure 1) represented an important milestone in automated decision analysis support (Olmsted, 1982; McNamee & Celona, 1987). Through the use of general-purpose commercial tools, decision analysis techniques have become more widely known and used than ever before. At the same time, there are several drawbacks to the use of decision trees as a representation device. For one thing, they grow exponentially with problem size, making them impractical for problems of significant size. Additionally, the tree metaphor for decision problems often leads participants to confuse chronology with the order of probabilistic expansions.

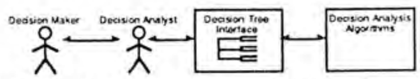

Figure 1. Decision analysis techniques have become more widely used through the availability of decision tree software.

Influence diagram software. Conceived by Howard and Matheson (1981), influence diagrams represent an important advance in the representation of decision problems (Figure 2). While retaining the essentials of the decision theoretic mathematical foundation developed for decision tree manipulation, they provide several advantages over decision trees. Technically, they are superior in that they can explicitly represent and exploit conditional independence relationships between variables. From a practical point of view, they provide a clear and intuitive way of communicating the structure of a decision model. With the advent of commercial software (e.g., Shachter, 1986b) and the assumption of continued success by researchers developing tractable methods for evaluating large influence diagrams (Horvitz, Breese, & Henrion, 1988), we expect influence diagrams to eventually replace decision tree representations for most applications.

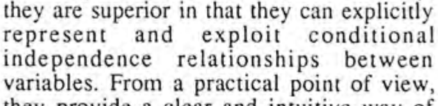

Figure 2. Influence diagrams provide both technical and practical advantages over decision trees.

Intelligent decision systems. Unfortunately, the effective application of both decision tree and influence diagram software requires a relatively high level of sophistication in the theory and practice of decision analysis. These tools contain some of the *algorithms* of decision analysis practice, but cannot embody the *experience* and *intuition* of decision analysis professionals in formulating and appraising decision models. Also, because these tools cannot conveniently store and reuse domain expertise, they cannot exploit the similarity between recurring decisions in the same domain. New decisions are typically modeled from scratch.

The importance of these issues led Holtzman (1989) to define an approach for a third generation of automated decision analysis software called *intelligent decision systems* (IDS)[1]. An IDS combines a set of automated decision analysis tools with a knowledge-based system that helps decision makers without extensive training in decision analysis build, evaluate, and refine decision models in some well specified domain (Figure 3). To build an IDS application for a class of decision problems, decision analysts and domain experts work with a knowledge engineer to configure the system with "application-independent"[2] knowledge (i.e., knowledge of decision analysis tools and methodology) and application-specific knowledge (i.e., knowledge about a particular domain). Once built, these knowledge bases can be used again and again to provide guidance and help during consultations with decision makers. Task-level consultation interfaces pose questions and interpret results in language and graphical presentations tailored to the decision maker, rather than in terms of standard decision analysis concepts.

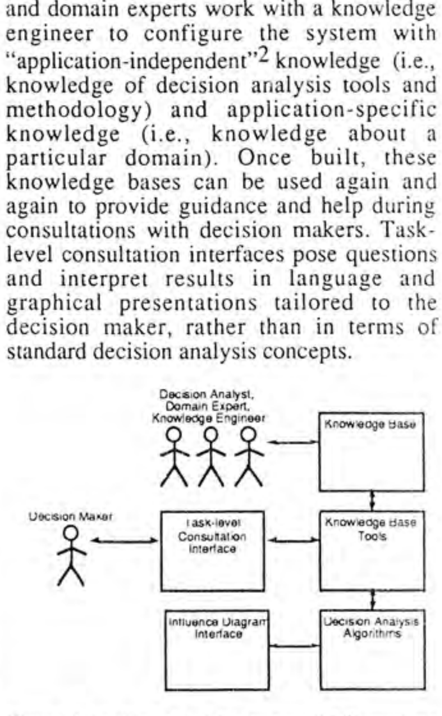

Figure 3. Intelligent decision systems (IDS) combine decision analysis tools with a knowledge-based system

Through implementing a general-purpose IDS architecture *(Axotl)* and applying it to the domain of R&D project selection within The Boeing Company *(PIE)* we have become convinced that decision analysis can be delivered effectively and economically through the use of knowledge-based systems (Bradshaw & Holtzman, 1987; Bradshaw, Covington, & Russo, 1989). We found that

[1] Related approaches to combining decision analysis and knowledge-based systems have been developed by Breese (1987), Keeney (1986), Moore & Agogino (1987), and Wellman (1986).

[2] Of course, no knowledge-base is truly application-independent; perhaps "multiple-use" would be a better term.

the development of knowledge bases for such a system was relatively straightforward compared to other knowledge-based systems we had created because we were dealing with a methodology (decision analysis) that was mature and rigorously defined. However, we were dissatisfied that so much hand-crafting of knowledge bases had to be performed by knowledge engineers. Through the development of knowledge acquisition tools, we hoped that much more of the knowledge base could be constructed by domain or decision analysis experts themselves.

1.2. Knowledge Acquisition for Fourth-Generation Decision Analysis Support Tools

Figure 4 shows how automated knowledge acquisition tools would fit within an architecture for advanced decision analysis support tools. Knowledge acquisition tools could help cut down the lengthy and error-prone revise-and-review cycle in the development of such systems, making delivery of IDS applications practical on a large scale. While it is unrealistic to expect that the role of knowledge engineers would entirely disappear, their participation in many aspects of knowledge base development and maintenance could be minimized. Figure 4 explicitly includes interfaces to conventional software and external data. The success of future systems in practical applications will depend on whether they can be effectively integrated with other software such as databases, spreadsheets, and hypermedia environments.

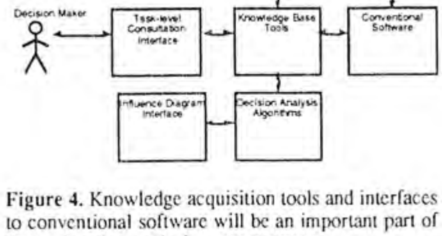

Figure 4. Knowledge acquisition tools and interfaces to conventional software will be an important part of future decision analysis support systems.

Knowledge acquisition research addresses the problem of designing appropriate representations and procedures to facilitate creation, validation, verification, and maintenance of knowledge bases over the lifetime of a knowledge-based system[1]. Thus, it might be said that researchers are attempting to do for knowledge engineering what CASE is doing for traditional software engineering (Bradshaw & Boose, 1989; Gaines, 1988).

Current work in knowledge acquisition emphasizes that creation of knowledge bases is a constructive modeling process and not simply a matter of "expertise transfer" (Bradshaw & Boose, 1990a, b). Automated knowledge acquisition tools augment the facilities typically available in knowledge-base development environments by providing interfaces that *constrain* the interaction between the tool and the expert or knowledge engineer. This support for the modeling process is usually achieved through the use of:

- role-limiting methods
- mediating knowledge representations
- interviewing techniques
- analysis tools.

Role-limiting methods. Many knowledge acquisition tools derive their power from relying on a well defined problem-solving model that establishes and controls the sequences of actions required to perform some task (Gruber, 1988; Karbach, Linster & Voss, 1990; Klinker, 1989). For example, SALT (Marcus, 1988) is based on a method for design called "propose-and-revise", while MOLE (Eshelman, 1988) uses a method of heuristic classification called "cover-and-differentiate". The problem-solving method defines the kind of knowledge applicable within each step, thereby making explicit the different roles knowledge plays. Once these roles are defined, knowledge acquisition tools

[1] Boose (1990) is a survey of current knowledge acquisition techniques and tools.

appropriate to each kind of knowledge are designed.

Acquirable knowledge representations. Knowledge acquisition tools attempt to minimize the problem of representation mismatch, the disparity between a person's task-level description of the problem and its realization in some computable form (Gruber, 1988). This problem is addressed by designing interfaces that bear a strong resemblance in appearance and procedure to the original task (e.g., cancer-therapy protocol forms in OPAL (Musen, 1988); engineering notebooks in vmacs (Sivard, Zweben, Cannon, Lakin & Leifer, 1989)) or by relying on some familiar, high-level, generic knowledge representation metaphor (e.g., repertory grids in ETS, (Boose, 1985, 1986)). Structured high-level representations are used to maintain the knowledge base, but if necessary they can be transformed into other forms such as rules. Transformation of knowledge into multiple forms or perspectives for visualization purposes is useful as a means of facilitating insight.

Interviewing techniques. A number of interviewing techniques have been developed or borrowed from fields such as psychology in order to guide experts through the knowledge acquisition process (Meyer, Booker & Bradshaw, 1990). For example, some of the techniques originally developed by Kelly (1955) to discover the conceptual structure of clients in psychotherapy have been applied to knowledge acquisition.

Analysis tools. Both formal and heuristic modes of analysis are available within many knowledge acquisition tools as a means of verifying and refining the knowledge base. Shaw and Gaines (1988), for instance, have developed a methodology for analyzing areas of consensus, conflict, correspondence, and contrast between the conceptual systems of two or more experts. Many knowledge acquisition tools have also been integrated with consultation systems, making extensive performance testing of the system possible (e.g., Boose, Shema & Bradshaw, 1989).

1.3. *DDUCKS*: An Integrated Environment for Knowledge Acquisition and Decision Analysis

In the Boeing Advanced Technology Center, we have undertaken development of an environment called *DDUCKS* (Decision and Design Utilities for Comprehensive Knowledge Support)[1]. The environment integrates components from *Aquinas*, a knowledge acquisition workbench based on personal construct theory (Boose & Bradshaw, 1987; Boose, Shema & Bradshaw, 1989; Boose, Bradshaw, Kitto, & Shema, 1990), with those of *Axotl*, a knowledge-based decision analysis workbench (Bradshaw, Covington, and Russo, 1989). The "open architecture" design of the *DDUCKS* environment allows knowledge-based coordination of local or networked applications such as spreadsheets, databases, or hypermedia software. One of *DDUCKS'* components, MANIAC (MANager for InterApplication Communication) supports asynchronous and synchronous communication between any number of multitasking applications. *DDUCKS* runs on Apple Macintosh II hardware; subsets of *DDUCKS* run in other environments.

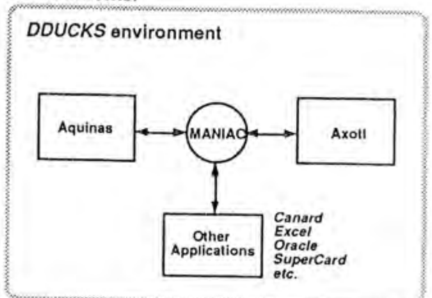

Figure 5. *DDUCKS* integrates components from *Axotl*, *Aquinas*, and other applications.

In the remainder of this paper, we will describe the decision analysis workbench and knowledge base tools of *Axotl*. Then we will outline the knowledge acquisition tools being developed to support the creation of content and process knowledge bases for applications of *Axotl*.

[1] Either the first or second *D* in *DDUCKS* is silent, depending on whether one using the tool in a decision or design context.

2. APPROACH

2.1. *Axotl:* A Knowledge-Based Decision Analysis Workbench

Axotl combines a general-purpose decision analysis workbench with knowledge base tools to assist individuals consulting with the system about a specific problem. The first version of *Axotl* was developed as part of a joint effort by Boeing Computer Services and Strategic Decisions Group (SDG) to build a flexible, general-purpose intelligent decision system that could be applied to internal R&D management decisions. The application is called PIE, for Project Investment Evaluator. Boeing Computer Services, with its experience in building knowledge-based systems (Bradshaw & Boose, 1990b), contributed expertise in software engineering, knowledge engineering, and Boeing-specific knowledge on this project. SDG has a major practice in applying decision analysis to R&D management decisions. In addition, SDG staff have pioneered the theory and practice of decision analysis (Howard & Matheson, 1984) and had previously implemented intelligent decision system application prototypes (Holtzman, 1989).

Following the successful demonstration of knowledge-based guidance of a decision analysis consultation in mid-1988, it was decided that Boeing and SDG would pursue further development work on the system independently. Since that time, we have made a number of extensions to *Axotl,* and have developed *DDUCKS* as the integrating framework for linking *Axotl* to *Aquinas* and other applications. We have loaned *DDUCKS* to a medical non-profit organization for evaluation of its applicability to bone marrow transplant follow-up care (Sullivan & Shulman, 1989). We are also applying portions of the environment to facilitate demonstrations of design knowledge capture for a NASA-sponsored Corporate Memory Facility project (Bradshaw, Boose, Covington & Russo, 1989; Boose, Bradshaw, Shema, & Covington, 1989; Shema, Bradshaw, Boose & Covington, 1990), and in business process management in a Boeing quality improvement context (Bradshaw, Kipersztok, Nguyen, & Holm, 1990). SDG has also enhanced their version of the software, and is developing a commercial application to R&D management, called *R&D Analyst*™[1]. Holtzman and Seiver (1988) have developed a successful application of the system that assists critical care clinicians in ventilator management decision-making.

Axotl was developed on the Apple Macintosh II (*MacXotl*) and runs on all platforms that support ParcPlace Smalltalk-80 (e.g., Sun, Apollo, Hewlett-Packard, IBM '386 compatibles). We will first describe the decision analysis workbench, then the knowledge base tools.

2.1.1. Decision Analysis Workbench

The decision analysis workbench contains a graphical editor that is used for creating and refining models of alternatives, preferences, and uncertainties relevant to a specific decision. These models, called *influence diagrams,* are solved to obtain recommended actions in a way that is consistent with probability and utility theory (Howard & Matheson, 1981).

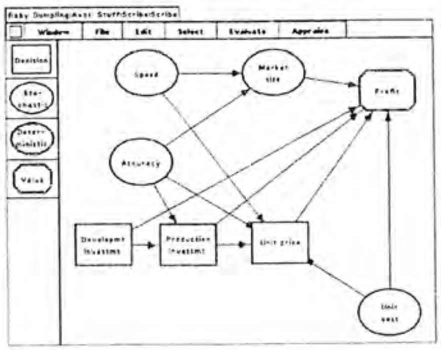

Figure 6. An influence diagram for an R&D investment decision.

Figure 6 shows a screen snapshot of an *Axotl* influence diagram (Howard & Matheson, 1981) representing a prototype R&D project decision problem. The problem is to determine an investment strategy for "Scribe" an automated speech-to-text

[1] *R&D Analyst* is a trademark of Strategic Decisions Group. All rights reserved.

transcriber, taking technical risks and market uncertainties into account. The investment strategy is composed of three decisions (development investment level, production investment level, and unit price), represented by rectangular nodes on the diagram. Oval nodes represent technical risk variables (accuracy, speed), production uncertainties (unit cost), and market uncertainties (market size). The eight-sided node labeled "Profit" has been designated as the criterion to maximize in evaluating the decision model to determine an optimal policy. Arrows between nodes represent relationships where influence or information is imparted from one variable to another. For example, arrows from "Speed" and "Accuracy" to "Market size" represent the assertion that the degree of technical achievement in these areas will affect the size of the market for "Scribe". The arrows from "Speed" and "Accuracy" to "Unit price" indicate that these uncertainties will be known at the time a pricing decision is made. An additional type of node, not shown in this diagram, can represent a deterministic function. This facilitates transparent links from the influence diagram to external procedures or to programs such as spreadsheets and databases.

The method of solving influence diagrams implemented in *Axotl* incorporates a new approach developed by Smith (1988) that allows a wide range of questions to be answered directly from the diagram. It preserves the entire underlying joint distribution during solution and inference procedures, rather than just the value lottery and decision policy as is usually done (e.g., Shachter, 1986a). The distribution editor, used to structure conditional probability distributions, is another feature unique to *Axotl*. It introduces the concepts of *coalescence* (i.e., sharing of atomic distributions) and *clipping* (i.e., explicit pruning of impossible or unnecessary conditioning paths and their atomic distributions).

One of the most significant results of *Axotl* development was the formulation of generic procedures for the use of influence diagrams in valuing information and control (J. Matheson (1988) discusses some of these issues). An approach was developed to allow the automated conversion of any influence diagram to "Howard Canonical Form" so that value-of-perfect-information or imperfect-information questions could be answered for any variable. Based on an understanding that value-of-control calculations can readily be interpreted only for causal influence diagrams, we developed heuristic techniques that specified when it was appropriate to ask value-of-perfect-control or -imperfect-control questions and to formulate generic procedures for answering them. Eventually, these techniques could be fully automated.

2.1.2. Knowledge Base Tools

When configured with the appropriate knowledge, the knowledge base tools in *Axotl* guide decision makers through the process of formulating, evaluating, and refining a decision model in a specific domain. The model is kept to tractable size by deliberately limiting the problem domain for a given application and using heuristic methods represented in knowledge bases to select and prune variables during influence diagram construction. These heuristics can be regarded as instances of the types of categorical methods described by Szolovitz and Pauker (1978):

> "When the complex problems need to be addressed—which treatment should be selected, how much of the drug should be given, etc.—then causal or probabilistic models are necessary. The essential key to their correct use is that they must be applied in a limited problem domain where their assumptions can be accepted with confidence. Thus, it is the role of categorical methods to discover what the central problem is and to limit it as strongly as possible; only then are probabilistic techniques appropriate for its solution."

We distinguish between two major types of knowledge in the knowledge base: content and process.

Content knowledge. Content knowledge is substantive; it is the *what* of decision model building during a consultation. Internally, content knowledge is represented as a set of partially defined influence diagram variables for a class of decisions, their interrelationships, and the conditions under which they may be added to or removed from the influence diagram being constructed.

During the knowledge acquisition phase of the PIE project, we constructed a "big influence diagram" showing all of the structure that would be considered during the consultation process. Once agreement was reached about this structure, it was further broken down into smaller, overlapping groups of related variables along with some possible value models. We found that the "big influence diagram" for R&D project decisions had three distinct stages (research, development, application) and three possible value models (contract, market, simple value).

These value models and groups of influence diagram variables serve as the building blocks for the first-cut decision model that is constructed during subsequent consultations with a decision maker. Through a process called *template development*, the system ask the decision maker questions about the project that will help it determine the stage of development and the appropriate value model (D. Matheson, 1988). The system uses this project information to construct a generic, skeletal template. The dialogue continues to help decision makers expand and refine the initial model according to their specific circumstances.

Process knowledge. Process knowledge is strategic; it is the *how* of decision model building during a consultation. It consists of generic and domain-specific expertise about what to do at each stage of the decision-making process. Internally, process knowledge is represented by activity graphs and rules. Figure 7 represents a view of the process of decision making for R&D projects that we used in developing the PIE application.

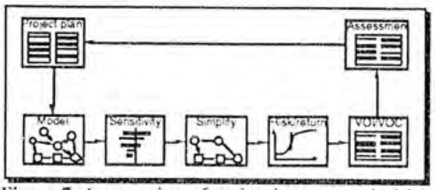

Figure 7. An overview of project investment decision making.

The cycle shown in Figure 7 represents general process knowledge about what to do at each stage of the R&D decision making process. We used the detailed process knowledge we acquired from decision analysis and Boeing experts to build a prototype R&D project selection decision activity graph (Figure 8) and rule base. Consultations using this knowledge base proved the ability of PIE to pose a series of questions to a Boeing manager about a project under consideration and to formulate, evaluate, and appraise an appropriate influence diagram model based on answers to these questions. The decision analysis tools ran under control of the knowledge base, and the appropriate decision analysis task modules were invoked by the knowledge base at different times to evaluate the model and provide answers to value-of-information and value-of-control questions. The influence diagram was linked to a Microsoft Excel spreadsheet model containing financial formulas. Facilities in *Axotl* allowed Excel to be launched and the spreadsheet computations made automatically during influence diagram solution.

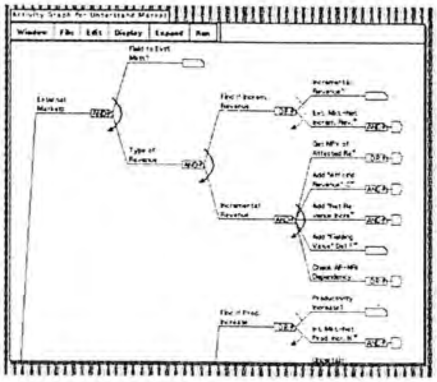

Figure 8. A portion of the R&D project selection activity graph.

An activity graph is a representation of the consultation process as a hierarchy of goals and activities (Holtzman & Russo, 1988). There are two kinds of nodes: goals and activities. The topmost goal in the hierarchy represents the successful completion of a consultation; any number of subgoals may be added. At the leaves of the graph, activities represent procedures that may be executed in support of goals during the course of a consultation (e.g., "assess probability of technical success," "calculate value-of-information on market size"). Successful completion of a consultation requires that a sufficient set of goals be satisfied, either through the execution of the supporting activities or through being explicitly declared satisfied by the individual.

The activity graph facility includes components for viewing and editing goals and activities graphically. Activity graphs can call other activity graphs. During a consultation, an agenda is constructed from a "cut set" through the activity graph. The activities are executed one by one until the failure of an activity or some other change in conditions necessitates a modification of the agenda. The activity graph and agenda are invisible to the decision maker, who sees only the consultation interface.

A knowledge base tool called the heuristic advisor selects and modifies activity graphs during consultations. The knowledge base for the heuristic advisor is currently represented as rules and facts within an MRS-like inference engine (Russell, 1985), implemented in Smalltalk-80.

2.2. Knowledge Acquisition Tools for *Axotl*

Knowledge acquisition tools are being implemented in *DDUCKS* to reduce development time and effort and enhance the quality of knowledge bases. These tools are organized around three different roles for content knowledge and two for process knowledge. Each role has one or more forms of representation associated with it; some have interviewing techniques or analysis tools as well.

2.2.1. Content Knowledge

Elements of the decision basis can be separated into three major components: alternatives, information, and preferences (Figure 9). Possibility tables, grids, and graphs are the major mediating representations used to facilitate knowledge acquisition.

Figure 9. Three roles for knowledge in the content knowledge base with associated forms of representation.

POSSIBILITIES	COMPONENTS								
	Spreadsheet	Home-Brew Prolog	Document'n of Code	Probability Encoder	DA Knowledge Base	Journal and Transcript	Dialog Boxes	UNDO Facility	Tutorial and User Manual
Tight Resources	None 0	Use XOP only 0	No add'l document'n 0	Leave as is 0	None 0	None 0	Sequential menus only 0	None 0	None 0
Base Case 2	Modify Modify existing spreadsheet 5	Straight port of IE w/o XOP emulation 2	Cursory update of PD document 3	Complete wheels and bar charts 1	Single, skeletal KB 5	Journal only 1	Ad hoc dialog boxes (no generality) 2	Support only a few commands [illegible]	Terse, thorough primer 3
Flashy Demo	Enhance existing spreadsheet [illegible]	IE with XOP syntax emulation 3	Throrough update of PD document 5	Add cumulative display 2	Single, saleable KB [illegible]	Basic transcript only [illegible]	Ad hoc but flexible dialog boxes 3	Support most commands 3	Verbose primer 5
All-out	Include some fancy features 5	IE with XOP & optimization 5		Add discretization 4	Two saleable KB's 15	Journal and fancy transcript 5	Generic dialog box facility 7	Multiple UNDO's up to a definable threshold 5	Full-fledged manual with tutorial [illegible]
	Comm'l qual spreadsheet 10				Fancy, useable KB 35				

Figure 10. A possibility table defining *Axotl* development alternatives.

Alternatives. *DDUCKS* includes *Canard*[1], a knowledge acquisition tool based on possibility tables that can be used to generate and structure complex alternatives (Shema, Bradshaw, Boose, & Covington, 1990). Links are maintained between the tables and decision nodes that are part of the content knowledge base. The possibility table representation is based on the manually developed strategy tables (McNamee & Celona, 1987) and morphological charts (Zwicky, 1969) that have been used by decision analysts and designers for many years. *Canard* automates this representation and extends its logic and structure to allow knowledge-based inference and the representation of more complex problems (e.g., hierarchical tables, explicit representation of constraints).

At one point, we used a manually developed possibility table similar to the one in Figure 10 to help us define alternatives for *Axotl* system development. Alternatives are shown in the leftmost column. Other columns represent important components of the system and various options for development within each one. The path of outlined boxes traced through the columns defines the "base case" alternative. Time and budget requirements and constraints could also be associated with variables in the table. Within *Canard* these constraints are used to "gray out" infeasible options during the definition of a path.

A major concern is helping people better explore the space of alternatives. Cognitive scientists have long known that humans typically retrieve only a small fraction of available alternatives when generating hypotheses. People tend to anchor on initial guesses, giving insufficient regard to subsequent data. For a variety of reasons, people may not be able to visualize whole classes of possibilities. Although it would be impossible in practice to guarantee that all relevant alternatives were indeed identified, *Canard* can help people consider a richer set of alternatives. An iterative search procedure that proposes new alternatives based on permutations of the constraint space assists in generation of alternatives. Through an analogous procedure, the system can hypothesize new constraints based on examples of previously defined alternatives.

Preferences. Knowledge acquisition tools can be used to help people determine important dimensions of value associated with the alternatives. Within *Aquinas*, a number of useful representation, interviewing, and analysis techniques have been developed. Many of these techniques are based on the research of George Kelly (1955), a clinical psychologist who emphasized the foundational role of distinctions (*personal constructs*) underlying the processes of perception and reasoning. For example, using a *triadic elicitation* interviewing technique, *Aquinas* would ask

[1] A canard is an airplane that sports a tail in front rather than in back—a symbol of our "backward" analytical approach to synthesis.

people to define preferences by considering groups of alternatives presented three at a time: "Think of an important consequence that two of A1, A2, and A3 share, but that the other does not. What is that characteristic?" After giving B as that characteristic, the person would be asked about A2, A3, and A4 and come up with a second characteristic C, and so forth.

One way of representing this information is through a *repertory grid* (Figure 11). A grid is a matrix with elements (i.e., alternatives or outcomes) ranged along the bottom and constructs (i.e., dimensions of similarity and difference between elements), defined by extension, as a horizontal row of point values (or probability distributions on those values) within the matrix. Although the grid and network representations in Figure 11 are logically equivalent, we find both of them useful and complementary as problem clarifiers from a human factors point of view (Jones, 1981). The grid presentation allows the person to see patterns of similarity and difference that would otherwise be difficult to grasp. Analysis techniques in *Aquinas* (e.g., similarity analysis, cluster analysis) exploit these patterns to help users discriminate more carefully among similar concepts as part of model refinement. Implication analysis helps users discover important dependencies between constructs. A more complete discussion of the relationship of personal construct and decision analytic methods is given in Bradshaw and Boose (1990).

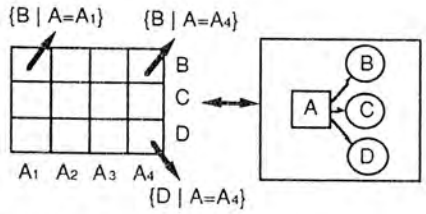

Figure 11 A preference grid and its corresponding influence diagram representation.

Information. Many of the same representation, interviewing, and analysis techniques mentioned above can be used to develop information models for the knowledge base.[1] Interviewing techniques (e.g., laddering) develop the structure of the networks by a recursive expansion procedure that terminates when all leaf nodes are directly observable. Heckerman has developed a related method called *similarity networks*, which uses techniques similar to those used in personal construct methodology to identify and display relationships indicating constraints on conditional independence relationships (Horvitz, Breese, & Henrion, 1988). Figures 12 and 13 show equivalent information grid and relevance diagram representations. When the structure of the dependencies between variables is simple, the transformation between grids and influence diagrams is straightforward. For complex dependencies, some of the methods described by Howard (1988) in his paper on *knowledge maps* may be useful. These methods allow complex probabilistic assessments to be broken down into a series of simpler ones by manipulating the conditioning relationships between chance nodes. We are sponsoring a project with Seattle University to develop procedures to make the use of disjoint and redundant knowledge maps practical in *Axotl*.

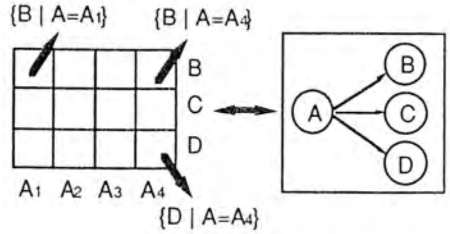

Figure 12. A simple information grid and its corresponding influence diagram representation.

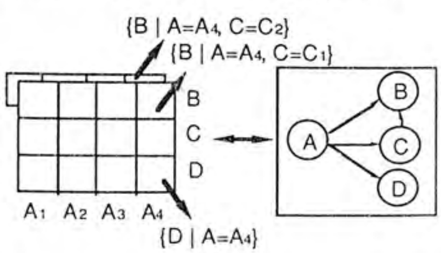

Figure 13. A more complex information grid and its corresponding influence diagram representation.

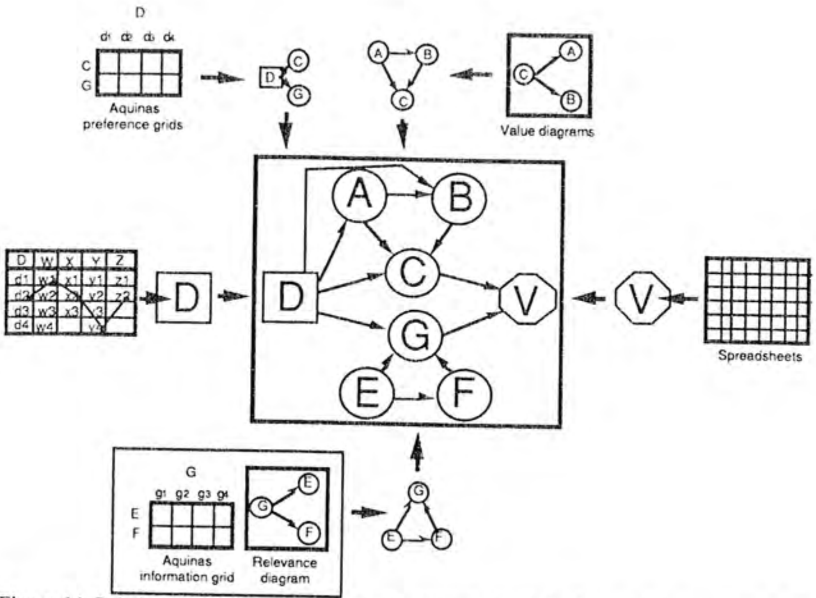

Figure 14. Relationships between elements in the content knowledge base and the structure of an influence diagram.

Figure 14 illustrates some of the relationships possible between elements in the content knowledge base and the structure of a particular influence diagram. Note that the direction of conditioning between chance nodes during the knowledge acquisition process is often the reverse of the direction that is required when the influence diagram is constructed.

2.2.2. Process Knowledge

We distinguish between two major types of process knowledge: plans and plan selection knowledge (Figure 15).

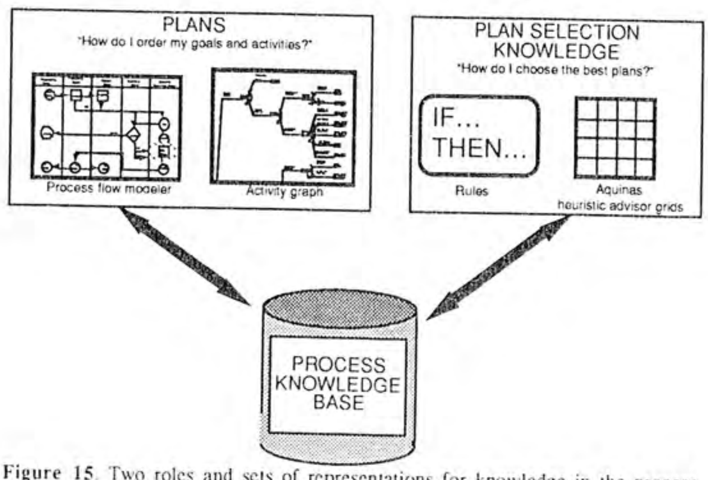

Figure 15. Two roles and sets of representations for knowledge in the process knowledge base.

Plans. Despite many advantages as a representational device, activity graphs have proved to be unworkable as a knowledge acquisition interface. For this reason, we are building an alternative means of activity graph entry based on hierarchical process flow charts (Bradshaw, Kipersztok, Nguyen, & Holm, 1990). Tasks described within process flows are automatically transformed into executable activity graphs (Figure 16). During a consultation, this activity graph is further compiled into an agenda, as described in section 2.1.2 above[1]. Analysis and simulation tools provide verification and validation of the process model.

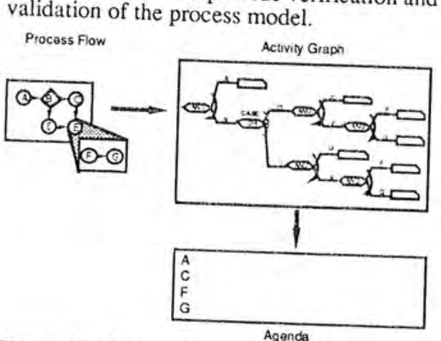

Figure 16. Process flows can be compiled into activity graphs[2].

Plan selection knowledge. Recently, we have begun to design knowledge acquisition tools for this type of knowledge. We have created graphical editors for world trees, rules, queries, and attributes that have simplified the construction of rules in knowledge bases. In the future, links to *Aquinas* will facilitate the construction of knowledge bases for the heuristic advisor. The heuristic advisor is a knowledge base tool that evaluates activities with respect to criteria such as completeness, balance, precision, and cost. Based on this evaluation, the heuristic advisor modifies the activity graph to re-order activities dynamically during a consultation. A related effort has dealt specifically with reasoning about computational resource tradeoffs (Horvitz, 1989). In contrast to our heuristic approach, Horvitz has developed these ideas within a formal framework.

3. DISCUSSION

The heart of our approach to making knowledge acquisition simpler for intelligent decision systems is "divide and conquer". Having experienced the development of many knowledge-based system prototypes using automated knowledge acquisition tools, we are convinced that such a strategy will permit experts to do much of their own knowledge engineering, without requiring a great deal of specialized training (Boose, Bradshaw, Kitto, & Shema, 1990). To the experienced decision analyst or knowledge engineer, the development of these tools may seem completely unnecessary; there is little question that such a person would find it more efficient to work directly with influence diagrams and activity graphs than with the knowledge acquisition tools we are defining. In addition, one could argue that by making it too easy on the naive users of a system, putting everything in their terms without requiring them to come to grips directly with the underlying methodology, we are promoting a "black box" mentality that makes it impossible for them to step in when the system breaks down or encounters a problem it can't solve.

There is no easy answer to these objections; there seems to be a real and inevitable tradeoff between the "acquirability" and the expressive power of knowledge representations (Gruber, 1988). Figure 17 shows this tradeoff as a dark curved line. The most powerful means of getting an idea into a computer is programming; unfortunately, even with advances in software engineering, programming remains the most difficult form for nonspecialists to express their knowledge in. On the other hand, form-filling interfaces that correspond to the way a user normally enters information on paper are very easy to learn but are very rigid and limit the applicability of the tools to the very specific problems that the system designer has foreseen. Knowledge acquisition tools do not eliminate the competition between

[1] In addition to their use in controlling a consultation, these tools are being applied to documentation and improvement of general business processes.

[2] This figure is somewhat simplified for illustrative purposes. In reality, a separate activity graph would be compiled for the main process flow and the subprocess (F->G). This subgraph would be called by the main graph dynamically during execution.

acquirability and expressive power, but they can act as a kind of magnet to help pull the curve out (light gray line). Through the application of automated techniques, acquirable interfaces can become more powerful, and powerful interfaces can be more easily learned and used. *Aquinas* and *Axotl* make the power of influence diagrams accessible to a wider range of people.

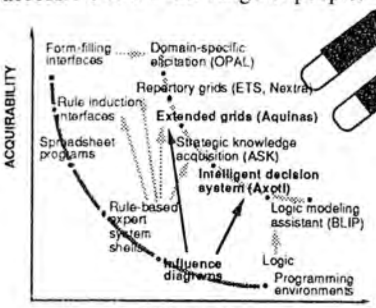

Figure 17. Automated knowledge acquisition tools can help make acquirable interfaces more powerful, and powerful interfaces more easily learned and used (figure adapted from Gruber, 1988).

Because new users will always prefer acquirable interfaces and experienced users will prefer powerful ones, the best strategy to adopt when adding new layers of mediating representation between the new user and the underlying problem-solving model is to leave the hooks to the lower level representations intact. Designers of such systems should also find ways to encourage the less experienced users to keep learning and move on to more expert modes of interaction as they become more familiar and comfortable with the system. Experienced users, on the other hand, should be able to go directly to the heart of the system without interference from a "friendly" interface. This is the approach taken in the nested "Russian doll" interface of HyperCard, where users can graduate from browsing to painting to authoring to scripting at their own rate while performing useful work with the system at each level. This is also the consistent with the "glass box" idea advocated by Wenger and Brown at IRL: to design software that promotes understanding of and access to the inner workings of the system (Feinstein, 1989).

Our conjecture is that high-level representations and interviewing and analysis tools will be helpful to experienced as well as naive users. This is a question that we hope to address as the tools are evaluated in real-world domains.

We are cautiously optimistic about the future of *DDUCKS*. Although current and future applications will no doubt continue to bring new challenges and difficult problems to solve, the tools embody many years of experience in decision analysis automation and knowledge-based system development. Furthermore, the system derives power from its integration of elements that we feel will be the building blocks of future systems for automated support of complex decisions: a decision analysis workbench for sound reasoning under uncertainty and resolution of preference issues; a knowledge-based system for provision of help in building formal decision models and to promote the re-use of domain knowledge; and a collection of knowledge acquisition tools tailored to the domain that allow such systems to be efficiently modeled, more easily maintained, and thoroughly tested.

ACKNOWLEDGEMENTS

We express our appreciation to Kit Bakke, Miroslav Benda, Kathleen Bradshaw, Beverly Clark, Karolyn Covington, Mike Freeman, Peter Friedland, Pete Holm, Oscar and Sharon Kipersztok, Cathy Kitto, Sandra Marcus, Art Nagai, Thom Nguyen, Guy Olney, Doug Schuler, Kish Sharma, David Shema, Greg Swietek, Lisle Tinglof-Boose, and Bruce Wilson for their contributions and support. Thanks also to reviewers who provided many helpful suggestions. Samuel Holtzman, Ronald Howard, Keh Shiou Leu, James Matheson, Léonard Bertrand, and James Smith of Strategic Decisions Group were co-designers and co-developers of the original *Axotl* prototype. This work has also benefitted from ongoing discussions with Brian Gaines, Max Henrion, and Mildred Shaw. *Aquinas*, *Axotl*, *Canard*, and *DDUCKS* are under development at the Boeing Advanced Technology Center. The original development of *Canard* was supported by NASA contract NAS2-12108.

REFERENCES

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

BOOSE, J.H. (1990). A survey of knowledge acquisition techniques and tools. *Knowledge Acquisition Journal,* in 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, 3-28. Also in J. Boose and B. Gaines (Eds.), *Knowledge Acquisition Tools for Expert Systems.* London: Academic Press, pp. 39-64.

BOOSE, J. H., BRADSHAW, J.M., KITTO, C.M., & SHEMA, D.B. (1990). From ETS to Aquinas: Six years of knowledge acquisition tool development. *Knowledge Acquisition Journal,* in press.

BOOSE, J. H., BRADSHAW, J.M., SHEMA, D.B., COVINGTON, S.P. (1989). Design Knowledge Capture for a Corporate Memory Facility, *IJCAI-89 Workshop on Knowledge Acquisition: Practical Tools and Techniques,* Detroit, August, 1989; *IJCAI-89 Workshop on Knowledge Acquisition,* Detroit, August, pp. 5-6.

BOOSE, J.H., SHEMA, D.S., & BRADSHAW, J.M. (1989). Recent progress in Aquinas: A knowledge acquisition workbench. *Knowledge Acquisition Journal,* 1(2): 185-214.

BRADSHAW, J. M. & BOOSE, J.H. (1989). Knowledge acquisition as CASE for knowledge-based systems. *Presentation at the Third International Workshop on Computer-Aided Software Engineering (CASE-89),* London, England, July 17-21, 1989.

BRADSHAW, J. M. & BOOSE, J.H. (1990a). Decision analysis techniques for knowledge acquisition: Combining information and preferences using *Aquinas* and *Axotl. International Journal of Man-Machine Studies,* 32(2): 121-186. Also in J.H. Boose and B.R. Gaines (Eds.), *Progress in Knowledge Acquisition for Knowledge-Based Systems..* London: Academic Press.

BRADSHAW, J.M. & BOOSE, J.H. (1990b). What is a knowledge acquisition tool? In K. Ford & J. Bradshaw (Eds.), special knowledge acquisition issue of the *Journal of Intelligent Systems,* 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: The application of decision analysis methods in *Axotl* and personal construct methods in *Aquinas* to design problems. *Third AAAI Knowledge Acquisition for Knowledge-based Systems Workshop,* Banff, Canada, November, 1988.

BRADSHAW, J.M., COVINGTON, S.P. & RUSSO, P.J. (1989). PIE: An application of *Axotl* to project selection decisions. Proceedings of the Boeing AI Conference, April.

BRADSHAW, J.M., COVINGTON, S., RUSSO, P., & BOOSE, J.H. (1989). KW'ACQ'ing with *DDUCKS:* Knowledge acquisition techniques for decision analysis using *Axotl* and *Aquinas. Proceedings of the Fourth AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff, Canada, October. *Knowledge Acquisition Journal,* in press.

BRADSHAW, J. M. & HOLTZMAN, S. (1987). Artificial intelligence and decision analysis. Presentation made to the Nineteenth Annual Meeting of the Society for Information Management, Seattle: October 17.

BRADSHAW, J.M., KIPERSZTOK, O., NGUYEN, T. & HOLM, P. (1990). Acquiring knowledge for process management using *eQuality.* Manuscript in preparation.

BREESE, J.S. (1987). Knowledge representation and inference in intelligent decision systems. Doctoral dissertation, Department of Engineering-Economic Systems, Stanford University. Reprinted as Research Report 2, Palo Alto, California: Rockwell International Science Center.

ESHELMAN, L. (1988). MOLE: A knowledge acquisition tool for cover-and-differentiate systems. In S. Marcus (ed.), *Automating Knowledge Acquisition for Expert Systems*. Boston, Massachusetts: Kluwer Academic Publishers.

FEINSTEIN, D. (1989). Glass boxes: Lifting the veil on information. *Benchmark*, Spring, 12-15.

GAINES, B.R. (1988). Software engineering for knowledge-based systems. *Proceedings of the Second International Workshop on Computer-Aided Software Engineering (CASE-88).*

GRUBER, T.R. (1988). The acquisition of strategic knowledge. Unpublished Ph.D. dissertation. Department of Computer and Information Science: University of Massachusetts.

HOLTZMAN, S.H. & RUSSO, P.J. (1988). Activity graphs. Unpublished PIE project internal technical memo.

HOLTZMAN, S. H. (1989). *Intelligent Decision Systems*. Reading, Massachusetts: Addison-Wesley.

HOLTZMAN, S. & SEIVER, A. (1988). Decision analysis: A framework for critical care decision assistance. *International Journal of Clinical Monitoring and Computing*, in press.

HORVITZ, E.J. (1989). Reasoning about beliefs and actions under computational resource constraints. In L. Kanal, T. S. Levitt, & J. F. Lemmer (eds.), *Uncertainty in Artificial Intelligence*, volume 3. Amsterdam: North-Holland.

HORVITZ, E. J., BREESE, J. S., & HENRION, M. (1988). Decision theory in expert systems and artificial intelligence. *Journal of Approximate Reasoning*, in press.

HOWARD, R. A. (1988). Knowledge maps. *Management Science*, in press.

HOWARD, R. A. & MATHESON, J. E. (1981). Influence Diagrams. Reprinted in Howard & Matheson, 1984.

HOWARD, R. A. & MATHESON, J. E. (Eds.) (1984). *Readings on the Principles and Applications of Decision Analysis*. Menlo Park, California: Strategic Decisions Group.

JONES, J.C. (1981). *Design Methods*. New York: John Wiley & Sons.

KARBACH, W., LINSTER, M. & VOSS, A. (1990). A confrontation of models of problem solving. In K. Ford & J. Bradshaw (Eds.), special knowledge acquisition issue of the *Journal of Intelligent Systems*, in press.

KEENEY, R. L. (1986). *Value-Driven Expert Systems For Decision Support*. Los Angeles: Systems Science Department, University of Southern California.

KELLY, G. A. (1955). *The Psychology of Personal Constructs*. 2 volumes. New York: Norton.

KLINKER, G. (1989). A framework for knowledge acquisition. *Proceedings of the Third Annual European Knowledge Acquisition Workshop*, Paris, France, July.

MARCUS, S. (1988). SALT: A knowledge acquisition language for propose-and-revise systems. In S. Marcus (ed.), *Automating Knowledge Acquisition for Expert Systems*. Boston, Massachusetts: Kluwer Academic Publishers.

MATHESON, D. (1988). Knowledge acquisition results on template development. Unpublished PIE project internal technical document.

MATHESON, J.E. (1988). Using influence diagrams to value information and control. *Proceedings of the Conference on Influence Diagrams for Decision Analysis, Inference, and Prediction*, Berkeley, California, University of California at Berkeley, May 9-11.

MCNAMEE, P. & CELONA, J. (1987). *Decision Analysis for the Professional—With Supertree*. Redwood City, California: Scientific Press.

MEYER, M.A., BOOKER, J.M. & BRADSHAW, J.M. (1990). A flexible six-step program for defining and handling bias in knowledge acquisition. Manuscript submitted to the Fourth European Workshop on Knowledge Acquisition for Knowledge-Based Systems, Amsterdam, July.

MOORE, E. A. & AGOGINO, A. M. (1987). INFORM: An architecture for expert-directed knowledge acquisition. *International Journal of Man-Machine Studies*, 26, 213-230.

MUSEN, M.A. (1988). *Generation of model-based knowledge acquisition tools for clinical-trial advice systems.* Doctoral dissertation, Computer Science Department, Stanford University. Available as KSL-88-06 and STAN-CS-88-1194.

NEAPOLITAN, R.E. (1990). *Probabilistic Reasoning in Expert Systems: Theory and Algorithms.* New York: John Wiley.

OLMSTED, S. M. (1982). SUPERTREE - decision tree processing program. Menlo Park, California: Strategic Decisions Group.

RUSSELL, S. (1985). The compleat guide to MRS. Stanford Knowledge Systems Laboratory, Report No. KSL-85-12. Stanford, California: Stanford University Computer Science Department.

SHACHTER, R. D. (1986a). Evaluating influence diagrams. *Operations Research*, 34: 871-882.

SHACHTER, R. D. (1986b). DAVID: Influence diagram processing environment for the Macintosh. In *Proceedings of the RCA/AAAI Workshop on Uncertainty and Probability in Artificial Intelligence.*

SHAW, M.L.G. & GAINES, B.R. (1988). A methodology for recognizing conflict, correspondence, consensus, and contrast in a knowledge acquisition system. *Proceedings of the Third AAAI Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Banff, Canada, November.

SHEMA, D. B., BRADSHAW, J.M., BOOSE, J.H., & COVINGTON, S.P. (1990). Design knowledge capture and alternative generation using possibility tables in *Canard. Knowledge Acquisition Journal*, in press.

SIVARD, C., ZWEBEN, M., CANNON, D., LAKIN, F. & LEIFER, L. (1989). Conservation of design knowledge. *Proceedings of the 27th Aerospace Sciences Meeting of the American Institute of Aeronautics and Astronautics*, Reno, Nevada, January 9-12.

SMITH, J. E. (1988). On the representation and evaluation of influence diagrams. Unpublished PIE project internal technical document.

SULLIVAN, K. M. & SHULMAN, H.M. (1989). Cyclosporine and long-term follow-up care of the marrow transplant recipient: Clinical histories of chronic graft-versus-host disease, obliterative bronchiolitis and graft-versus-leukemia effect. *Transplant Proc.*, in press.

SZOLOVITZ, P. & PAUKER, S.G. (1978). Categorical and probabilistic reasoning in medical diagnosis. *Artificial Intelligence*, 11:115-144.

WELLMAN, M. (1986). Reasoning about assumptions underlying mathematical models. In J. S. Kowalik (ed.), *Coupling Symbolic and Numeric Computing.* Amsterdam: Elsevier.

ZWICKY. F. (1969). *Discovery, Invention, Research through the Morphological Approach.* New York: Macmillan.