Knowledge Acquisition (1993) **5**, 405–448

Knowledge acquisition techniques for group decision support

JOHN H. BOOSE,† JEFFREY M. BRADSHAW, JOSEPH L. KOSZAREK AND DAVID B. SHEMA

Boeing Computer Science Technology Organization, Boeing Computer Services, 7L-64, PO Box 24346, Seattle, WA 98124, USA

(Received and accepted 16 July 1993)

Existing group decision support systems used in meeting rooms can help teams reach decisions quickly and efficiently. However, the decision models used by these systems are inadequate for many types of problems. This paper describes our laboratory's experience with knowledge acquisition systems and decision support tools. Our studies led us to develop a comprehensive decision model for group decision support systems. This decision model combines current brainstorming-oriented methods, structured text argumentation (using the gIBIS model), repertory grids, possibility tables (morphological charts) and influence diagrams from decision analysis. Each component addresses weaknesses in current group decision support systems. We are assembling these group decision support components together into a group decision workbench.

1. Enhancing group productivity with decision support

To better solve complex problems and foster interdisciplinary work, The Boeing Company has made major organizational changes. For example, the design and fabrication of a commerical airplane requires hundreds of disciplines to cooperatively work together to satisfy customer expectations. These activities require group decisions which integrate sources of information such as customer requirements, technological advancements, and manufacturing methods. Decision alternatives must be evaluated against assumptions, objectives, and constraints. For the next generation airplane, this will require the development of new collaborative information support environments.

To promote the development of a collaborative support environment, we are conducting experiments combining group decision support systems and knowledge acquisition techniques (see Figure 1).

1.1. ADVANTAGES OF GROUP DECISION SUPPORT SYSTEMS

The need for electronic support of meetings is clear. In most organizations people spend an average of 40% of their time in meetings, which are often frustrating, due to the many barriers to productivity inherent in their structure (see Table 1).

Group decision support systems (GDSSs) address these problems. Tools such as GroupSystems (Nunamaker Jr., Applegate & Konsynski, 1988; Ventana, 1990; Daniels, Dennis, Hayes, Nunamaker Jr. & Valacich, 1991), SAMM (Dickson, 1991;

† Present address: Fred Hutchinson Cancer Research Center, 1124 Columbia Street, FB-600 Seattle, WA 98104, USA.

1042-8143/93/040405 + 43$08.00/0

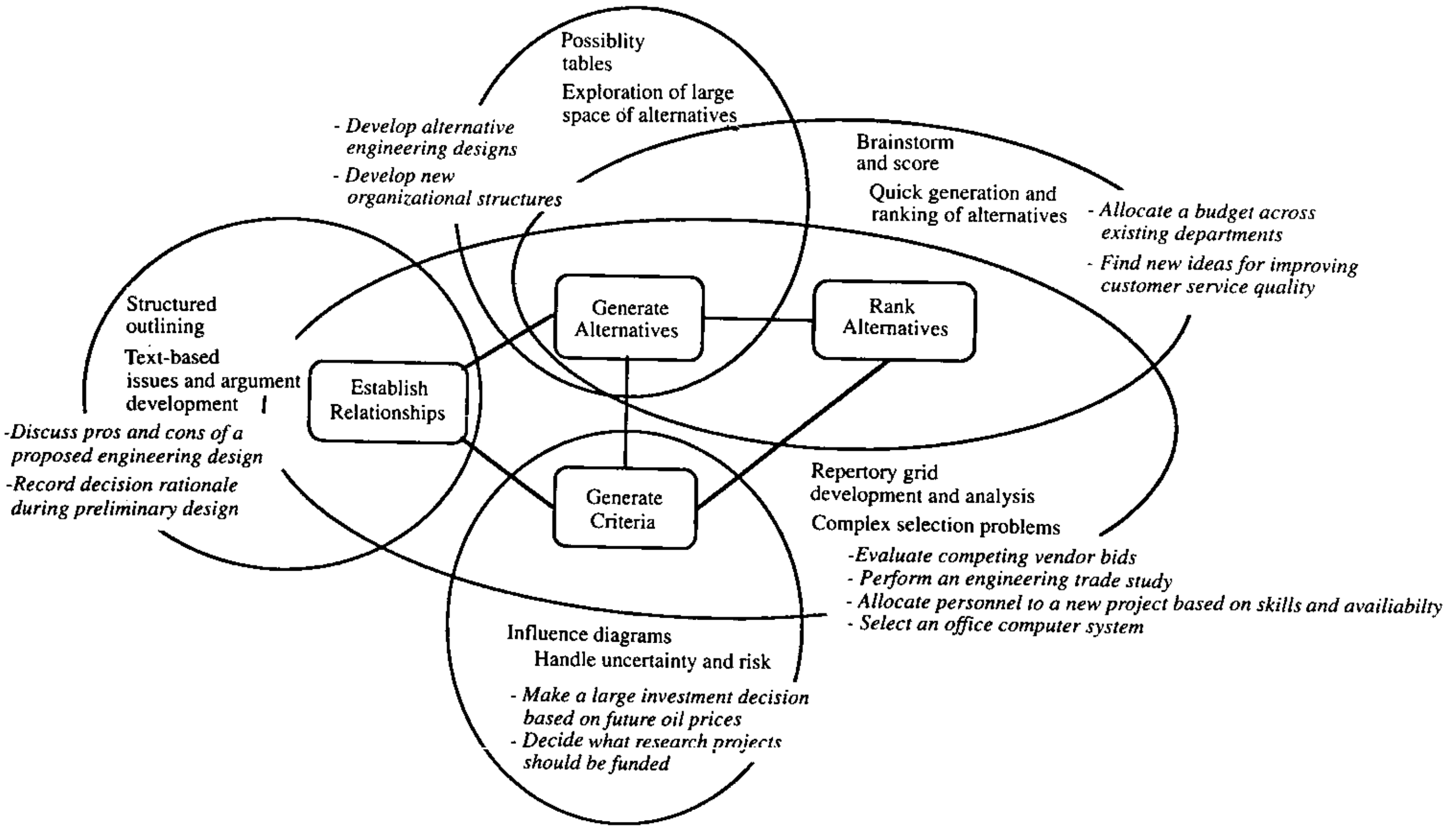

FIGURE 1. Our decision support model combines components of current group decision support systems and knowledge acquisition methods.

TABLE 1

Barriers to productivity often frustrate meeting participants (Nunamaker Jr., Vogel, Heminger, Martz, Grohowski & McGoff, 1989; *Nunamaker Jr.,* 1991)

It is easy to stray from the agenda
Less assertive individuals may have little opportunity to offer important ideas
Many meetings end with no sense of closure or accomplishment
The group must partition available time among members
Members can't contribute comments as they occur. They may forget or suppress them later in the meeting as they seem less original, relevant, or important
Fewer comments are made because members concentrate on remembering comments (rather than thinking of new ones) until they can contribute them
Members must listen to others speak and cannot pause to think and generate new comments
Members lack focus on communication, missing or forgetting the contributions of others
Members are reluctant to criticize the comments of others owing to politeness or fear of reprisals
Fear of negative evaluations cause members to withold ideas and comments
Members rely on others to accomplish goals, owing to cognitive loafing, the need to compete for air time, or because they perceive their input to be unneeded
Discussion moves along one train of thought without deviating. Members don't make comments that are not directly related to the current discussion
Non-task discussion reduces task performance, although some socializing is usually necessary for effective functioning
Some group members exercise undue influence or monopolize the group's time in an unproductive way
Members receive information faster than it can be processed
Often a strategy is missing for reaching a decision
Most of the meeting is undocumented
Scheduling meetings is difficult
Travel for meetings is expensive
Unproductive meetings frustrate teams

Tan, Wei & Raman, 1991), TeamFocus (Nunamaker Jr., Vogel, Heminger, Martz, Grohowski & McGoff, 1989) and VisionQuest (Collaboration Technologies Company, 1991) use the processes of brainstorming, idea organization, and scoring and ranking. These tools support many types of application problems (see Table 2).

Studies show that groups using GDSSs can solve more problems with greater efficiency (Kramer & King, 1988). Experiments suggest that computer support can improve the *quality* of the decisions (Gallupe, DeSanctis & Dickson, 1988; Jarvenpaa, Rao & Huber, 1988; Nunamaker Jr., Applegate & Konsynski, 1988). Other benefits of electronic meeting room decision support systems appear in Table 3.

1.2. PROBLEMS WITH GDSS DECISION MODELS

There are many areas of potential improvement for current GDSSs (such as the user interface, process modeling support, portability, cost). This paper focuses on their

TABLE 2

Group decision support systems enable productivity gains in many application areas (Dennis, George, Jessup, Nunamaker Jr. & Vogel, 1988; *Nunamaker Jr., Weber, Smith & Chen,* 1988; *Daniels, Dennis, Hayes, Nunamaker Jr. & Valacich,* 1991; *Sycara & Roboam,* 1991)

Budget/resource allocation
Concurrent engineering
Crisis planning
Development of marketing image
Exploration of business challenges
Idea generation
Joint application design (JAD)
Market planning
Negotiation
Policy development
Requirements gathering and definition
Resolution of sensitive issues
Strategic planning
Systems analysis and design
Team building

decision modeling ability in the context of same time, same place meetings. Many of the methods discussed below could also be applied to any time, any place meetings.

Sometimes the decision models provided by current systems break down when applied to more complex problems. Examine the following scenario:

Department heads meet with their manager to help allocate money among projects for the next year:

1. First, electronic brainstorming helps produce a list of important project related issues.
2. Each member separately allocates a fixed amount of resources among the projects.
3. Members further discuss issues and the system displays the group consensus and spread of opinion.
4. The manager makes minor adjustments to the consensus and implements the budget.

This *brainstorm-and-score* decision model can work well when there are few interacting constraints among alternatives and when the space of alternatives is fixed in size. But this simple list-prioritization scheme cannot represent the following kinds of decision information:

Complex criteria. What reasons are there for picking Project A over Project B? How important are the reasons? How does the group rate each project on each reason? What aspects of criteria do the members prefer? Do certain reasons only

TABLE 3

GDSSs can help facilitate group processes (Dennis, George, Jessup, Nunamaker Jr. & Vogel, 1988; *Nunamaker Jr., Vogel, Heminger, Martz, Grohowski & McGoff,* 1989; *Dennis, Valacich & Nunamaker Jr.,* 1991; *Nunamaker, Jr.,* 1991)

General group process advantages:
- A group as a whole has more information than any one member
- A member uses information in a way that the original holder did not, because that member has different information or skills
- Groups are better at catching errors than are the individuals who proposed ideas
- Working as part of a group may stimulate and encourage individuals to do better
- Members may learn from and imitate more skilled members to improve performance

Advantages of electronic support:
- Entries made by individuals are anonymous
- The meeting focus is on content and task, not personalities
- Simultaneous entry enables equal participation from all members
- Entries are parallel and simultaneous. One or two individuals do not dominate a meeting. Everyone generates and evaluates ideas
- Meetings produce electronic records
- Everyone generates more data in less time
- Meeting times shorten
- Fewer meetings are needed
- IBM on average has reduced calendar time 90% and people hours 60% for many types of meetings
- It is often easier to reach agreement on volatile issues since ideas are depersonalized
- Participants feel greater sense of ownership in outcomes
- The environment imposes a process structure; the structure helps to focus meetings
- The environment fosters innovation and creativity

apply to certain projects?

Minimums, maximums, ranges. Project C needs a minimum allocation of $200 000. Project D needs a range of $150 000–$450 000. Our budget limit is $2 000 000.

Restructuring complex alternatives. Projects E, F and G seem to have overlapping customer requirements and objectives. Is this desirable? Will it lead to duplicated effort or uncommon solutions? Can we restructure projects and their objectives so that we don't duplicate effort but still meet all the requirements?

Exclusivity. If we do Project H, we shouldn't do I.

Enabling conditions. If the full $450 000 were allocated for J, we could purchase critical resources from ACME Products.

Risk management. We have a make or buy decision. We're not sure ACME will supply a critical resource in time for Project K. If they don't, we'll have to build the resource ourselves, and it will cost an additional $200 000. Should we hold this

money in a contingency fund? How long should we wait for ACME to deliver the resource? Project L has a similar problem. Can we share the risk across both projects (how is the year's budget affected if both, neither, or one project will need additional money)? What amount of risk should we tolerate?
Effect of uncertainty on payoffs. Project M's success is less certain than that of Project N but would probably have a bigger pay back if it was successful.
Timing. Project O depends on enabling technology from another department. We shouldn't start the project until it's delivered. How can we use the uncertainty about the delivery date to decide how much to allocate to Project O next year? If it is delivered in March we would spend more money next year than if it were delivered in September, even though the multiple-year cost would remain the same.
Documenting running discussions. The requirements changed since the last budget decisions—what arguments contributed to the selection of one alternative over another? What will be the effect on requirement satisfaction if we use a different alternative?

The brainstorm-and-score model can't handle these types of decision information. For instance, it uses a satisficing method—"Find a way to solve the problem"—not "Make sure you find the best way—or at least a good way—to solve the problem". A more complex model can help when we need assurance that a large portion of the space of alternatives has been considered (for instance, engineering design alternatives for a major system or subsystem) or when there are interacting constraints.

Brainstorm-and-score models lack explicit criteria, or at most include one criterion for scoring, such as "importance" or "value". It is often important to illuminate many of the criteria affecting the decision. Explicit criteria can help point out areas of agreement and disagreement and enable special forms of analysis. We can express preferences for criteria values. We can assess uncertainty about aspects of the decision. We can take into account attitudes toward risk and timing of deliverables. We can measure the value of gathering further information or spending more resources to control aspects of a situation. The model can represent constraints between criteria. The reuse of criteria for future changes and similar problems also helps offset the cost of collecting information about criteria.

Current GDSSs offer powerful environments to help users enumerate ideas during a *divergence* phase of activity. But another problem occurs during the *convergence* phase, when users must reduce and categorize the idea list. This synthesis process can be dissatisfying and painful for group members. Methods discussed below can reduce this frustration by providing an information constraint framework (see Section 3.2.4 on possibility tables).

Section 2 discusses the foundations of our knowledge acquisition approach for representing the missing information in a decision model. In Section 3 we develop a model that incorporates the missing decision elements needed to handle these problems. We use the budget problem and other examples to illustrate features of the model.

2. A knowledge acquisition approach to GDSS design

We take a unique approach to GDSS decision modeling. We base our approach on our laboratory's experience with knowledge acquisition and decision support tools.

Using knowledge acquisition methods could significantly improve the effectiveness of GDSSs by addressing their modeling inadequacies.

Many knowledge acquisition tasks are similar to group decision support system tasks, and many knowledge acquisition tools use decision models (Boose, 1991). In both cases, decision information must be elicited, represented, used to gain insight about a decision, verified, and validated. Decision models in both areas guide the type of information that should be captured. Users iteratively refine decision models as they expand and test them. Eliciting and using knowledge from multiple experts is similar to the problem of gathering information in a group setting.

The field of knowledge acquisition has developed powerful methods that help users directly build, test, and refine models. We focus on representations and analysis methods used by successful knowledge acquisition tools that we and others have built.

Knowledge acquisition for decision support follows these steps:

1. Identify a problem-solving model by examining features of the problem. Section 2.1 describes knowledge acquisition as a *modeling activity*.
2. Define knowledge types and roles. Important roles in a GDSS include information, preferences, and alternatives. Section 2.2 discusses these elements of a decision.
3. Section 2.3 describes the role of *mediating representations* that effectively communicate important problem aspects.
4. Section 2.4 discusses the cost–benefit tradeoff of building more complex decision models.
5. Design or adapt elicitation and analysis techniques. Section 3 describes the components of our decision model.

2.1. KNOWLEDGE ACQUISITION AS MODELING

Recent work in knowledge acquisition for knowledge-based systems has emphasized that the creation of knowledge bases is a constructive *modeling* process, and not simply a matter of expertise transfer of knowledge capture. Clancey (1986) stated that knowledge acquisition is the familiar scientific problem of trying to create a model where, in principal, none existed before (e.g. Clancey, 1984, 1990; Musen, 1988; Wielinga, Akkermans, Schreiber & Balder, 1989; Bradshaw & Boose, 1990; Gruber, 1990; Shaw & Woodward, 1990; Cox, 1991; Ford & Adams-Webber, 1991).

We use knowledge acquisition methods to guide our thinking about the problem of building decision systems. We use related concepts to define problem types and the roles knowledge can play—the elements of a decision. We fill in the model gaps in group decision support systems using successful approaches from our knowledge acquisition experience.

A model-based description of the problem given in a form the user can intuitively understand has many advantages. One important one is that it can serve to mediate communication between users of the system, helping them articulate and understand the broader, higher-level problem context. Section 2.3 on mediating representations discusses some of these advantages.

Knowledge-based system researchers have defined many application problem and problem-solving method taxonomies (Boose, 1992). These models establish and control the sequence of actions required to do a task. For example, *Analysis*

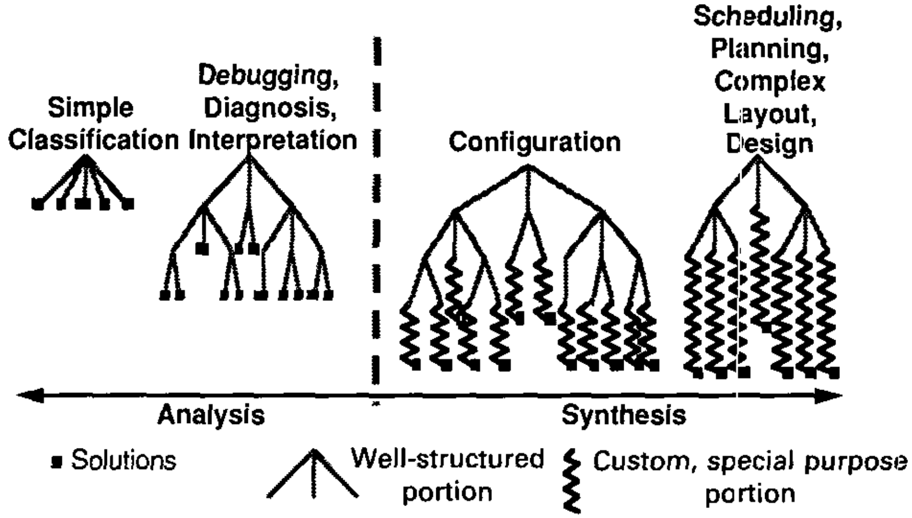

FIGURE 2. Problem classification techniques borrowed from knowledge-based systems help define decision support models.

problems involve identifying sets of objects based on their features. *Synthesis* (generative, or constructive) problems require that a solution be built up from components pieces or subproblem solutions (see Figure 2). Associating a problem with a category helps define the kinds of knowledge, elicitation, analysis and reasoning that we need to reach a decision.

Ideally, decision modeling tools should also support the entire decision life-cycle, from initial conceptualization to finished implementation of a solution. Each phase of the life-cycle, however, poses its own problems and has its own requirements. Many of the problems associated with information acquisition and maintenance stem directly from the inadequacies of representations used at various stages in the development of decision models. Our representations and tools should support a smooth evolution of the model from an easily communicated, unconstrained, conceptual statement of the problem to an unambiguous specification of the decision reasoning system. This suggests a requirement that decision tools accommodate the changes in representation that may accompany successive stages in model construction: from mental models to increasingly refined conceptual models via elicitation and analysis techniques, and eventually, from these highly elaborated models to an operational decision model information base via formalization and implementation procedures (Shaw & Woodward, 1989).

2.2. ELEMENTS OF A DECISION

A complete decision model, containing relevant items of problem-solving knowledge and their interrelationships, constitutes the *decision basis* (Howard & Matheson, 1984). The decision basis combines *information*, *preferences* and *alternatives*. Information consists of the knowledge about the problem. Preferences are factors that determine the desirability of an alternative, such as cost, effectiveness or risk. Alternatives are the possible solutions for the decision. Figure 3 shows these three types and some important subtypes of knowledge.

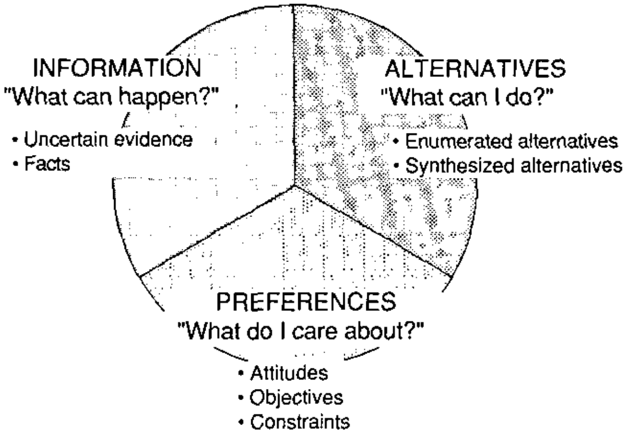

FIGURE 3. Building blocks of a decision include information, alternatives, preferences, and their interelationships.

2.2.1. *Information*

One important part of decision knowledge is *information* about actual or possible circumstances in the world that affect the decision. For example, we might want to know the likelihood of the timely delivery of a critical resource before deciding whether to fund a project. Finding out about a predicted delay may change our decision.

It is useful to think of information as being of two types: *uncertain evidence*, which are statements believed to be true with some probability, and *facts*, which are statements believed with certainty. Facts are a special case of uncertain evidence in which the mapping on to the space of probability distributions happens to be concentrated on a single point. However, by making this distinction, we can often formulate a problem with additional clarity and computational efficiency.

We can have incomplete information about the present and uncertainty about the future. The decision may involve high degrees of risk. These problem affect the decision process.

2.2.2. *Preferences*

Preferences describe the multiple, often competing goals that we value as outcomes of a decision. The strength of our preferences motivates us to make a careful choice among alternatives. If we really didn't prefer project success to failure, or prefer spending less money to spending more money, any effort we put into making a decision would be wasted.

It is useful to distinguish between *direct* and *indirect* preferences. Direct preferences relate to things we value for their own sake. Indirect preferences have no intrinsic value except as they relate to direct ones. For example, when purchasing a car, most people place a direct value on cost. However, fuel economy is usually only indirectly valued because of its contribution to overall cost.

There are three different types of preferences: *attitudes, objectives* and *constraints*:

1. **Attitudes** consist of items such as *time preference* (e.g. the desire to receive

good outcomes sooner rather than later) and *risk attitude* (e.g. the desire to pursue a possibly less profitable policy to avoid risk).
2. **Objectives** relate to the significant positive and negative consequences of the alternatives that a decision maker wishes to maximize or minimize. In decisions with multiple objectives, we must find methods of quantification and joint measurement (i.e. commensuration) to make tradeoffs between them.
3. **Constraints** specify the conditions for maximizing objectives. They define the limits of the space of acceptable outcomes. Constraints may be matters of definition (e.g. "There is $2 million budgeted for the project this year") or of principle (e.g. a moral belief that precludes consideration of alternatives such as corporate spying to speed project development). We can relax many hard constraints. For example, if we determine that a contemplated project cannot be finished within its original time and budget constraints, we may decide to increase the budget and hire additional personnel to meet the deadline.

Explicit modeling of preferences leads to the identification of important criteria, the ability to handle complex tradeoffs, and the ability to reach a difficult consensus:

1. **We can address unique tradeoffs.** The model represents preferences and tradeoffs explicitly so they can be seen by all and modified directly.
2. **We can evaluate the effects of pieces of evidence.** By performing *sensitivity analysis* we can examine whether a particular piece of information in favor of a project alternative will have any real effect on the allocation decision.
3. **We can measure the value of obtaining additional information.** We can ask, "What is the most I should pay to gather intelligence about a competitor's project in this area?"
4. **We can determine the value of controlling an uncertain variable.** To guarantee the timely delivery of a critical resource we could assess the value of acquiring or merging with another company.
5. **We can measure and use risk attitude and time preference.** In some situations, it is worthwhile to model the company or department's attitude toward risk before making a decision. Time-critical situations may also require explicit modeling of the risks and benefits of delaying or hastening the course of a project.
6. **We can express recommendations in value terms.** It may be important for a system to not only recommend the favored budget plan, but also to estimate its total cost or benefits in some meaningful unit of measurement. We can address the quantification and joint measurement of "intangibles" such as company image and quality of worker life.

2.2.3. *Alternatives*

Alternatives are the courses of action that may be recommended, consistent with the decision basis. After hearing an economic report (information) and determining the effect of the report on the success of the projects (preferences) we will choose between projects (alternatives).

Often alternatives are synthesized from constrained components. for example, including certain components of a design may preclude other components. Possibility tables, discussed below, represent complex inter-component constraints and help users synthesize new solutions.

2.3. MEDIATING REPRESENTATIONS

Effective mediating representations are critical to the success of both knowledge acquisition tools and group decision support systems. They are the users' window on the decision model. Many knowledge acquisition tools achieve success in part through the development and adaptation of good mediating representations (see Figure 4).

We use the term mediating representation to "convey the sense of . . . coming to understand through the representation" (Johnson, 1989). The design of a mediating representation should be optimized for human understanding. Effective mediating representations ease participant involvement in a group setting.

Winston (1984) says effective representations make important things explicit and

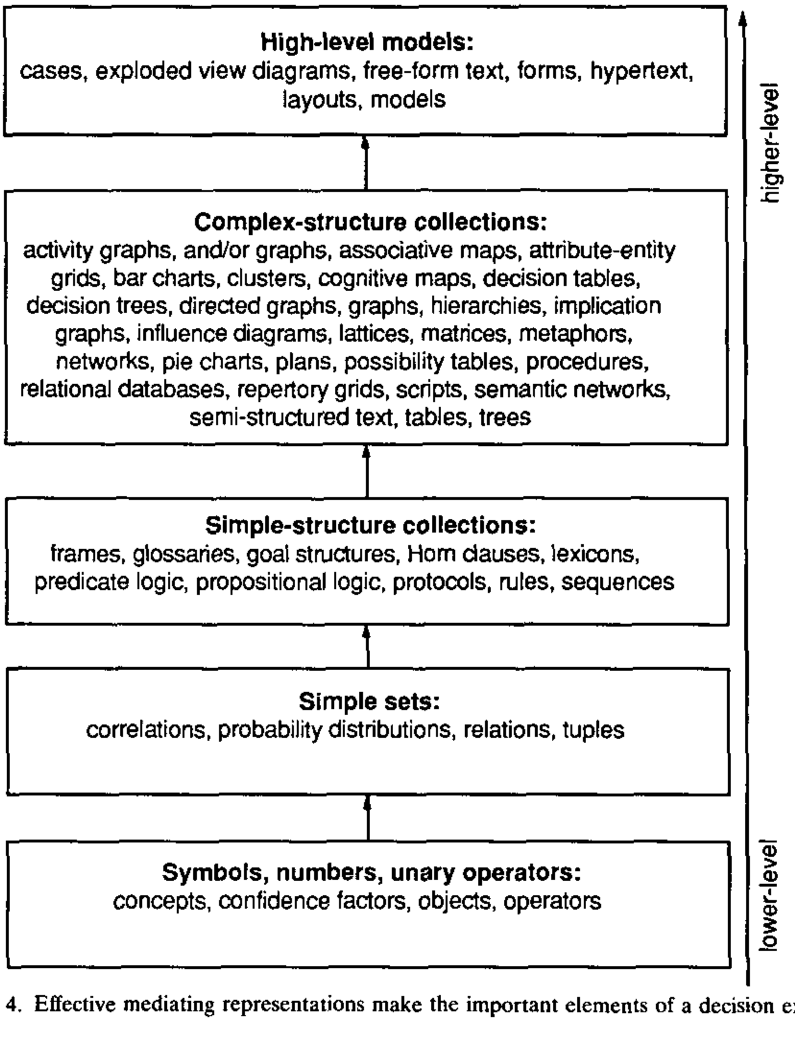

FIGURE 4. Effective mediating representations make the important elements of a decision explicit.

hide unnecessary detail. They expose natural constraints, facilitate computation, and are complete and concise. The choice of representation can have an enormous effect on human problem-solving performance (Larkin & Simon, 1987). As an example, consider that we can represent numbers as Arabic numerals, Roman numerals, or as bits in computer memory. While all of these forms are logically equivalent, they are not the same in a practical sense. It is much more efficient for a computer to multiply numbers represented as bits than as numeric symbols. Similarly, from a human perspective, it is easier to do multiplication with Arabic numerals than with Roman numerals or binary numbers.

A good mediating representation can simplify modeling processes by providing a medium for users to model their valuable but difficult-to-articulate knowledge about an explicit external form. The mutual development of a representation supplementing the exchange of information between participants promotes and enriches communication, leading gradually to a shared understanding of the emerging conceptual model of the domain (Norman, 1988, 1991). Mediating representations enable participants to cooperatively build decision models. Mediating representations may also simplify maintenance and explanation by enabling users to explore the conceptual domain model without resorting to low-level representations (e.g. C code, Lisp, rules, pure text).

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 do some task (Gruber, 1989; Klinker, 1989; Karbach, Linster & Voß, 1990). The problem-solving model defines the type of knowledge applicable within each step, thereby making explicit the different roles that knowledge plays. Once these roles are defined, we design representations and procedures needed for acquiring each type of knowledge. Research on mediating representations has generally attempted either to improve the computational expressiveness of human-efficient representations (e.g. repertory grids, decision trees, hypertext) or to improve the learnability of computationally powerful ones (programming-by-example, fourth-generation languages).

Automated knowledge acquisition tools are beginning to incorporate effective mediating representations. These tools tend to adopt one of two approaches. Either they contain interfaces that bear a close resemblance in appearance and procedure to the original manual task—for example, cancer-therapy protocol forms in OPAL (Musen, 1988) and engineering notebooks in vmacs (Sivard, Zweben, Cannon, Laken & Leifer, 1989)—or they rely on some easily-learned, generic knowledge representation form—for example, object hierarchies and repertory grids in DART (Boose & Bradshaw, 1987; Boose, Shema & Bradshaw, 1990a, 1990b).

2.4. DECISION MODEL COSTS AND BENEFITS—APPLYING JUST ENOUGH EFFORT TO A DECISION

There is a tradeoff between the costs and benefits of building more complex decisions models. Ideally, we should apply just enough effort to solve a problem with the needed level of accuracy. Finding this level is usually an iterative process. We wish to start with simple models and expand them in critical areas. Sensitivity analysis can help identify these areas. A simple budget model that starts with lists of

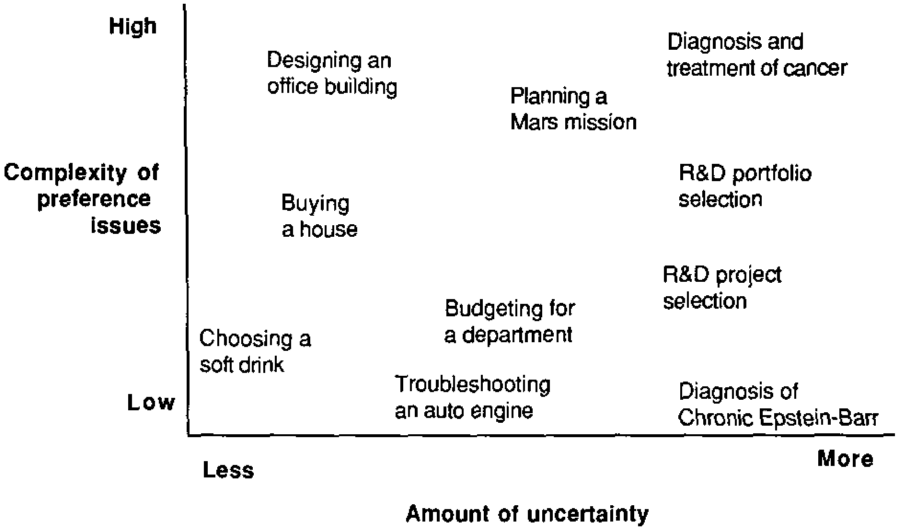

FIGURE 5. Problems with higher uncertainty and complexity require more complex decision models.

projects may expand to include explicit criteria, interacting constraints, and risk management factors.

Decision analysis (Howard & Matheson, 1984), for example, is a rigorous methodology that involves eliciting numbers that represent uncertainty, value and risk. Many decisions do not require this level of rigor. Sometimes the cost of such analysis is not justified by the small size of the problem; sometimes the problem seems so simple that the best alternative is obvious and we can act immediately (Cyert & March, 1963; March, 1978). For personal decisions involving ethical issues, such as whether to consider an abortion, the methodology itself may be largely inappropriate (Levi, 1986).

Even among complex decisions, analysis requirements vary: different problems demand different amounts of emphasis on modeling different kinds of problem-solving knowledge. Complex decision problems require both information and preference modeling for the system to enable insight and effectively formulate recommendations. Grids, for example, enable new forms of analysis in a group setting. These analysis methods can be keys to gaining understanding and insight about a problem.

Figure 5 displays a set of problems as a function of the amount of uncertainty and the complexity of preference issues.

We should determine the complexity of the model, then, by examining the types of information needed to solve the problem, the precision and accuracy needed, the time and resources available and the relative importance of the decision.

3. Building a stronger model for a GDSS

In this section we develop a GDSS decision model that incorporates aspects of many tools and techniques. First we discuss DDUCKS, our integration test-bed. Then we build the decision model by adding knowledge roles, methods, and mediating representations to solve different kinds of problems.

3.1. AN INTEGRATING INFRASTRUCTURE

As part of a project entitled Design of Information Systems (Benda, 1990) we are defining an "open architecture" integrating environment. We call this environment DDUCKS (decision and design utilities for comprehensive knowledge support). DDUCKS is the underlying infrastructure for our group decision support workbench. Previously we used individual components of DDUCKS for several applications. Now we are exploring how the components can work cooperatively to help users solve complex decision problems (Bradshaw, Covington, Russo & Boose, 1990, 1991).

An underlying intermediate representation in DDUCKS stores decision information. DDUCKS transforms this information between internal components and ultimately between mediating representations. Users seeing information in different forms triggers insights since each representation makes different information explicit while hiding other information.

Sometimes DDUCKS needs additional information to assist the transformation. For instance, gIBIS criteria must receive range, type and weight information before their use in a repertory grid. The most difficult transformation—between grids and influence diagrams—is discussed in detail in Bradshaw and Boose (1990) and in Bradshaw, Covington, Russo and Boose (1990).

We attempt to use a common vocabularly for knowledge roles and processes across all components of the model. For instance, we use the term *criterion* for: the decision literature term *decision variable,* the statistical term *dimension,* the psychological terms *construct, characteristic* and *trait,* the decision table term *attribute,* the IBIS term *argument,* and the design term *rationale*. Designing the common information structure underlying these terms was a large part of the effort of building our integrated model. Understanding these relationships simplifies the problem of representing and transforming information for different modeling tasks.

While this paper focuses on decision model content, DDUCKS also contains process management tools. These could be used to model and execute the process of using the group decision support workbench. They could also model the decision's business enterprise context (Bradshaw, Holm, Kipersztok, Nguyen & Covington, 1992).

3.2. BUILDING THE MODEL

The first section below discusses a simple brainstorm-and-score model where we generate and rank alternatives.

In later sections we add additional knowledge roles and methods to the brainstorm-and-score model. We build knowledge roles from the elements of a decision—information, preferences and alternatives. Each new section describes the portions of the model needed for solving certain additional kinds of problems.

Described along with each new decision model component are its knowledge roles, application problem characteristics and examples, processes and methods used, example mediating representations and group use issues.

FIGURE 6. The processes of generating and scoring alternatives are at the heart of most decision problems.

3.2.1. Part 1: brainstorm and score—rapid decisions using implicit criteria

A simple decision-making process includes generating and ranking alternatives (Figure 6). This process is valuable for making rapid decisions or decisions where it is not beneficial to build any more of the underlying model (Figure 7).

Description. A team that wants to increase the quality of customer service might use a brainstorm-and-score system as follows:

1. Participants use **brainstorming** to generate ideas for the question, "What inhibits our customers from being fully satisfied with our products and our services?" Participants enter and view ideas anonymously, in parallel.
2. **Idea organization** tools help the team pull out key inhibitors to achieving customer satisfaction from the brainstormed list. Participants organize comments from brainstorming into a list of key issues.
3. Each group member **votes**, rating each inhibitor on it potential for improving customer satisfaction.
4. **Brainstorming** again generates ideas on how to overcome the worst inhibitors.
5. Finally, **electronic commenting** helps form team policy—"To improve customer satisfaction, who in this room should do what and when? What are the next steps?" Again participants enter and view ideas anonymously.

Section 1.1 summarized the applications, benefits, and group uses of systems supporting this model.

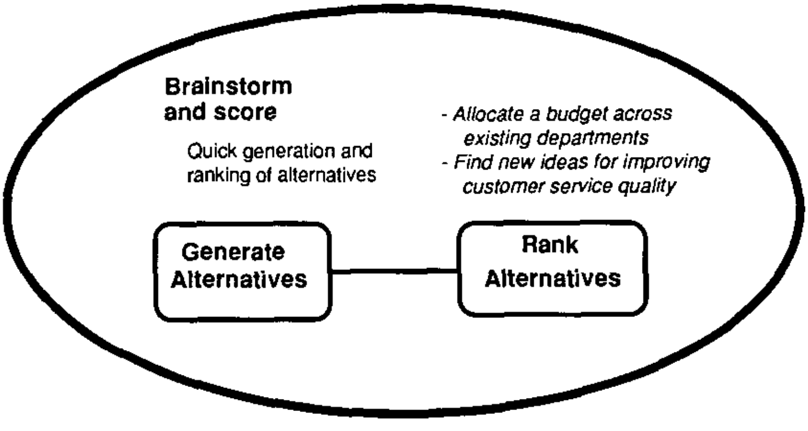

FIGURE 7. A fast and simple way to reach a group decision is to brainstorm alternatives and score them. Tools such as GroupSystems, TeamFocus and VisionQuest support individual and group display of decision results.

Processes, methods, knowledge roles. This section includes typical processes, methods and knowledge roles for brainstorm-and-score systems.

Generate requirements—Define the needs that help identify the problem.
Define problem—Define the problem and a process for solving it. Some tools have *templates* or *agendas* that define process paths through sets of tools.
Organize problems—Use *outlining* to group problems in categories.
Define participants—Define the members and their roles (such as participant or facilitator) in the problem-solving process with *rosters* or *group lists.*
Organize participants—Categroize participants by session or problem type.
Generate alternatives—Identify potential solutions to a problem through *brainstorming* or *structured commenting.*
Organize alternatives—Organize alternatives in categories with an *outlining* or *grouping* tool.
Weight (score) alternatives—Score solutions for a problem individually or by group. Common methods include placing items in order (*ranking*), assigning *rating score, voting* (yes/no/abstain), *selecting* several alternatives from a list, and *allocating fixed resources* among alternatives.
Generate and view team results—Display *consensus* results and *variation* within the group.

3.2.2. Part 2: structured outlining—linking alternatives and criteria

We can increase the utility of the group decision model by using explicit criteria (Figure 8). The relationships between alternatives and criteria and the criteria themselves can have simple or complex structures.

The gIBIS model uses unstructured criteria (text entries) in structured relationships (tagged outlines) (Conklin & Begeman, 1988, 1989) (Figure 9). Although designers use gIBIS primarily for design rationale capture the structure applies equally well for developing other kinds of decision discussions in a structured format.

A gIBIS-like model addresses the problems of **documenting running discussions** and **making criteria explicit** (mentioned in Section 1.3).

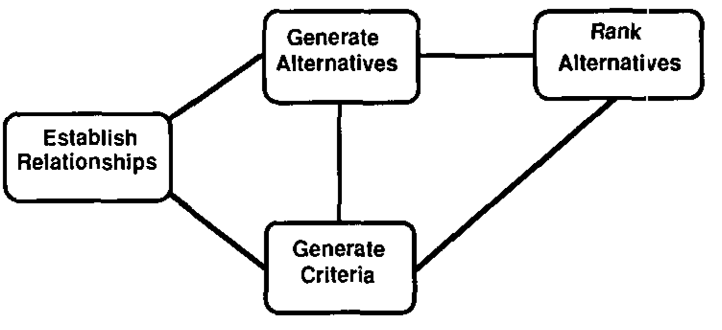

FIGURE 8. Sometimes making criteria explicit helps the group to reach a better decision. Explicit criteria allow alternatives to be examined in detail. They can also illuminate areas of agreement and disagreement.

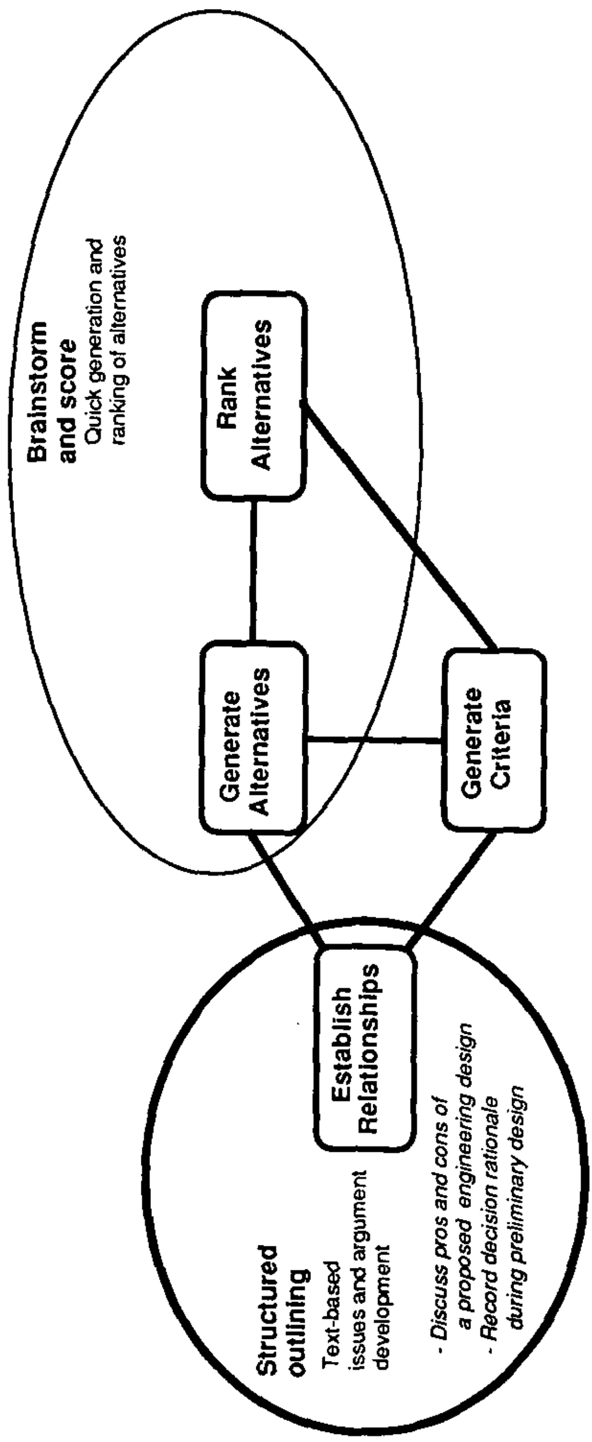

FIGURE 9. Group members can record detailed arguments for and against alternatives with structured outlining tools such as gIBIS.

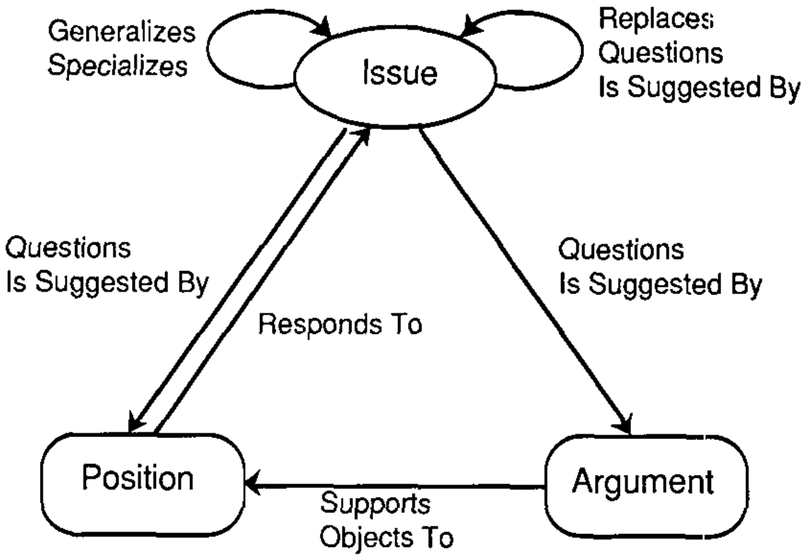

FIGURE 10. gIBIS represents several types of criteria for alternatives.

Description. We have adapted gIBIS's vocabulary to blend with our decision model. gIBIs's *issue* is the overall problem; *positions* are alternatives; *arguments* are criteria.

The term gIBIS means "graphical issues-based information system". gIBIS initially captured design rationale. The system provides a language and browser that allow individuals to add criteria for and against alternatives for certain problems (Figure 10).

The mediating representation is an outline where *indentation* shows relationships between alternatives and criteria, and *tags* show other aspects of criteria, such as whether criteria are for or against a position (Figure 11). This language is straightforward to learn and read. gIBIS provides a browser that shows the relationships graphically.

Since gIBIS records arguments as they evolve the model also captures aspects of change over time.

Other tools such as SYBIL extend the gIBIS model (Lee, 1990). SYBIL helps

```
*I: Which processor should be used?
     ?P: Processor A.
          AS: Fast
     *P: Processor B.
          AS: Already in use, thus cheaper.
     -P: Processor C.
          AO: Won't be available in time.
```

FIGURE 11. Indentations in IBIS text files represent hierarchical relationships. Issues are labelled with "I", positions with "P", supporting arguments as "AS", and objecting arguments as "AO". "?" means that the position is still open; "*" means that the position is resolved; "–" means that the position is rejected (Yakemovic & Conklin, 1990).

manage dependencies, uncertainty, viewpoints, and precedents by extending the relationship and tagging structure.

Applications. The IBIS method is useful for design and planning (Kunz & Rittel, 1980). It helps in structuring exploratory thinking in groups, addressing group decision support, conversational structuring, and management of group memory. Designers in group projects used gIBIS during document analysis, requirements and design meetings, and personal brainstorming (Yakemovic & Conklin, 1990).

Processes, methods, knowledge rules. This section includes typical processes, methods, and new knowledge roles for gIBIS.

Define the problem—Identify the problem issue and distribute it to the group.

Develop alternatives—The group develops position alternatives, adding them to the information base.

Develop criteria—The group advances arguments for and against alternatives in a running format.

Generate document—The group generates a final document showing accepted, rejected, and open alternatives.

3.2.3. Part 3: grids—linking multiple alternatives and criteria

We can further increase the utility of the group decision model by adding structure to the relationship between criteria and alternatives and by adding structure to the criteria themselves. Explicit criteria in a *grid* format show relationships between alternatives and enable analysis of decision information. Static and dynamic analyses measure a grid's problem-solving power. Other analysis tools measure the independence and completeness of the alternative and criterion space. Defining criteria and later analyses can be critical parts of the decision modeling process (Figure 12).

Building and analysing grids address the problems of defining **complex criteria** and measuring their effect (mentioned in Section 1.3). Decision methods associated with grids address some of the problems of representing criteria **minimums, maximums** and **ranges**.

Description. DART (Design Alternative Rationale Tradeoffs) is a repertory-grid-based knowledge acquisition tool. We originally developed DART for NASA as part of an effort to capture design knowledge for the Space Station Freedom program (Boose, Shema & Bradshaw, 1990a, 1990b). Similar tools and concepts have been under development at The Boeing Company for many years (Boose, 1984, 1985; Boose & Bradshaw, 1987; Boose, Bradshaw, Kitto & Shema, 1989; Boose, Shema & Bradshaw, 1989). DART contains elicitation, analysis, representation, and inference metods derived from personal construct theory (Kelly, 1955). Other researchers have contributed elicitation and analysis techniques to grid methods for knowledge acquisition, notably Shaw (1979), Gaines and Shaw (1981, 1990), Diederich, Ruhman and May (1987), Garg-Janardan and Salvendy (1987), Gaines (1987), Shaw and Gaines (1987), and Ford, Stahl, Adams-Webber, Novak and Jones (1990).

Knowledge acquisition tasks performed by DART include eliciting criteria, decomposing problems, combining uncertain information, incremental testing, integration of data types, automatic expansion and refinement of decision information, use of constraints during decision reasoning, and providing process guidance. DART interviews users and helps them analyse, test and refine the model. The

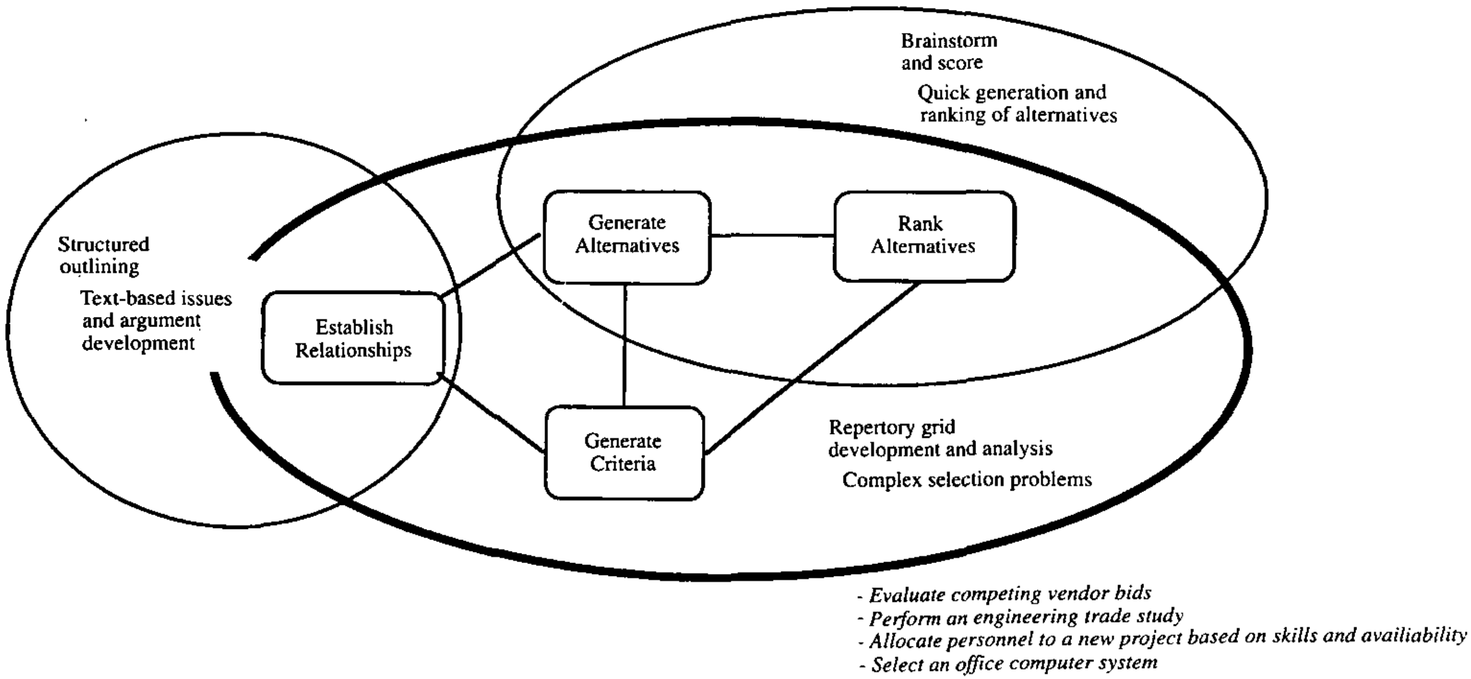

FIGURE 12. Adding more structure to the relationships between alternatives and criteria enables more sophisticated analysis of the decision information. Repertory grids represent information in a rich tabular form that aids statistical analysis and use of machine learning methods. Tools such as DART augment grids with constraints and preferences.

system represents, analyses, and reasons with information from multiple users separately or in combination. DART derives decision results by propagating information through alternative and criterion hierarchies.

DART uses tools to elicit and structure information about alternatives, criteria, constraints and preferences. Static and dynamic analysis tools in DART help determine the adequacy of the decision model and help focus users' attention on parts of the model needing further refinement.

DART also uses *machine learning* techniques to help refine the decision model. Group members may use induced information to further refine the decision model. Machine learning takes place in DART in interactive and automatic forms. Interactive forms include implication generation, analysis, and review. Automatic forms include strategies embedded in the reasoning mechanism and methods to automatically improve the decision model.

Applications. Figure 13 shows a repertory grid for a NASA Space Station Freedom problem. Two teams verified a module's location on the current design configuration. We used DART to elicit and analyse information from both teams separately and then combined the grids together to reach a solution.

Repertory grids have been applied to hundreds of kinds of problems. In Boeing we applied them to aircraft design and manufacture (e.g. aerodynamic analysis, noise certification, documentation updating, materials technology, plant location), computer systems maintenance and design (e.g. hardware and software selection, product design and impact, statistics interpretation), defense and space system design (e.g. Space Station Freedom trade studies, composite materials, energy control systems, image analysis), procurement and sales (e.g. vendor selection, customer product matching, services advising, sales support), and organization support and administration (e.g. employee evaluation, hot-line help, office automation, personnel selection, training) (Boose, 1988). Table 4 shows examples of repertory grid use at Boeing.

Processes, methods, knowledge roles. This section includes typical processes, methods, and new knowledge roles for repertory grids.

Solution (TEAM–2.SOLUTION), Trait (TEAM–2. TRAIT)

2	2	1	1	1	5	5	4	4	60	R	TRAFFIC–PATTERN	(1 LESS–DISRUPTION, 5 MORE–DISRUPTION)
2	2	1	1	1	5	5	4	4	100	R	SUPPLY–ROUTE	(1 MINIMUM–DISTANCE, 5 LONGEST–DISTANCE)
4	4	4	4	4	2	2	2	2	20	R	INSTALLATION–REMOVAL	(1 LONG–TIME, 5 SHORT–TIME)
2	1	2	1	3	2	1	3	1	40	R	CG–SHIFT–IMPACT	(1 LESS–CG–SHIFT, 5 GREATER–CG–SHIFT)
5	1	5	1	1	1	5	1	1	80	R	GROWTH–PATH	(1 BETTER–FOR–GROWTH, 5 WORSE–FOR–GROWTH)
2	2	2	2	3	1	3	3	5	40	R	STATION–VISIBILTY	(1 IMPROVES–VISIBILTY, 5 HINDERS–VISIBILTY)
3	2	3	2	1	3	1	1	2	80	R	METEOR–DEBRIS	(1 INCREASES–WEIGHT, 5 DECREASES–WEIGHT)

NODE–4–NADIR
NODE–4–PORT
NODE–3–STARBOARD
NODE–3–ZENITH
NODE–2–PORT
NODE–2–NADIR
NODE–2–ZENITH
NODE–1–NADIR
NODE–1–ZENITH

FIGURE 13. One design team's simple repertory grid for an engineering trade study problem. DART combines information from several designers to show consensus and dissension.

TABLE 4
Repertory grids have a wide application range

Aircraft Design and Manufacture

Aerodynamic Anal.—Geom. Adv.
Aerodynamic Anal.—Front End Advisor
Aerodynamic Anal.—Back End
Aircraft Fault Isolator
Airplance Mass Property Est. Risk Analyser
Airplane Design Flutter Analyser
Airplane Noise Certification Advisor
Automated Numerical Control—Cutter Consultant
Automatic Flight Controls Diag. Aid
B1 Diagnostic Consultant
Bond Durability Consultant
Documentation Update Consultant
Failure Modes and Effects Analyser
Finish Advisor for Design Engineers
Finish and Corrosion Control Consultant
Flight Controls Human Factors Assistant
GT-STRUDL Structural Analysis Adv.
Jet Engine Diagonsitc Aid
Jet Engine Fault Isolator
Jet Engine Manual Advisor
Materials Technology Advisor
Molded Rubber Seal Advisor
Parts Quality Control Consultant
Plant Location Selector
Propulsion System Advisor
Resin Advisor for Composite Parts
Rivet Selector
Structural Analysis Software Selector
Transport Airplane Config. Selection Adv.
Velocity Analysis Advisor

Computer Systems Design and Maintenance

Business Computing Needs Advisor
Business Graphics Package Con.
Comp. Comm. System Bug Adv.
Computer Languages Consultant
Database Management System Consultant
Graphics Package Advisor
Micro Computer—Needs Analysis
Micro Computer—System Consultant
Micro Computer—Workstation Config. Adv.
Micro Computer Trouble Shooter
Product Design and Impact Advisor
Product—Comparative Analysis
Product Marketing Advisor
Prog. Language Applications Adv.
Programming Language Eval. and Selector
Questionnaire Development
Rel. Database Construction Adv.
Software Management Consultant
Software Quality Advisor
Software Release Advisor
Software Services Advisor
Statistics Interpretation
Statistics Package Use Advisor

Defense and Space System Design

Carbon Dioxide Removal (Space Station Freedom)
Circuit Breaker Interface (Space Station Freedom)
Composite Materials Advisors
Drawing Formats (Space Station Freedom Program)
Energy Control System Model Eval.
Experiment Configuration
Helicopter Avionics Diagnostic Aid
Helicopter Hover Advisor
Helicopter Stick Position Advisor
Helicopter Vibration Diagnostic Aid
Image Feature Analysis
Navigation System Advisor
Potable Water System (Space Station Freedom)
Pressurized Logistics Module Placement (Space Station Freedom)
Shuttle Experiment Config.
Space Station Window Materials Con.
Visual Target Identifier

Procurement and Sales

AI Vendor Consultant
Contract Award Advisor
Customer–Product Matching
On-line Services Advisor
Sales Support Information Analysis
Technical Sales and Serivces Consultant

Organization Support and Administration

Delphi Group Information Gathering
Employee Evaluation
Hot-line Consulting Aid
Management Motivation Analyser
Negotiations—Unreasonable Offer Response
Negotiations Advisor—Info. Seeking
Negotiations—Joint Gain
Negotiations Advisor—Obtaining Leverage
Negotiation Arbitration Environment
Office Automation System Advisor
Office Automation Workstation Advisor
Organization Devel. Intervention
Organizational Climate Diagnostician
Personnel Resource Mgt Decision Aid
Personnel Technology Matching
Personnel Selection Consultant
Research and Devel. Lab Site Advisor
Research Project Priority Advisor
Survey Reporting and Analysis
Task Priority Manager
Tax Regulation Consultant
Training Course Evaluator
Training Curriculum Advisor

Generate alternatives—Potential solutions for a problem form the horizontal grid axis. *Similarity analysis, implication analysis,* and *table completion* help generate new alternatives or classes of alternatives not yet represented in the grid.

Weight alternatives—Score solutions for a problem. This may be done with *rank ordering* which scores alternatives based on criteria value preferences. Scoring is also possible using *pairwise comparison. Similarity analysis* of the alternatives yields a static measure of the criteria's ability to discriminate among the alternatives.

Organize alternatives—Organize sets of alternatives for better analysis, comprehension, and to reduce complexity. *Cluster analysis* suggests possible organizations (Figure 14).

Generate criteria—Make factors affecting alternatives explicit. Criteria definitions include type (nominal, ordinal, interval, ratio), value range, weight, and other information that helps the reasoning process. Many methods help generate criteria. *Triadic* and *dyadic comparison* methods elicit criteria that distinguish between certain sets of alternatives. *Alternative similarity analysis* points out highly similar alternatives that would always receive similar scores during decision making. The user enters new criteria to help further distinguish between them. *Criteria similarity analysis* measures criteria subsumption and independence. *Boundary analysis* encourages users to generate hidden or extrnal criteria that may control a situation. *Laddering* helps generate criteria at higher or lower levels of abstraction to ground problems and to break important criteria into subcriteria. Criteria are also generated during some types of decision model *verification* and *debugging* based on rank order results.

Organize criteria—Organize sets of criteria for better analysis, comprehension, and to reduce complexity. *Cluster analysis* suggests possible organizations (Figure 15).

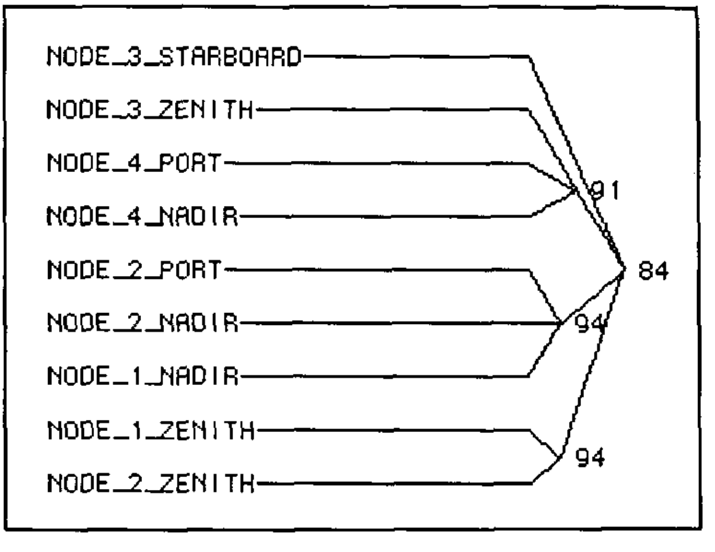

FIGURE 14. Cluster analysis shows users similarities and helps them decompose or reorganize problems. Users can label numeric junctions and DART automatically subdivides the grid into subgrids.

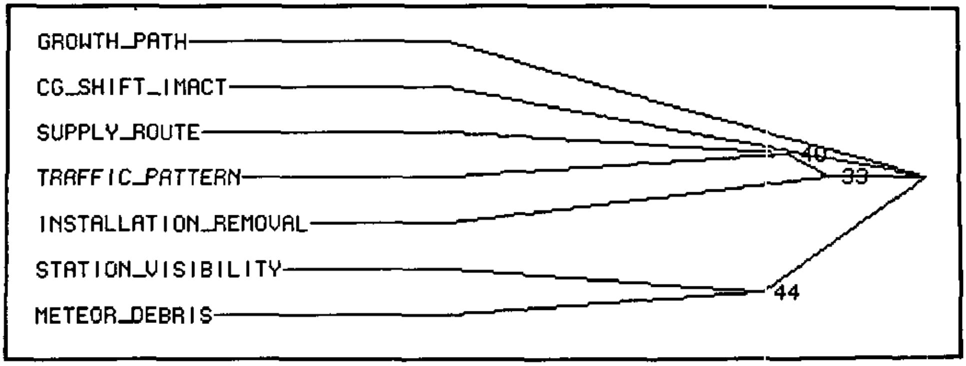

FIGURE 15. Here cluster analysis is done for criteria for the configuration design problem.

Laddering can help organize criteria into networks. *Implication analysis* shows subsumption relationships and other patterns among criteria.

Weight criteria—*Weights* record the relative importance of criteria. *Pairwise comparison* can measure the consistency of weight assignments. A decision tree generation algorithm does a *dynamic cost benefit analysis* of criteria during reasoning. *Sensitivity analysis* shows where to focus on model expansion.

Develop criteria—Assign a criterion type (nominal, ordinal, interval, ratio, cyclic) according to the level of precision needed. *Dialogs* and *value analysis* may suggest criterion types.

Generate repertory grid—Score sets of alternatives against sets of criteria in a table format.

Generate preferences—Express preferences for criteria values (soft and hard constraints). Criteria value preferences help to score and rank alternatives.

Generate generalizations—Make generalizations from information in the grid. *Implication analysis* helps check the accuracy of the grid by using a machine learning technique to generate information at a higher level of abstraction (Figure 16). This also shows criteria independence. Implication patterns can show criteria *inconsistencies, ambiguities, and equivalences.*

An algorithm developed by Gaines finds implications between criterion values (Gaines, 1989). The system uses a repertory grid as a set of examples. Criterion

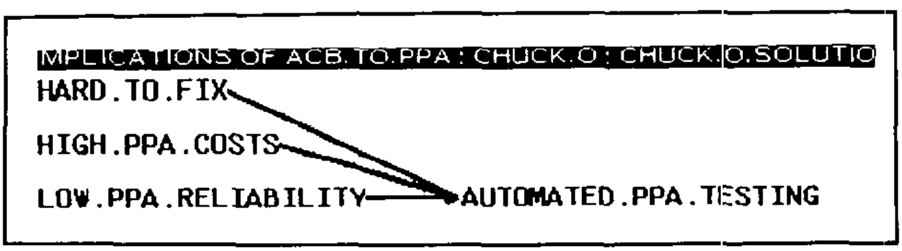

FIGURE 16. The graph shows the results of an implication analysis. Each of the criterion values HARD.TO.FIX, HIGH, PPA.COSTS, and LOW.PPA.RELIABILITY imply that AUTOMATED.PPA.TESTING is necessary (from an on-board circuit breaker engineering trade study). Participants use inductive generalizations to help refine the model.

values are viewed as logical predicates, alternatives are the operands of the predicates, and ratings are fuzzy truth values. Graphs and lists show implications and their strengths.

Implications show relationships at higher levels of abstraction that are implied by a repertory grid. If the user disagrees with an implication, DART helps refine the grid. Frequently, the user can think of an exception to the implication (a new alternative) that disproves it. The user enters this alternative, rates it, and the implication strength is reduced appropriately. Sometimes implications point out inconsistencies in the way that the user applies a criterion. In such cases a specialization generalization dialog (*laddering*) helps decompose inconsistent criteria into consistent subcriteria.

Generate constraints—Elicit constraints between alternatives and between criteria. Some constraints limit the choice of alternatives given certain criteria values. Other constraints restrict the values of interdependent criteria during reasoning.

Generate team results—Show combined results of alternative ranking from members of a team (Figure 17). Show team *consensus* and corresponding *dissenting opinions*. DART finds a dissenting opinion by computing a correlation score between each individual and the consensus; the individual with the lowest correlation score is listed as the dissenting opinion. Dissenting opinions show users the *range* of opinion about a decision, not just the top scoring list. Dissenting opinions give decision makers confidence that the top rated alternatives were sound choices or point out areas of disagreement for further exploration.

DART also analyses and summarizes the *differences* between grids (Figure 18). Other methods point out possible differences in vocabulary and meaning and compute *subsumption* relationships between team members (discussed below).

Evaluate results—Use static and dynamic analyses to evaluate score results and improve the decision model.

Static evaluation—Evaluate the *potential* of a decision model. *Similarity analysis of alternatives* measures the ability of criteria to distinguish between alternatives. *Similarity analysis of criteria* measures the coverage of alternatives and measures criteria subsumption and independence. *Implication analysis* looks for accuracy of

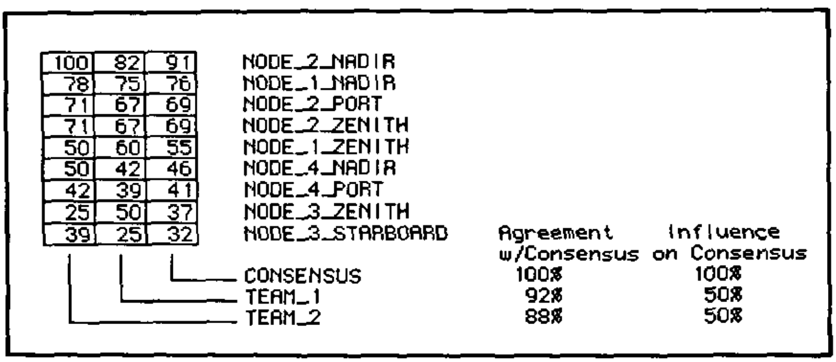

FIGURE 17. A rank ordering of alternatives shows the consensus and the contributions of individuals.

```
Difference Grid

.25  -  .25  -  .25 .50 .75 .25 .75 1: .33 ******* CG_SHIFT_IMACT
.75  -  .75  -  1.0 .75 .25 1.0 .75 2: .58 ************ GROWTH_PATH
-----------------------------------------
.50 .00 .50 .00 .62 .62 .50 .62 .75 Total: 8.25 (0.46 ave/cell)

1: .50 **********  NODE_1_ZENITH
2: .00   NODE_1_NADIR
3: .50 **********  NODE_2_ZENITH
4: .00   NODE_2_NADIR
5: .62 ************  NODE_2_PORT
6: .62 ************  NODE_3_ZENITH
7: .50 **********  NODE_3_STARBOARD
8: .62 ************  NODE_4_PORT
9: .75 ***************  NODE_4_NADIR
```

FIGURE 18. Differences between grids point out areas of agreement and disagreement. Here both teams agree most about the alternatives NODE_1_NADIR and NODE_2_NADIR but disagree most about NODE_4_NADIR and the criterion GROWTH_PATH.

information at higher levels of abstraction and checks criteria for consistency, ambiguity and equivalence.

Dynamic evaluation—Evaluate the *results* of a decision model by testing it. Compare results with *expectations* to measure and verify the performance of the decision model. Use this information to generate new criteria to fix specific problems and improve the performance of the grid.

Group use. Grids have been used before in several specialized group settings. Grids *combine in different ways depending on commonality* between alternatives and criteria (Figure 19). Groups can use iterative methods to proceed from grids where nothing is in common to those with shared alternatives and criteria. Grids with common features can be analysed in a variety of ways. Shaw used grid comparison techniques to measure difference between pairs of grids and subsumption relationships across groups of grids (Shaw, 1979, Shaw, 1988). Boose adapted this work and added consensus and dissenting opinions when using grids to reach group decisions (Boose, 1986, 1988, 1989; Boose & Bradshaw, 1987; Boose, Shema & Bradshaw, 1989). Chang used similarity analysis between grids in a dynamic group setting to show similarities and differences as grids evolved (1986, 1991).

Shaw, Gaines and Woodward explored the conceptual space of consensus, conflict, correspondence and contrast available when comparing grids (Shaw & Gaines, 1988, 1989; Shaw & Woodward, 1988; Figure 20). The facilitator may use several methods for conflict resolution. Boose (1986) describes a structured negotiation technique where participants independently develop grids (divergence), iteratively identify and merge important criteria and alternatives (convergence), and finally reach a decision showing consensus and dissenting opinions (also used in Schuler, Russo, Boose & Bradshaw, 1990). Shaw and Gaines (1989) describe a detailed method of problem formalization, conceptualization and feedback, exchange and comparison of information, and validation.

3.2.4. *Part 4: possibility tables—developing models of alternatives*

Often a simple text entry is not enough to represent an alternative. Alternatives may need to be synthesized from components that involve interacting constraints. For

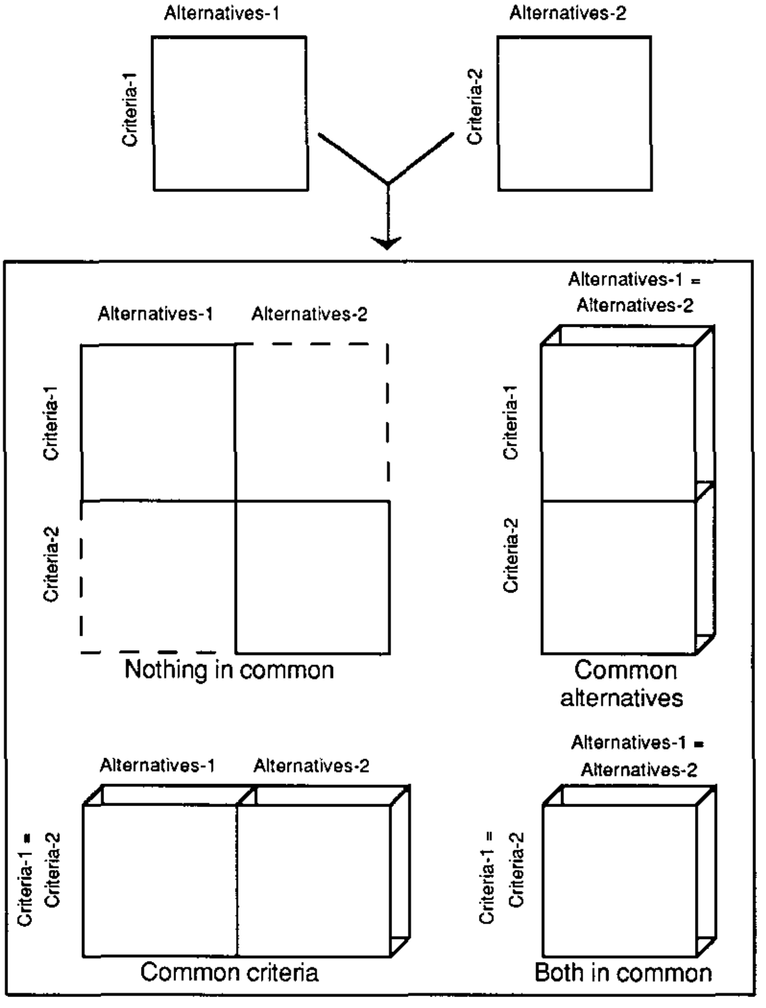

FIGURE 19. Grids combine in different ways depending on commonality between alternatives and criteria. Group processes help members build common grids that enable further analysis and shared understanding.

example, during the evolution of a design concept, designers must integrate diverse sources and kinds of information about requirements, constraints, and tradeoffs. In doing so, they evaluate alternatives for suitability under certain assumptions and by applying criteria. Unfortunately, much of this process is implicit, making later review difficult if not impossible. When requirements change, impacts on the design are difficult to trace. This can lead to serious errors and costly rework.

Defining and choosing complex alternatives can be an important part of the decision modeling process (Figure 21). Possibility tables handle some of the problems of representing **minimums, maximums** and **ranges**. They also address the problems of **restructuring complex alternatives** and **exclusivity** mentioned in Section 1.3.

Description. We developed a stand-alone tool named Canard that helps synthesize

Terminology

Criteria	Same	Different
Same	**Consensus** Participants use terminology and concepts in the same way	**Correspondence** Participants use different terminology for the same concepts
Different	**Conflict** Participants use same terminology for different concepts	**Contrast** Participants differ in terminology and concepts

FIGURE 20. Shaw and Gaines (1989) measured sets of grids and mapped the results based on differences and similarities in terminology and concepts. Their system discovered, for example, that one expert used the criterion "low level data—high level data" in the same way that another used the criterion "nominal data—interval or ratio data". This type of analysis leads to sharing an understanding of vocabulary and concepts between group participants (figure adapted from Shaw and Gaines, 1989).

alternatives from potentially large search spaces (Bradshaw, Boose, Covington & Russo, 1989; Shema, Bradshaw, Covington & Boose, 1990). Canard helps generate and structure complex alternatives in a possibility table. We adapted the possibility table representation from manually developed strategy tables (McNamee & Celona, 1987) and morphological charts (Zwicky, 1969). Decision analysts and designers have used these tools 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).

Canard also enhances possibility tables with constraint-handling tools that reduce the possible solution space and capture important information about the design. The system allows entry of both hard and soft constraints. A designer adds hard constraints to capture information about incompatibilities and interdependencies between component possibilities. During alternatives generation, these hard constraints prevent selection of any incompatible components.

Designers may add soft constraints and utility scores to criteria that characterize generated paths. Preferences for these criteria are the design goals. Conflicts between two or more design goals often require making tradeoffs between the goals. For example, a goal of low cost is often in conflict with a goal of high reliability. Using Canard, the designer can specify the acceptable range of a criterion and map attribute values to the utility of these values. Canard then guides the designer toward possibilities that optimize the tradeoffs between the soft constraints.

Canard provides documentation of the alternative synthesis history. It captures and stores the information used in defining the alternatives. With such a capability, a more complete record of the decision is available for later review and revision.

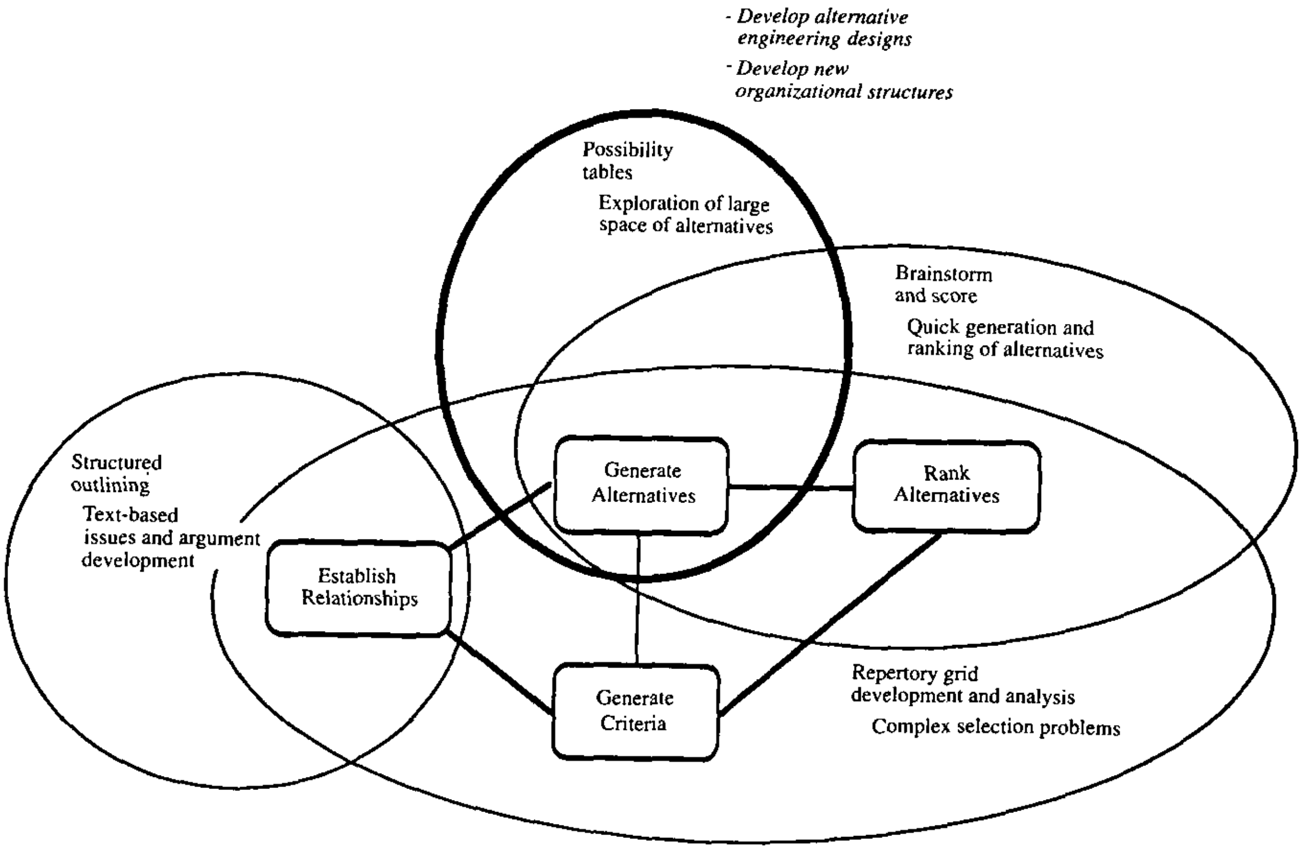

FIGURE 21. Tools other than brainstorming may be necessary when a problem requires more thorough exploration of a large space of potential alternatives. Possibility tables (also known as morphological charts) help decision makers define paths and constraints through complex components of alternatives.

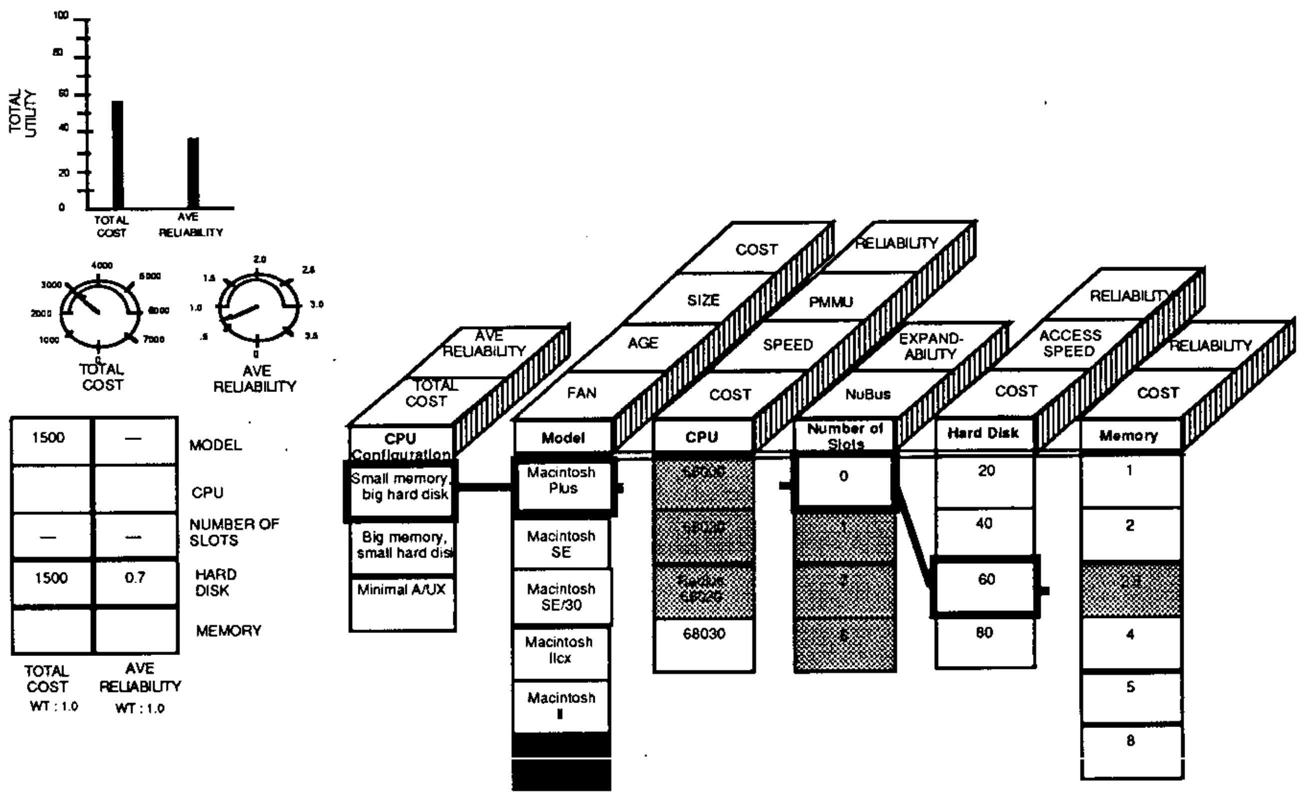

FIGURE 22. A Canard possibility table for a portion of a network design problem. The thick horizontal line is a partial solution path linking component alternatives. Shaded cells show hard and soft constraints associated with the path. The left-most column shows classes of solutions associated with paths. Repertory grids "plug in" to each column, recording criteria relationships and enabling analysis. Repertory grid reasoning methods help rank the components in a column. Automated methods combine ranking information with constraints to produce a set of best possible solutions.

We are also concerned with helping decision makers better explore the space of alternatives. Cognitive scientists have long known that people typically retrieve only a small fraction of available alternatives when generating hypotheses (Wise, 1985). People tend to anchor on initial guesses, giving insufficient regard to subsequent data. For various other reasons, people may not be able to visualize whole classes of possibilities (Kahneman, Slovic & Tversky, 1982). DDUCKS maintains links between possibility tables and related repertory grids. By coupling the alternative generation facility to its analysis tools and those in DART, Canard helps users get a better feel for the effects of the constraints on different alternatives and helps them better evaluate the consequences of assumptions and tradeoffs.

Possibility tables could also help with the idea convergence process following brainstorming. Possibility tables could capture the constraints and reasoning behind synthesized categories as items are added. Then as the constraints and reasoning change Canard could automatically manage the associated alternatives.

Applications. Typical applications of possibility tables in Canard include developing alternative engineering designs and structuring new organizations. Figure 22 shows a possibility table for a Boeing network design problem. In this problem Canard stores standard network configuration components and constraints. Customers specify requirements and givens (for example, existing hardware, the number of workstations, or needed applications). The system shows allowable configurations as paths through the component columns. The designer may modify a recommended design (path) based on exceptions or special circumstances. Gauges show running costs and other constraints.

Canard also helps create new problem solutions. The designer builds several component columns, graphically specifying incompatibilities and other constraints between component columns. Combinations of constraints suggest certain partial paths through the columns, which in turn suggest additional components, new columns of components, new constraints, new partial paths and so on. Shema, Bradshaw, Covington and Boose (1990) show an example of an engineer discovering new robot arm applications and designs. A similar method could help administrators design a new organization structure.

Possibility tables in Canard have a graphical interface which eases creation and change of design information. The designer creates new solution classes and alternatives and defines solution paths. The interface also assists the designer in managing the complexity of large design problems by presenting the design in an easily comprehensible view.

Processes, methods, knowledge roles. This section includes typical processes, methods, and new knowledge roles for possibility tables.

Generate **alternative classes** and **class components**—A column header and its cells represent design components or functions and possible alternatives for the component.

Generate **compatibility constraints**—Constraints between cells within a column or between columns record incompatibilities and help the designer find paths through the table. These appear in the table or on a *compatibility tree* (Figure 23).

Generate a **solution path**—Build a set of links between cells that represent a partial or complete solution class. These may be specified by the designer or through

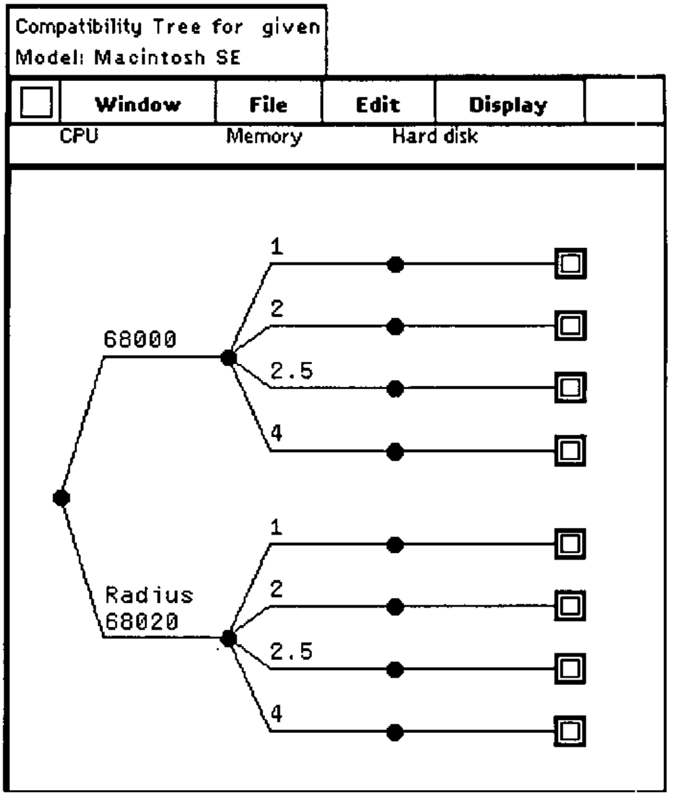

FIGURE 23. A mediating representation for a partial compatibility tree, showing CPU, memory, and hard disk configurations. Alternative values and criteria value probabilities are recorded in a similar fashion.

semi-automatic *path generation* based on optimizing solution path criteria such as cost and reliability.

Generate **solution classes**—Cells in the left-most column show names of completed paths through the table. Partial solution paths may suggest new solution classes.

Generate **solution path criteria**—Global criteria associated with solution paths keep track of items such as cost and reliability. These criteria may reside in an underlying repertory grid and possess weights, preferences, ranges, types and constraints. A utility mapping tool helps users record preferences and constraints (Figure 24).

Group use. Experiments using possibility tables in groups are in progress. In the simplest situation, multiple designers contribute non-overlapping information covering different aspects of computer network design. This information combines to form a larger possibility table.

More complex situations will involve:

1. Combining columns and cells in reasonable structural ways.
2. Merging different cells under the same column headings.
3. Computing differences in constraints when multiple participants share the same cells across columns.

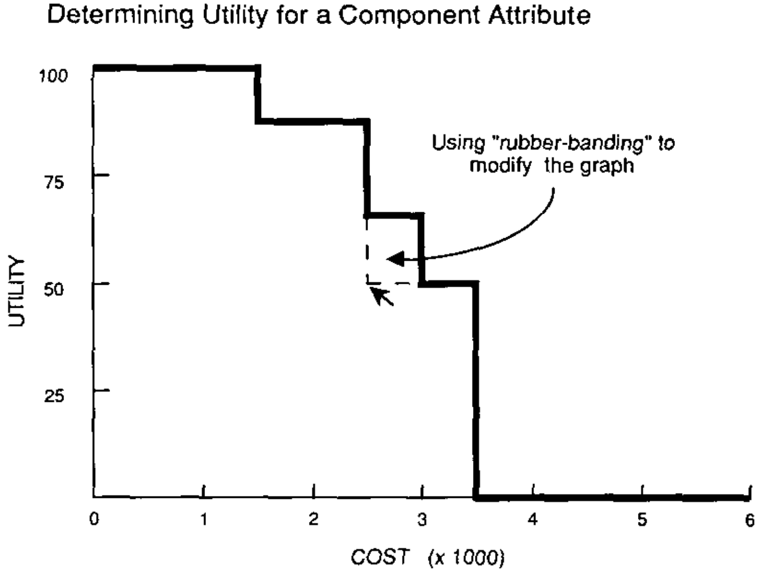

FIGURE 24. Users map preferences, hard and soft constraints and values to utility scales.

4. Measuring content and pointing out conflicts of vocabulary or concepts (similar to the way that Shaw and Gaines measure such differences for repertory grids, above).
5. Specifying constraints between different segments of the model contributed by different participants.
6. Analysing differences in the related repertory grids that hold path criteria.
7. Showing consensus and range scores for decisions using this information.

We are developing group facilitation processes to resolve differences when they are uncovered. We would like to dynamically measure differences across a group working in parallel on individual yet overlapping parts of a large table.

3.2.5. Part 5: decision analysis—modeling uncertainty and risk

Uncertainty and risk are important components of many decisions yet they are seldom included in a formal model (Figure 25). Gathering information about risk and uncertainty may incur a high cost, but automated techniques and good mediating representations can reduce this cost. We have borrowed and extended techniques from decision analysis and knowledge-based systems to build Axotl, a stand-alone decision analyst's workbench. In Axotl, *influence diagrams* from decision analysis (a concise form of probabilistic decision trees) show the conditional relationships between criteria (Figure 26). They enable rigorous decision path scoring. Associated techniques can help expand the decision model and uncover areas where it is important to find further information. In the future we will be incorporating methods from Axotl in our group decision support workbench.

Influence diagrams and associated representations address the problems of modeling **enabling conditions, risk,** the **effects of uncertainty,** and **timing** issues (mentioned in Section 1.3).

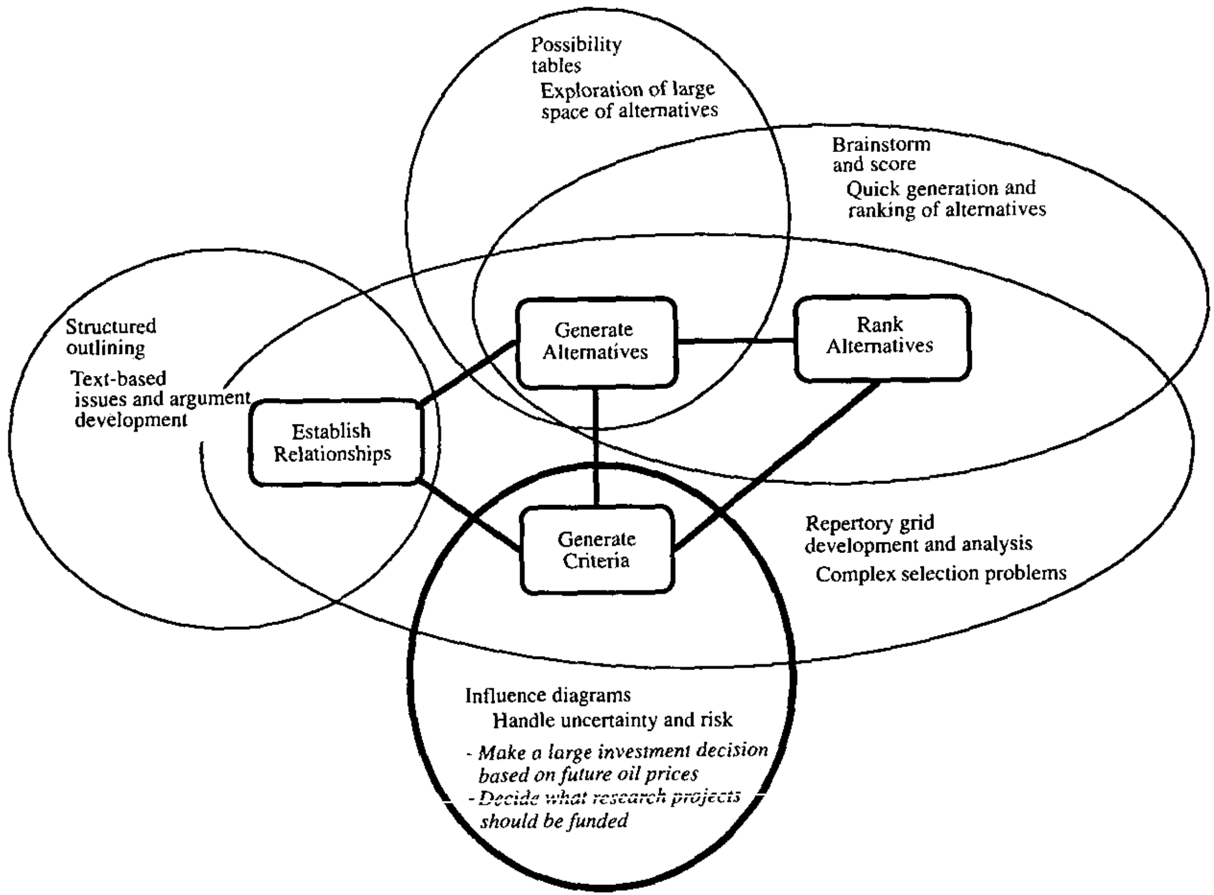

FIGURE 25. Decision analysis techniques handle conditions of high risk or high uncertainty for some types of important problems. An influence diagram is a concise form of a probabilistic decision tree showing the conditional relationships between criteria. It enables rigorous decision path scoring. Associated techniques help expand the decision model and uncover areas where it is important to find further information.

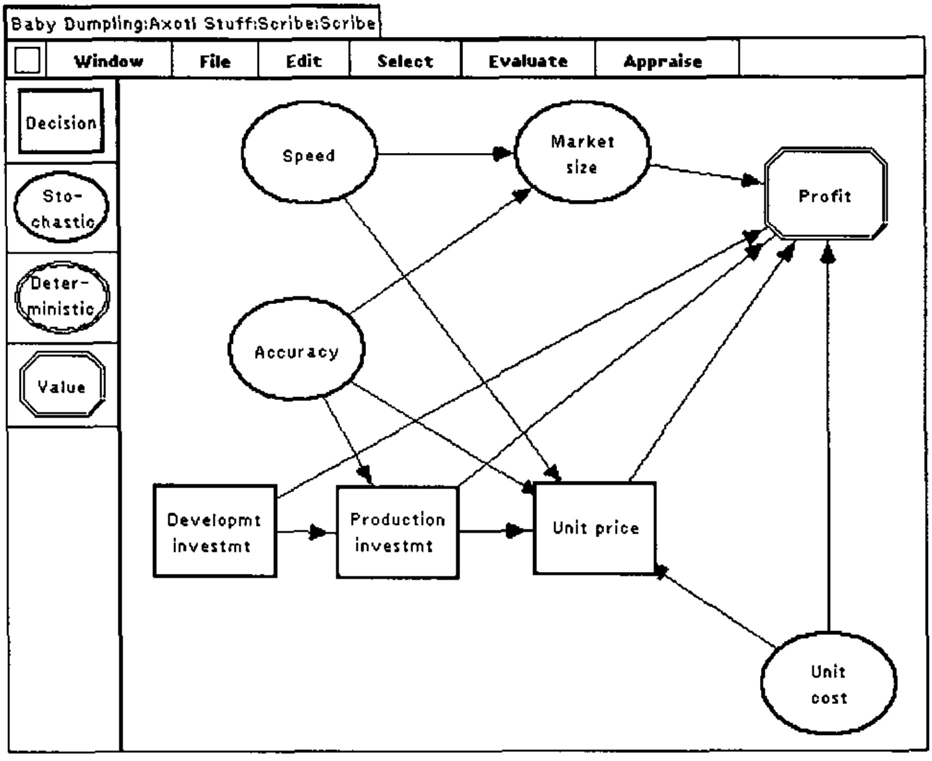

FIGURE 26. An influence diagram for an R&D investment decision about an automated speech-to-text transcriber. Three decisions form the investment strategy (development investment level, production investment level and unit price). Rectangular nodes on the diagram represent these decisions. Oval nodes represent technical risk variables (accuracy, speed), production uncertainties (unit cost), and market uncertainties (market size). The eight-sided node labeled "Profit" is the criterion to maximize in evaluating the decision model to determine an optimal policy. Arrows between nodes represent relevance or information flows between variables. For example, arrows, from "Speed" and "Accuracy" to "Market size" represent judgements about the relevance of technical achievement to our assessment of market size.

Description. A promising approach for dealing with risk and uncertainty is decision analysis (Howard, 1966; Raiffa, 1968; Keeney & Raiffa, 1976; Howard & Matheson). In the past few years, several tools attempted to help automate portions of the decision analysis process. However, the current generation of automated decision analysis tools (such as decision tree software) are limited in scope and assume a high level of sophistication in the theory and practice of decision analysis. These tools contain some of the *algorithms* of decision analysis practice, but do not embody the *experience* and *intuition* of decision analysis professionals in formulating and appraising decision models. Also, because current 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.

Axotl works with knowledge-based templates that contain some of the experience of a decision analyst. It combines a decision analysis workbench with knowledge-based tools to assist individuals consulting with the system about decisions involving high stakes, difficult tradeoffs, or critical uncertainties and risks (Bradshaw & Boose 1990; Bradshaw, Covington, Russo & Boose, 1990, 1991). Axotl contains a graphical editor that helps create and refine influence diagrams that model relevant decision alternatives, preferences, and uncertainties. Influence diagrams are solved to obtain recommended actions in a way that is consistent with probability and utility theory (Howard & Matheson, 1980). The system computed an *expected value* or *utility* for each alternative that expresses the anticipated range of benefit or cost for a given course of action.

While mathematically similar to probabilistic decision trees, influence diagrams possess several advantages: (1) influence diagrams grow linearly in their graphical representation as contrasted with the exponential growth of trees; (2) they can represent and exploit conditional independence and (3) in implementation they connect to external procedures and functions in a straightforward way. Additionally, our experience confirms that influence diagrams are an effective way of communicating important issues among participants in a decision, even for those who may not understand the mathematical underpinnings.

The influence diagram solution method implemented in Axotl incorporates a new approach that allows a wide range of questions to be answered directly from the diagram, and preserves the entire underlying joint distribution during solution and inference procedures, rather than just the value lottery and decision policy as is usually done (Schachter, 1986). The distribution editor is a another feature unique to Axotl. It helps users structure conditional probability distributions.

Applications. Decision analysis applies to areas such as business portfolio management, environmental issues, facilities investment and expansion, hurricane seeding, investment strategies, market forecasting, medicine, new product introduction, nuclear plant construction, research and development portfolio management, space program planning, strategic planning, and systems engineering (Howard & Matheson, 1984). At Boeing we have used Axotl to build a prototype for making research and development investiment decisions. Other applications under way include process management and bone marrow transplant follow-up care.

Developers configure the knowledge-based tools in Axotl with *application-independent* knowledge (i.e. knowledge of decision analysis tools and methods) and *application-specific* knowledge (i.e. knowledge about a particular domain) to provide guidance and help during a decision.

Processes, methods, knowledge roles. This section includes typical processes, methods, and new knowledge roles for decision analysis techniques used by Axotl.

Generate alternatives—Generate alternative choices.

Generate alternative values—Enter the amount of relative worth of each alternative using direct entry or *pairwise comparison.*

Generate criteria (influence diagram variables)—Generate the decision variables for an influence diagram.

Generate constraints—Store constraints in a *compatibility tree* structure.

Generate influence diagrams—Show decision alternatives and their values, criteria and their interrelationships, and associated conditional probabilities.

Weight criteria—Measure the relative importance of criteria using *sensitivity analysis.* Sensitivity analysis enables individuals to determine which variables are

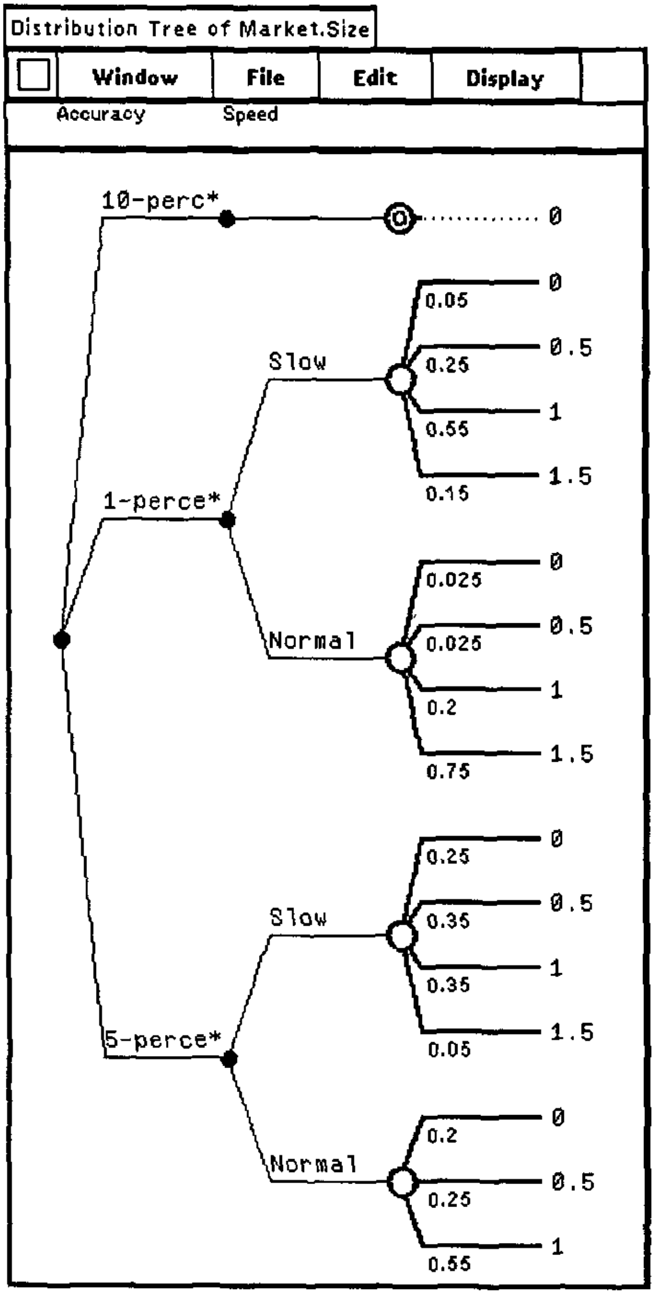

FIGURE 27. The distribution tree for "Market size". The right-most nodes and their branches represent the atomic distributions. The atomic distributions of "Market size" are conditioned on its direct predecessors in the influence diagram, "Accuracy" and "Speed". The *conditioning tree* is the part of the distribution tree that precedes the atomic distributions. The ability to explicitly structure conditioning trees and to represent them using a richer language permits reasoning about distributions with additional clarity and efficiency.

the most important determinants of final value (e.g. "Is speed more important than accuracy?"). *Value-of-information* analysis is useful in understanding the importance of resolving uncertainty for specific components of the model (e.g. "How much should I spend to estimate the size of the market?"). *Value-of-control* analysis focuses attention on new alternatives that can increase the ability to bring critical uncertainties under our control (e.g. "Should we control unit cost by acquiring a semiconductor company that manufactures the required chips?"). Graphical facilities supporting value-of-information analysis and value-of-control analysis enable those consulting the system to determine the value of undertaking activities to gather new information or generate new alternatives (Howard & Matheson, 1984).

Generate criteria probabilities (uncertainties)—Estimate the uncertainty associated with criteria values. A *distribution editor* helps structure conditional probability distributions (Figure 27). The system also uses *probability wheel* techniques to encode uncertainty. *Knowledge maps* can help break down complex probability assessments into simpler ones.

Generate value-of-information for criteria—Solve the influence diagram to find the value of reducing uncertainty about criteria values.

Generate value-of-control for criteria—Solve the influence diagram to find the value of being able to control criteria values.

Generate expected values—Solve the influence diagram to find the most reasonable alternative.

Group use. Merging influence diagram information in group settings will involve:

1. Merging smaller individual influence diagrams into larger, more complex ones.
2. Computing and reporting differences in the values of outcomes when participants share the same alternatives.
3. Computing and reporting differences in the values of variable probabilities when participants share the same criteria.
4. Finding differences in preferences, value and risk by comparing utility graphs.
5. Showing consensus and range scores for expected values of decisions using this information.

4. Summary

Existing group decision support systems can help teams reach decisions quickly and efficiently. We pointed out some of the deficiencies of these decision models when using them for complex problems. Weaknesses include the lack of the models' ability to handle documentation of running discussions, complex criteria and alternatives, numeric ranges, exclusivity, enabling conditions, risk management, uncertainty and timing issues.

We proposed that models used by some successful knowledge acquisition could handle many of these problems. Knowledge acquistion and group decision making are modeling activities. We discussed the importance of good mediating representations. Many of the mediating representations used by knowledge acquisition tools would be useful in a group decision support system. We are in a unique position to combine methods from these areas given our laboratory's extensive experience in implementing and using knowledge acquisition techniques.

Next we pointed out the basic buidling blocks of decision models—information, preferences and alternatives. There is a tradeoff between the cost and the benefit of building a more complex model. Techniques such as sensitivity analysis can help point out when to expand critical parts of the model.

Finally we described components of a complex decision model. We described the modeling strengths contributed by each component. We gave a short description of each component along with the component's methods, applications, knowledge roles (built up from the basic decision element building blocks) and group usage. The components included the brainstorm-and-score model, a gIBIS-like model for recording running discussion, repertory grids for enabling analysis, possibility tables to handle complex alternatives and constraints, and decision analysis techniques to handle risk and uncertainty.

All these components exist in successful tools. Their analyses and methods will be critical in helping groups to gain insight and understanding about complex problems. We are blending them into a single workbench that will be powerful enough to handle the complex problems facing Boeing's work teams.

References

Benda, M. (1990). *Design of information systems: towards an engineering discipline.* Boeing Computer Services technical report.

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

Boose, J. H. (1985). A knowledge acquisition program for expert systems based on personal construct psychology. *International Journal of Man–Machine Studies,* **23(5),** 495–525.

Boose, J. H. (1986). Rapid acquisition and combination of knowledge from multiple experts in the same problem domain. *Future Computing Systems Journal,* **1(2),** 191–216.

Boose, J. H. (1988). Uses of repertory grid-centered knowledge acquisition tools for knowledge-based systems. Speical issue on the 2nd AAAI-sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop, 1987. *International Journal of Man–Machine Studies,* **29(3),** 287–310.

Boose, J. H. (1989). Using repertory grid-centered knowledge acquisition tools for decision support. *Proceedings of the 1989 IEEE Hawaii International Conference on System Sciences,* January.

Boose, J. H. (1992). Knowledge acquisition tools, methods, and mediating representations. In S. Shapiro, Ed. *Encyclopedia of Artificial Intelligence, 2nd Edition.* New York: Wiley (in press), 719–742.

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 AAAI-sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop, 1986, Part 1. *International Journal of Man–Machine Studies,* **26(1),** 3–28.

Boose, J. H., Bradshaw, J. M., Kitto, C. M. & Shema, D. B. (1989). From ETS to Aquinas: six years of knowledge acquisition tool development. *Proceedings of the Fourth Knowledge Acquisition for Knowledge-Based Systems Workshops,* pp. 5.1–17, Banff, October.

Boose, J. H., Shema, D. B. & Bradshaw, J. M. (1989). Recent progress in Aquinas: a knowledge acquisition workbench. *Knowledge Acquisition* **1(2),** 185–214.

Boose, J. H., Shema, D. B. & Bradshaw, J. M. (1990*a*). Capturing design knowledge for engineering trade studies. In B. Wielinga, J. Boose, B. Gaines, G. Schreiber, M. van Someren, Eds. *Current Trends in Knowledge Acquisition.* Amsterdam: IOS Press.

BOOSE, J. H., SHEMA, D. B. & BRADSHAW, J. M. (1990*b*). Design knowledge capture for a corporate memory facility. *Proceedings of the Fifth Conference on Artificial Intelligence for Space Applications*, Huntsville, May.

BRADSHAW, J. M. & BOOSE, J. H. (1990). Decision analysis techniques for knowledge acquisition: combining information and preferences using Aquinas and Axotl. *International Journal of Man–Machine Studies*, **32(2)**, 121–186.

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. *Proceedings of the Third Knowledge Acquisition for Knowledge-Based Systems Workshop*, pp. 4.1–15, Banff, November.

BRADSHAW, J. M., COVINGTON, S., RUSSO, P. & BOOSE, J. H. (1990). Knowledge acquisition techniques for Intelligent Decision Systems: integrating *Axotl* and *Aquinas* in DDUCKS. In M. HENRION, R. D. SCHACHTER, L. N. KANAL & J. F. LEMMER, Eds. *Uncertainty in Artificial Intelligence 5.* Amsterdam: Elsevier.

BRADSHAW, J. M., COVINGTON, S., RUSSO, P. & BOOSE, J. H. (1991). Knowledge acquisition techniques for decision analysis using *Axotl* and *Aquinas*. *Knowledge Acquistion*, **3(1)**, 49–78.

BRADSHAW, J. M., HOLM, P., KIPERSZTOK, O., NGUYEN, T. & COVINGTON, S. (1992). Acquiring knowledge for process management using eQuality. *Hawaii International Conference on Systems Science*.

CHANG, E. J. H. (1986). Participant systems. *Future Computing Systems*, **1(3)**, 253–270.

CHANG, E. J. H. (1991). Group coordination in participant systems. *Proceedings of the 24th Annual Hawaii International Conference on System Sciences, Vol. III*, pp. 589–599, January.

CLANCEY, W. J. (1984). Classification problem solving. *Proceedings of the National Conference on Artificial Intelligence*, Austin, Texas.

CLANCEY, W. J. (1986). Knowledge acquisition. *Presentation at the First Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, October.

CLANCEY, W. (1990). Implications of the system-model-operator metaphor for knowledge acquistion. *Proceedings of the First Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop (JKAW'90)*, pp. 65–80, Tokyo: Ohmsha.

COLLABORATION TECHNOLOGIES COMPANY (1991). *VisionQuest Manuals*. Austin, TX: Collaboration Technologies Company.

CONKLIN, J. & BEGEMAN, M. L. (1988). gIBIS: a hypertext tool for exploratory policy discussion. *ACM Transactions on Office Information Systems*, **6**(4), 303–331.

CONKLIN, J. & BEGEMAN, M. L. (1989). gIBIS: a tool for all reasons. *Journal of the American Society for Information Science*, May, 200–213.

COX, L. A. (1991). Knowledge acquisition for model building. In K. M. FORD & J. M. BRADSHAW, Eds. Special knowledge acquisition issue of the *International Journal of Intelligent Systems*, in press.

CYERT, R. M. & MARCH, J. G. (1963). *A Behavioral Theory of the Firm*. Englewood Cliffs, NJ: Prentice-Hall.

DANIELS, R. M., DENNIS, A. R., HAYES, G., NUNAMAKER, J. F., Jr. & VALACICH, J. (1991). Enterprise Analyzer: electronic support for group requirements elicitation. *Proceedings of the 24th Annual Hawaii International Conference on System Sciences, Vol. III*, pp. 43–52, January.

DENNIS, A. R., GEORGE, J. F., JESSUP, L. M., NUNAMAKER, J. F., Jr & VOGEL, D. R. (1988). Information technology to support electronic meetings. *MIS Quartely*, **12(4)**, 591–624.

DENNIS, A. R., VALACICH, J. S. & NUNAMAKER, J. F., Jr. (1991). Group, sub-group, and nominal group idea generation in an electronic meeting environment. *Proceedings of the 24th Annual Hawaii International Conference on System Sciences, Vol. III*, pp. 573–579, January.

DICKSON, G. W. (1991). An overview of the Minnesota GDSS research project and the SAAM system. In G. R. WAGNER, Ed. *Computer Augmented Teamwork: A Guided Tour*. New York: Van Nostrand Reinhold.

DIEDERICH, J., RUHMAN, I. & MAY, M. (1987). KRITON: a knowledge acquisition tool for expert systems. *International Journal of Man–Machine Studies*, **26(1)**, 29–40.

ESHELMAN, L., EHRET, D., MCDERMOTT, J. & TAN, M. (1987). MOLE: a tenacious knowledge acquisition tool. *International Journal of Man–Machine Studies*, **26**, 41–54.

ESHELMAN, L. (1988). MOLE: a knowledge acquisition tool that buries certainty factors. Special issue on the 2nd Knowledge Acquisition for Knowledge-Based Systems Workshop, 1987. *International Journal of Man–Machine Studies*, **29(5)**, 563–578.

FORD, K. M. & ADAMS-WEBBER, J. R. (1991). Knowledge acquisition and constructivist epistemology. In R. HOFFMAN, Ed. *The Psychology of Experts: Cognitive Research and Empirial AI*. New York: Springer-Verlag.

FORD, K., STAHL, H., ADAMS-WEBBER, J., NOVAK, J. & JONES, J. C. (1990). ICONKAT: an integrated constructivist knowledge acquisition tool. *Proceedings of the 5th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada, November.

GAINES, B. R. (1987). Advanced expert system support environments. In J. H. BOOSE & B. R. GAINES, Eds. *Proceedings of the Second Knowledge Acquisition for Knowledge-Based Systems Workshop*, pp. 8.0–8.14, Banff, October.

GAINES, B. R. (1989). An ounce of knowledge is worth a ton of data: quantitative studies of the trade-off between expertise and data based on statistically well-founded empirical induction. *Proceedings of the 6th International Workshop on Machine Learning*, pp. 156–159, San Mateo, CA.

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.

GAINES, B. R. & SHAW, M. L. G. (1990). Cognitive and logical foundations of knowledge acquisition. *Proceedings of the 5th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada, November.

GARG-JANARDAN, C. & SALVENDY, G. (1987). A conceptual framework for knowledge elicitation. *International Journal of Man–Machine Studies*, **26(4)**, 521–531.

GALLUPE, R. B., DESANCTIS, G. & DICKSON, G. (1988). Computer-based support for group problem finding: an experimental investigation. *MIS Quarterly*, **12(2)**, 277–296.

GRUBER, T. R. (1989). *The Acquisition of Strategic Knowledge*. New York: Academic Press.

GRUBER, T. R. (1990). Justification-based knowledge acquisition. In H. MOTODA, R. MIZOGUCHI, J. BOOSE & B. GAINES, Eds. *Knowledge Acquisition for Knowledge-Based Systems*. Amsterdam: IOS Press.

HOWARD, R. A. (1966). Decision analysis: applied decision theory. In D. B. HERTZ & J. MELESE, Eds. *Proceedings of the Fourth International Conference on Operational Research*. Reprinted in HOWARD & MATHESON, 1984.

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

JARVENPAA, S. L., RAO, V. S. & HUBER, G. P. (1988). Computer support for meetings of groups working on unstructured problems: a field experiment. *MIS Quarterly*, **12(4)**, 645–665.

JOHNSON, N. E. (1989). Mediating representations in knowledge elicitation. In D. DIAPER, Ed. *Knowledge Elicitation: Principles, Techniques and Applications*. New York: John Wiley.

KAHNEMAN, D., SLOVIC, P. & TVERSKY, A. (1982). *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge: Cambridge University Press.

KARBACH, W., LINSTER, M. & VOß, A. (1990). Model-based approaches: one label—one idea? In B. WIELINGA, J. BOOSE, B. GAINES, G. SCHREIBER & M. VAN SOMEREN, Eds. *Current Trends in Knowledge Acquisition*, pp. 173–1789. Amsterdam, Washington, Tokyo: IOS Press.

KEENEY, R. L. & RAIFFA, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York: Wiley.

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

KLINKER, G. (1989). A framework for knowledge acquisition. *Proceedings of EKAW-89:*

Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems, pp.
102–116, Paris, July.

KRAMER, K. L. & KING, J. L. (1988). Computer-based systems for cooperative work and group decision making. *ACM Computing Surveys*, **20(2)**, 115–146.

KUNZ, W. & RITTEL, H. (1980). *APIS: a concept for an argumentative planning information system.* Working Paper No. 324, Institute of Urban and Regional Development, University of California at Berkeley.

LARKIN, J. H. & SIMON, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, **11**, 65–99.

LEVI, I. (1986). *Hard Choices: Decision Making Under Unresolved Conflict.* Cambridge: Cambridge University Press.

MARCH, J. G. (1978). Bounded rationality, ambiguity, and the engineering of choice. *The Bell Journal of Economics*, **9(2)**, 587–608.

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

MUSEN, M. A. (1988). An editor for the conceptual models of interactive knowledge-acquisition tools. *Proceedings of the the third Knowledge Acquisition for Knowledge-Based Systems Workshop*, pp. 23.1–20, Banff, November.

NORMAN, D. A. (1988). *The Psychology of Everyday Things.* New York: Basic Books.

NORMAN, D. A. (1991). Cognitive artifacts. In J. M. CARROLL, Ed. *Designing Interaction: Psychology at the Human–Computer Interface.* New York: Cambridge University Press.

NUNAMAKER, J. F., Jr. (1991). Facilitation in group support technologies. Tutorial notes, *24th Annual Hawaii International Conference on System Sciences*, January.

NUNAMAKER, J. F., Jr., APPLEGATE, L. M. & KONSYNSKI, B. R. (1988). Computer-aided deliberation: model management and group decision support. *Operations Research*, **23(6)**, 826–848.

NUNAMAKER, J. F., Jr., VOGEL, D., HEMINGER, A., MARTZ, B., GROHOWSKI, R. MCGOFF, C. (1989). Group support systems in practice: experience at IBM. *Decision Support Systems*, **5(2)**, 183–196.

NUNAMAKER, J. F., Jr., WEBER, E. S., SMITH, C. A. P. & CHEN, M. (1988). Crisis planning systems: tools for intelligent action. *Proceedings of the 21st Annual Hawaii International Conference on System Sciences, Vol. III*, pp. 25–34, January.

RAIFFA, H. (1968). *Decision Analysis: Introductory Lectures On Choices Under Uncertainty.* Reading, MA: Addison-Wesley.

SCHULER, D. C., RUSSO, P. J., BOOSE, J. H. & BRADSHAW, J. M. (1990). Using personal construct techniques for collaborative evaluation. *International Journal of Man–Machine Studies*, **33**, 521–536.

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

SHAW, M. L. G. (1979). Conversational heuristics for eliciting shared understanding. *International Journal of Man–Machine Studies*, **11**, 621–634.

SHAW, M. L. G. (1988). Problems of validation in a knowledge acquisition system using multiple experts. *Proceedings of the Second European Knowledge Acquisition Workshop (EKAW-88)*, pp. 5.1–15, Bonn, June.

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(3)**, 251–280.

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 Knowledge Acquisition for Knowledge-Based Systems Workshop*, pp. 30.1–19, Banff, November.

SHAW, M. L. G. & GAINES, B. R. (1989). Comparing conceptual structures: consensus, conflict, correspondence and contrast. *Knowledge Acquisition* **1(4)**, 341–364.

SHAW, M. L. G. & WOODWARD, J. B. (1988). Validation in a knowledge support system: constructing consistency with multiple experts. Special issue on the 2nd Knowledge Acquisition for Knowledge-Based Systems Workshop, 1987. *international Journal of Man–Machine Studies*, **29(3)**, 329–350.

SHAW, M. L. G. & WOODWARD, J. B. (1989). Mental-models in the knowledge acquisition process. *Proceedings of the Fourth Knowledge Acquisition for Knowledge-Based Systems Workshop*, pp. 29.1–24, Banff, October.

SHAW, M. L. G. & WOODWARD, J. B. (1990). Modeling expert knowledge. *Knowledge Acquisition*, **2(3)**, 179–206.

SHEMA, D. B., BRADSHAW, J. M., COVINGTON, S. P. & BOOSE, J. H. (1990). Design knowledge capture and alternatives generation using possibility tables in Canard. *Knowledge Acquisition*, **2(4)**, 345–364.

SIVARD, C., ZWEBEN, M., CANNON, D., LAKEN, F. & LEIFER, L. (1989). Conservation of design knowledge. *Proceedings of the 27th Aerospace Sciences Meeting (AIAA-89)*, January, Reno, NV.

SYCARA, K. & ROBOAM, M. (1991). Intelligent information infrastructure for group decision and negotiation support of concurrent engineering. *Proceedings of the 24th Annual Hawaii International Conference on System Sciences, Vol. III*, pp. 658–667, January.

TAN, B. C.-Y., WEI, K.-K. & RAMAN, K. S. (1991). Effects of support and task type on group decision outcome: a study using SAAM. *Proceedings of the 24th Annual Hawaii International Conference on System Sciences, Vol. III*, pp. 537–546, January.

VENTANA CORPORATION (1990). *GroupSystems Manuals*. Tucson, AZ: Ventana.

WIELINGA, B., AKKERMANS, H., SCHREIBER, G. & BALDER, J. (1989). A knowledge acquisition perspective on knowledge-level models. *Proceedings of the Fourth Knowledge Acquisition for Knowledge-Based Systems Workshop*, pp. 36.1–22, Banff, October.

WINSTON, P. (1984). *Artificial Intelligence, 2nd Edition*. Reading, MA: Addison-Wesley.

WISE, J. A. (1985). Decisions in design: analyzing and aiding the art of synthesis. In G. WRIGHT, ed. *Behavioral Decision Making: Theory and Analysis*. New York: Plenum Press.

YAKEMOVIC, K. C. B. & CONKLIN, E. J. (1990). Report on a development project use of an issue-based information system. *Proceedings of the Conference on Computer-Supported Cooperative Work*, pp. 105–118, October.

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