250 IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-4, NO. 3, MAY 1982

GODDESS: A Goal-Directed Decision Structuring System

JUDEA PEARL, SENIOR MEMBER, IEEE, ANTONIO LEAL, AND JOSEPH SALEH, MEMBER, IEEE

Abstract—This paper describes an operational version of a computerized, domain-independent, decision support system which is based on a novel, goal-directed structure for representing decision problems. The structure allows the user to state relations among aspects, effects, conditions, and goals, in addition to actions and states which are the basic components of the traditional decision tree approach. The program interacts with the user in a stylized English-like dialogue, starting with the stated objectives and proceeding to unravel the more detailed means by which these objectives can be realized. At any point in time, the program focuses the user's attention on the issues which are most crucial to the problem at hand. The structure used is more compatible with the way people encode knowledge about problems and actions, and therefore promises to offer the following advantages: 1) judgments and beliefs issued by the user constitute a more valid representation of the user's experience; and 2) the user may be guided toward the discovery of action alternatives he otherwise would not have identified.

Index Terms—Decision analysis, decision support systems, knowledge acquisition, knowledge-based systems, means–ends analysis, planning aids, problem structuring.

I. Introduction

DECISION support systems (DSS's) can be classified into two major categories: knowledge-based systems and situation-based systems. Knowledge-based systems store and employ a large database, which contains the features and constraints specific to a given problem environment (e.g., they may employ a large medical or legal library) and enable the user to obtain immediate access to factual information from the problem environment. It is the user's task then to mentally incorporate this information with additional inputs regarding the specific problem situation and come up with a decision strategy. Situation-based systems are domain-independent, acquiring knowledge and generating inferences concurrently. They rely on the user carrying most of the background knowledge and expertise and only map into the machine that section of knowledge which the user perceives as relevant to the problem at hand. In this mode the machine acts as a sophisticated and friendly "sounding board"; it does not provide information of its own, but it assists the user in structuring and searching his own knowledge and provides advice on alternative courses of action.

Manuscript received June 26, 1980; revised November 16, 1981. This work was supported in part by the Office of Naval Research under Contract N00014-78-C-0372 and in part by the National Science Foundation under Grant MCS 78-0748.

J. Pearl is with the Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles, CA 90024.

A. Leal and J. Saleh are with the Textrax Corporation, Canoga Park, CA 91306.

Decision-analytic technology employs situation-based support. Decision analysts who are called upon to assist in the solution of a given planning problem usually possess less specific knowledge about the problem domain than their customers. The benefit of their services stems primarily from their familiarity with a skeleton structure (i.e., a decision tree) *common to all problems*, and their ability to represent all problems within the confines of this structure and to draw optimal conclusions from the formal structure once it solidifies. While the optimization process is usually performed on electronic computers, the formalization phase has been conducted manually, using lengthy interviews with persons intimately familiar with the problem domain.

From a practical viewpoint, however, the major drawback of manual interviews is their length and cost. Since real-time analysis of decision trees is beyond the limitation of human computational capability, it invariably happens that many hours of interviews are spent on eliciting portions of the decision tree which do not have decisive bearing on the problem(s) at hand. This fact can be discovered only at a later stage once the problem structure is formalized and a sensitivity analysis has been conducted on an electronic computer. During the interview itself, however, it is impossible for the analyst to process the entire information obtained by him up to that point and to select the optimum course of conducting future inquiries.

A direct man-machine interface can provide three distinct advantages. First, it offers the capability of real-time sensitivity analysis, which in turn can be used to guide the growth of the decision tree in only the more promising directions. Second, it provides an inexpensive means of updating the program with new knowledge, even by the nontechnical decision maker. Finally, it opens the way to computerized real-time Delphi methods for aggregating opinions of several remotely located experts.

The goal of constructing a computerized structuring aid was pursued by Leal in 1976 [5]. It culminated in "An Interactive Program for Dynamic Elicitation of Decision Structures," exhibiting a computerized system which interacts with a user in stylized English and provides assistance in structuring his/her problem perception, making plan recommendations and communicating the structure to others [6]. The program's main techniques were borrowed from both artificial intelligence (AI) and decision analysis (DA). DA provided a formal structure of knowledge representation in the form of a decision tree quantified with probability and value assessments. AI provided techniques for heuristic search of game

0162-8828/82/0500-0250$00.75

trees, and to a lesser degree, some capabilities for natural languages processing.

Since the completion of Leal's program, the feasibility of automating the process of tree elicitation has attracted the interest of several other laboratories. Merkhofer *et al.* [8] describe a tree structuring support system for command and control applications. Leal *et al.* [7] and Steeb and Johnston [15] describe an interactive computer aiding system for group decision making designed to support crisis management situations.

GODDESS, the structuring-aid system reported in this paper, represents a methodological extension of the works above in breaking away from the confines of decision tree representations and employing a richer structure which, we believe, is more compatible with the way people perceive their problems. The paper is organized as follows. Section II presents the deficiencies of decision tree representations which prompted us to adapt the goal-directed structure outlined in Section III. Section IV describes the network of relationships constructured by GODDESS and how judgments about these relationships propagate through the network. Section V outlines the philosophy and procedures used by GODDESS to control the user's focus of attention. Section VI presents a sample dialogue between GODDESS and a user seeking financial advice. Conclusions and prospects for future developments are discussed in Section VII.

II. Deficiencies of Decision Tree Representation

Experience with the operation of Leal's program confirmed earlier hopes that due to the structural simplicity of decision trees, only very primitive levels of language-understanding would be sufficient to conduct natural, English-like dialogues. However, the lack of sophisticated language understanding features, aside from accounting for the simplicity of the program, also resulted in several deficiencies. The most serious deficiency arises from the constraint of representing knowledge in tree form.

In many real-world applications, the decision maker may not perceive a problem in the form of a time sequence of decision alternatives and event outcomes, but rather as a static network of influences surrounding *issues* and *factors*. Consider, for example, our perception of the environmental pollution problem. The issues of capital investment, energy needs, energy supply, unemployment, public health, etc., all seem to be tightly interwoven in a network of cause and effect relationships. The first step in attacking such a problem should be to explicate the underlying causal network rather than to hypothesize and evaluate various action/event scenarios.

When a person confronts such a complex problem he is rarely aware of the set of relevant alternative actions available to him at the onset. In fact, he usually hopes the analyst would help him identify those alternatives on the basis of certain things he desires to achieve and others he wishes to prevent. The user may become aware of his immediate options only after unraveling the processes which influence the desired and undesired effects, the preparations needed to make these processes more or less effective, and the conditions which should prevail before an action becomes applicable.

The major difference in the formal representation required for such problems and the one handled by decision trees is that the atomic entities admitted by the latter representation are restricted to be descriptions of "world states" or decision "situations." The decision maker can express relations among these situations but is unable to express relations between their constituents. For example, when a decision maker is asked to assess the value of a situation resulting from a given event/action sequence, he is presented with the entire sequence and is forced to aggregate the effects of all the event/action components by mental manipulations. He cannot, for example, explicitly express the belief that raising taxes is a positive contributor to unemployment regardless of other situational factors such as air pollution or the energy embargo.

Decision analysis is founded on the paradigm that the reliability of human judgments increases when the format of these judgments are made more compatible with the internal format used by people to encode experience. In fact, the sole rationale of the problem-decomposition "divide and conquer" approach is to reformulate a given problem statement in terms of many so-called more "elementary" problem statements to which reliable judgments can be assigned. The reason that one expects these elementary judgments to be more reliable than those involving global considerations is only that the former are more likely to match the format in which human experience is encoded. Research showing that decomposition improves judgment has been reported by Armstrong *et al.* [1] and Gettys *et al.* [4].

The decomposition affected by decision tree analysis offers only the first step toward a structural match between the external and the internal codes. The fragmentation, however, remains too crude to allow the user to express beliefs in a natural and therefore more reliable manner.

The main objective of the current research project has been to devise a richer structure for eliciting knowledge about decision problems, a structure in which aspects, issues, and conditions are represented as independent entities. On the basis of such a structure it becomes feasible to construct a decision support program that starting with the stated objectives, guides the decision maker toward the discovery of action alternatives he otherwise would not have identified.

III. A Goal-Directed Approach

To facilitate an "issue-oriented" problem elicitation program, the internal machine representation of problem situations could be based on the methodology known in artificial intelligence as "problem reduction" or "means–ends analysis" [9]. Each element in this structure (denoted by a node in a graph) represents a subproblem or a subgoal rather than a state description. The task of describing a problem as a collection of interdependent issues (i.e., hopes and concerns) is regarded as a reduction of the global problem into several components. These can be further reduced to their constituencies, and so on.

A "means–ends analysis" was first employed in the general-problem-solver (GPS) program developed in the late 1960's [2]. The program is controlled by "differences": a set of features which make the goal different from the current state. The programmer had to specify along what dimensions these

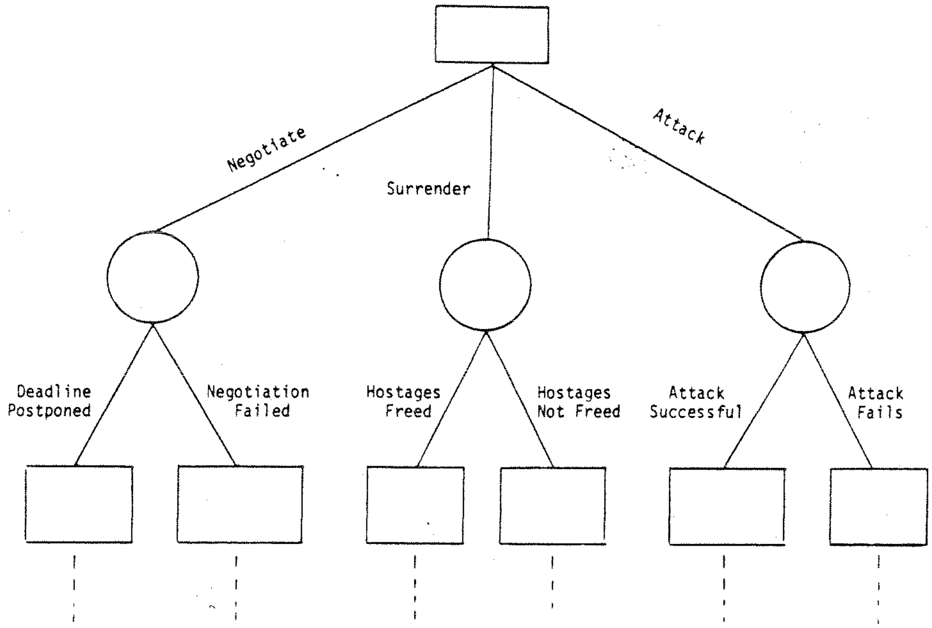

Fig. 1. Decision tree representation of terrorist attack problem.

differences are measured, which differences are easier to remove, what are the operators available for the reduction of the differences, and under what condition each reduction operator is applicable. A successful planning program called STRIPS, based on the same principles, was implemented at SRI to plan the actions of an object-manipulating robot [3]. In STRIPS also actions are brought up for consideration by virtue of their potential for reducing the differences (mismatched logical assertions) standing between the desired goal and the current state. When the current state does not possess the conditions necessary for enacting a desired difference-reducing operation, a subgoal is created to generate the missing conditions. The structure underlying this form of reasoning is no longer a tree but an *AND/OR graph.* The *OR* nodes represent various types of actions one can employ in attempting to achieve a given subgoal, and the *AND* nodes represent the remaining subgoals (differences) *all* of which should be resolved before a solution is reached. These latter sets of subproblems are of two types: the first contains a set of *preconditions* that must be realized before the enactment of a previously identified desirable action becomes feasible; the second contains a set of adverse *effects* (additional differences) introduced by such an action.

A similar *AND/OR* graph structure has been selected as the basic representation for our decision-structuring program, and since at each level of expansion the content of deeper levels is determined by the available set of subgoals, we call it a *goal-directed* program [10] with the acronym GODDESS.

To demonstrate the difference between this structure and the traditional decision tree, consider two possible conceptualizations of the problem of handling a terrorist attack. Fig. 1 represents a possible beginning of a decision tree describing the crisis, while Fig. 2 represents a goal-directed structure for the same problem. The two basic entities in the latter structure are actions (in ⌂ boxes) and subgoals or issues (in □ boxes). The root of the graph labeled TERRORIST ATTACK is recognized as involving two main issues: securing the hostages' safety and discouraging future attacks. These are connected by an AND arc to indicate that both issues must be dealt with simultaneously. At this point the natural question for the computer to ask would be, "Could you think of an action which would serve the hostages' safety and at the same time would deter future attacks?" The possibility of "ATTACK TO RESCUE" immediately comes to one's mind, and the various aspects of this suggestion are explicated. Other actions, intended to resolve each subgoal separately, are then elicited. Each action is characterized by two lists: 1) a preconditions list, and 2) an effects list. Any one of the preconditions which is not yet satisfied generates a subgoal (e.g., the condition "terrorists must agree to postpone deadline" generated the subgoal "provide terrorists with incentive for postponement").

Some arcs of the graph may point back toward higher levels in the structure. For example, one of the effects of "surrender to demands" is found to be "encourage future attacks." This generates, since it is an adverse effect, a subgoal of eliminating this effect, namely the subgoal "deterring future attacks" which is already listed in the first level.

The main advantage of this structure is that the *intent* of each action is spelled out explicitly prior to naming the action. The analysis proceeds *from the ends toward the means* encouraging the user to discover novel alternatives. For example, the alternative "negotiate for release of sick hostages" may only came to mind after drawing the subgoals "obtain information on terrorists mentality and capabilities." Clearly, similar goals may also implicitly influence one's thoughts during a decision tree elicitation. For example, the alternative "negotiate" in Fig. 1 may have been identified for the purpose of obtaining additional information about the terrorists' mentality. However, not having such purposes spelled out formally may cause the user to neglect exploring a large set of alternatives which can make up a workable solution plan.

In formal problem-solving, such as theorem proving or robot planning, problems are said to be solved when a sequence of

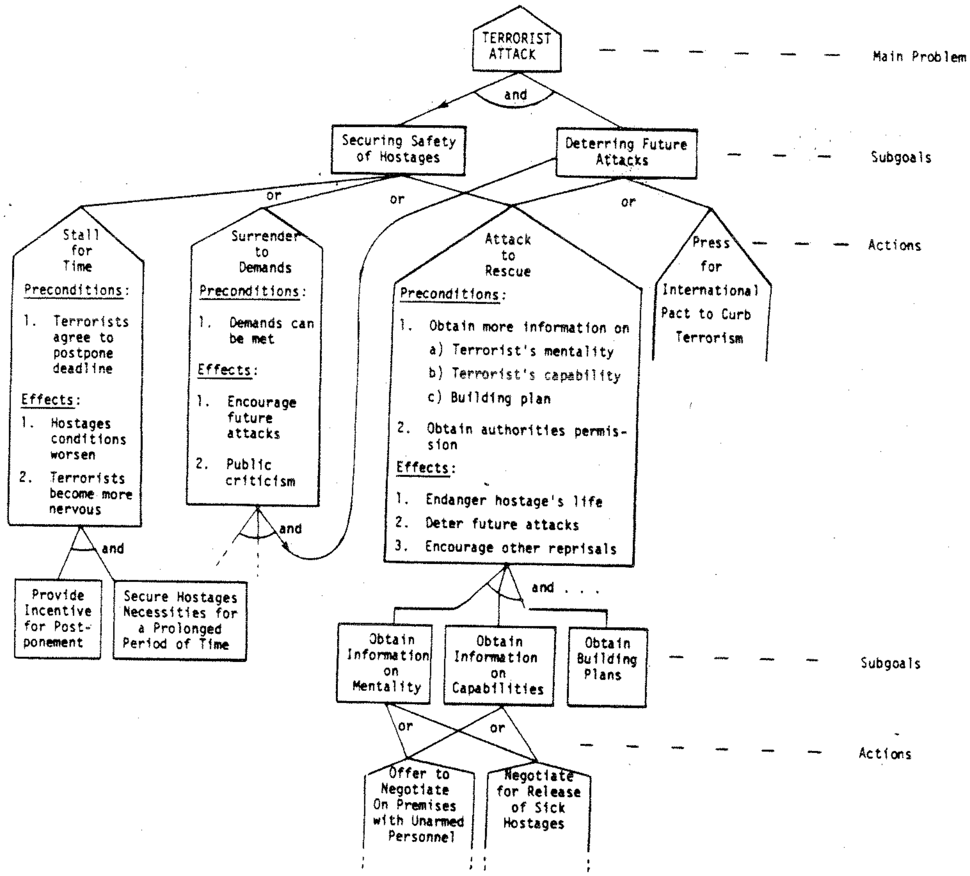

Fig. 2. Goal directed representation of terrorist attack problem.

operators is found which removes all differences between the desired and the current state. In real-life problems, such as the terrorist problem above, issues seldom get "resolved." They are at best alleviated or controlled within acceptable ranges. For example, one has no guarantee that meeting the terrorists' demands would result in the hostages' safety. The latter is only a plausible expectation. Similarly, one cannot be sure of the degree to which storming the building would deter future terrorist attacks. Such estimates must be assessed using educated guesses and quantified using a formal structure. The descriptions of the actions should also contain information on the degree to which each of the preconditions contributes to the realization of each subgoal. For example, the action "attack to rescue" should qualitatively specify how critical it is to obtain the desired information in order to secure the hostages' safety during the attack. Similarly, a value judgment must be attached to each of the mentioned subgoals in order to determine both the relative merit of candidate solution plans and the direction of future elicitation queries.

Whereas GODDESS's basic representation is tailored after the condition–action–effect structure of STRIPS, it also contains two novel features. First, it is equipped with procedures for processing partially satisfied as well as uncertain and value driven relationships. Second, GODDESS contains procedures for systematic and directional acquisition (or elicitation) of the knowledge required for synthesizing plans.

Note that the structure depicted in Fig. 2 could also constitute a "*frame*" (or template) for representing the generic aspects of terrorist-attack problems. Once elicited in detail, such a structure could be prestored as an "expert" on terrorist confrontations and be consulted when a particular crisis develops. The advantage of prestoring the "frame" is that during the crisis only the problem-specific parameters need be explored in detail. On the basis of these parameters the program could also suggest prestored contingency plans for consideration by the user, provide explanation for its suggestions, and, to some degree, be able to understand queries posed to it in English.

IV. Organizational Design and Value Propagation

Fig. 3 shows the skeleton of the structure used by GODDESS in each knowledge-acquisition cycle. Its main components are the following.

1) Goal: The major objective of the decision maker.

2) Subgoals: The goal "dimensions," "attributes," or detailed items that combine to form the overall goal.

3) Actions: The major action strategies that are open to the decision maker for advancing a particular subgoal.

4) Modes: The possible implementation methods of performing each action.

5) Preconditions: Those states of nature or the environment that are desired for permitting a particular (action) mode to be implemented effectively.

Fig. 3 should be thought of as a decision network. Thus, the goal is divided into several subgoals, each subgoal has a number of possible actions that could accomplish it, each action has a

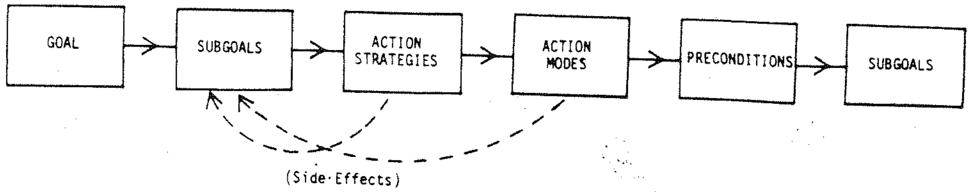

Fig. 3. Model structure.

number of ways (modes) it can be performed, and each mode has a number of preconditions that must be completed. Once the preconditions are specified, they lead directly to new subgoals, that is, the subgoal of completing the specific precondition that allows actions to be taken, etc. If the realization of a precondition is beyond the direct control of the user and is instead perceived to depend on externally controlled eventualities, that precondition is then treated as an *uncertain event* node quantified by likelihood estimates. This structure can then be repeated recursively.

Cross-relationships can also exist in the graph when an action is perceived having a beneficial or adverse effect on a subgoal to which it is not directly connected. GODDESS instructs the user to identify and characterize such "side effects" so that the program accounts for the full impact of each action.

During the knowledge-acquisition phase the user is instructed to quantify the strengths of various relationships between these components. A brief summary of the rules used by GODDESS to propagate these quantifiers through the network is provided below.

The Major Goal: The objective of GODDESS is to maximize the expected value of $G(0 \leqslant G \leqslant 1)$, which is a numerical value attached to the major goal and represents the user level of satisfaction with the accomplishments realized in the various subgoals.

Subgoals: GODDESS forces the user to express all issues, hopes, as well as concerns, as areas for potential improvements, i.e., subgoals. For example, the fear of losing one's job could be expressed as a subgoal "maintain job" or "reduce likelihood of losing job."

The goal value G is obtained from the subgoal levels and weights by a linear combination $G = \Sigma_i W_i V_i$ (i.e., linear multiattribute model) where the level $V_i (0 \leqslant V_i \leqslant 1)$ of subgoal i is the degree to which it may be completed by the plans stated and the weight $W_i (0 \leqslant W_i \leqslant 1)$ for subgoal i is a measure of its importance relative to other subgoals.

Actions: After a list of specific subgoals has been established, the user is asked to think of possible action strategies that would help produce improvements in each one of the subgoals listed. The strategies are treated as mutually exclusive; if a combination strategy is perceived to be especially promising, it ought to be listed separately.

Each action strategy is characterized by an "effectiveness" quantifier $E_j (0 \leqslant E_j \leqslant 1)$, which measures the level of subgoal attainment to be expected if action strategy j were executed.

Each action strategy can be supported by a set of *action modes* which contain more detailed specifications of how the action strategy may be implemented. For example, the mode may specify the time, place, technique, and various resources to be used in support of the parent strategy. The action mode effectiveness $E_k (0 \leqslant E_k \leqslant 1)$ is the amount that the corresponding mode affects the success of the parent strategy.

The benefit of characterizing actions by a two-level structure lies in the fact that some properties of an action strategy (e.g., preconditions) would be identical to all its modes. This would enable us to store these common sets of properties in the description of the parent strategy, thus saving the storage and elicitation time otherwise consumed by duplication.

Preconditions: A "precondition" is a feature of the environment that must exist before an action mode (or strategy) can be implemented effectively. GODDESS instructs the user to characterize each precondition by a *criticality threshold* $C(0 \leqslant C \leqslant 1)$, which represents a threshold on the degree of achievement of the precondition below which the effectiveness of the corresponding action mode is nullified.

The relationship between the precondition completion level L, its criticality C, and the effectiveness of the support mode is captured by a truncated linear function: $\delta(L, C) = (L - C)/1 - C$ if $L \geqslant C$ and $\delta(L, C) = 0$ if $L < C$.

The overall effectiveness of an action mode requiring several preconditions is obtained by taking the product $E = \Pi_i \delta(L_i, C_i)$ over all the connecting preconditions.

Whenever GODDESS realizes (using the dialogue management procedure described later) that the success or failure of the overall plan hinges critically on a given precondition, it proclaims the fulfillment of this procondition as a new subgoal. This proclamation calls the user's attention to a new spectrum of problems aimed toward satisfying the corresponding precondition, thus repeating the entire structure including action strategies, action modes, further preconditions, etc.

Uncertain Events: Uncertain events are treated as uncontrollable preconditions, i.e., a precondition whose level of attainment is affected by factors other than the planner's actions. GODDESS associates two parameter-vectors with each uncontrollable precondition. The first vector $[p(t_1), p(t_2), \cdots]$ contains the probability of occurrence of each uncertain outcome. The second vector $[(L|t_1), (L|t_2), \cdots]$ contains the level of completion of the precondition, given the occurrence of the corresponding uncertain outcome. Once these vectors are elicited, the system examines the elements of the second vector and proclaims a new subgoal aimed at increasing the probability of the most desirable outcome, i.e., the one with the highest $(L|t)$.

As the expansion of the new subgoal continues, the probability vector is updated. Using the two vectors above, the expected level of attainment of the major goal can be calculated by $G = \Sigma_i (G|t_i)\, p(t_i)$, where $(G|t_i)$ is the value of the major goal computed by assuming that outcome t_i has occurred and that the best action was accordingly selected.

Side Effects of Actions on Subgoals: GODDESS alerts the

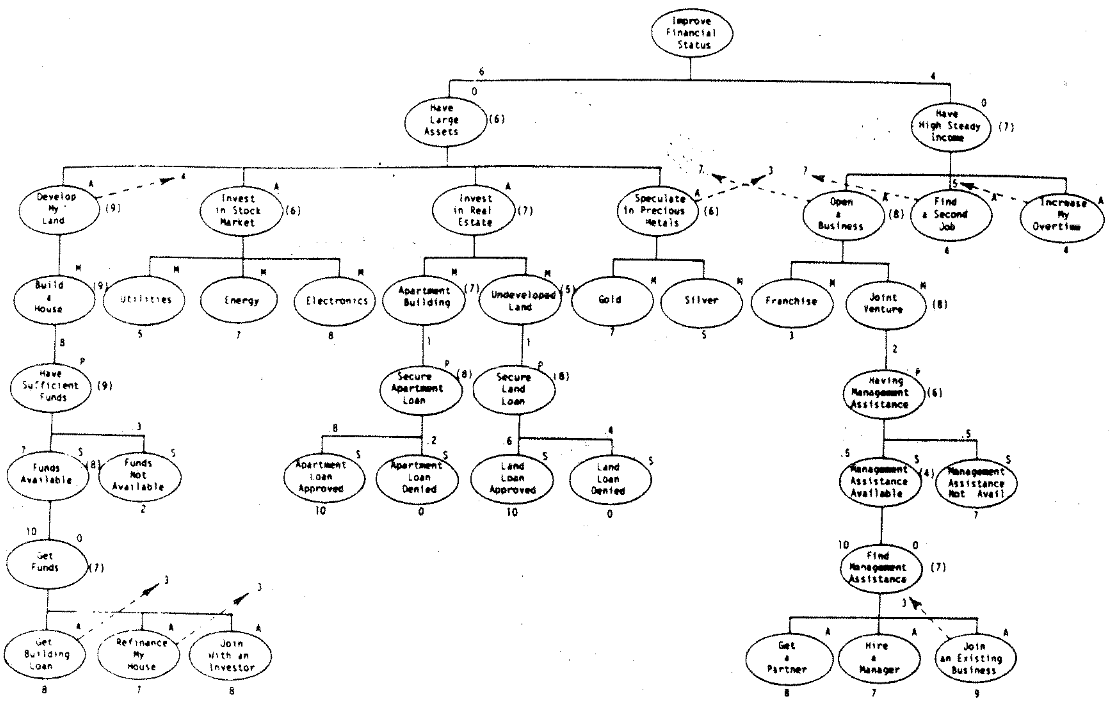

Fig. 4. Graphic representation of problem structure elicited from the dialogue.

— — — — 2) Secure apartment loan 8 1
State (level 5)
— — — — — 3) Apartment loan approved 10 0.8
— — — — — 4) Apartment loan denied 0 0.2
— — — 6) Invest in real estate—Undeveloped land 5 0
Precondition (level 4)
— — — — 3) Secure land loan 8 1
State (level 5)
— — — — — 5) Land loan approved 10 0.6
— — — — — 6) Land loan denied 0 0.4
— — 4) Speculate in precious metals 6 0
Action Mode (level 3)
— — — 7) Speculate in precious metals—Gold 7 0
— — — 8) Speculate in precious metals—Silver 5 0
— 2) Have high steady income 7 4
Action Strategy (level 2)
— — 5) Open a business 8 0
Action Mode (level 3)
— — — 9) Open a business—Franchise 3 0
— — — 10) Open a business—Joint venture 8 0
Precondition (level 4)
— — — — 4) Having management assistance 6 2
State (level 5)
— — — — — 7) Assistance available 4 0.5
Objective (level 1)
— — — — — — 4) Find management assistance 7 10
Action Strategy (level 2)
— — — — — — — 11) Get a partner 8 0
— — — — — — — 12) Hire a manager 7 0
— — — — — — — 13) Join an existing business 9 0
— — — — — 8) Assistance not available 7 0.5
— — 6) Find a second job 4 0
— — 7) Increase my overtime 4 0

Side Effects:

(Action number, Affected objective number, Adverse effect)

1	2	0.625
4	2	0.75
5	1	0.375
6	1	0.375
7	1	0.5
8	2	0.75
9	2	0.75
13	1	0.75.

System Recommendation

The goal of "Improve my financial status" can be attained to level of 4.55 if the following actions are taken.

Implement "Invest in stock market–Electronics" toward the objective "Have large assets."

Simultaneously, implement "Get a partner" toward "Find management assistance" which will eventually facilitate implementation of "Open a business–Joint venture" leading to attainment of "Have high steady income."

VII. Evaluation and Future Prospects

In this phase of the research we have made no special effort to equip the program's queries with a more natural "flair," which may partially account for the repetitive, mechanical style displayed by the present form of the program. A significantly more human style of conversation can, for example, be obtained by a random selection of synonymous phrases to avoid repetition (see [6]) and by exposing the queries' purpose, e.g., "It is crucial that we first examine ways of achieving '*X*'" or "I am trying to find out whether you foresee any special difficulties in executing '*Y*'," etc. Simple language-analysis features such as syntactic transformations, word matching, and key-word control would also greatly enhance the natural flavor of the dialogue style, but may on occasion lead to grossly incorrect phrases.

A more drastic leap toward natural discourse can, of course, be achieved by equipping GODDESS with some rudimentary knowledge about the domain of discourse. For example, basic knowledge of real estate relationships could assist GODDESS in producing the phrase "Let us consider the option of investing in real estate by purchasing an apartment building," instead of the awkward concatenation, "Invest in real estate–apartment building" used by GODDESS. However, our primary commitment in this project has been to construct and explore a totally domain-independent system. We believe that the weakness of GODDESS' style of discourse is a small price to pay for the benefit of using a single program to assist any advice seeker, from a real estate investor to a National Policy Planner.

Several observers of GODDESS have also commented that they sometimes feel uncomfortable assigning numerical values to the judgments requested, and that they occasionally feel unsure of what these numerical values represent or how to calculate them. The current system is equipped with several instructural features which can provide, upon request, a more detailed explanation of the nature of the assessment requested. Part of the "assessment discomfort" can be alleviated by improving these features, and part would be remedied when the user is asked to provide not a single number but a range of possible values.

However, we attribute the basic difficulty connected with assessing levels of attainment and strengths of influences to the fact that in everyday discourse these same concepts and relations are communicated in qualitative, nonnumeric castings. Not too long ago, before the general public became accustomed to numerical broadcasting of weather-predictions and accident statistics, the quantification of likelihood judgments (i.e., probability) met with similar resistance and uneasiness. We also found that after several days of working with the system, users saw no difficulty in interpreting and performing the assessments required. Consequently, we hope that the decision makers who could benefit from frequent consultation with such support systems will quickly become familiar with its somewhat nontraditional parameters.

For the occasional, inexperienced, and nontechnical users, we are currently examining a more drastic but more promising solution: *disposing of numerical estimates altogether.* Most of human knowledge and skills are acquired via nonnumerical media. Most training manuals and committee's reports convey useful information in purely linguistic terms. We read a newspaper article and feel very comfortable with statements such as "This vote by Congress would substantially impair the President's bargaining power." Although phrased qualitatively, we do acknowledge that such a statement conveys important and useful factual information without insisting on numerical explication of the degree of impairment. Similarly, it would be more natural and comfortable for the common decision maker to respond to queries such as the following:

Computer: "Is this condition absolutely necessary for action *X* or just desirable?" or

Computer: "Is it very likely or just probable? Choose the most appropriate term.

remotely possible, possible, probable, quite probable, likely, very likely, almost sure, . . ."

Behind the scenes, the program can map the user's linguistic response onto an appropriate numerical scale and propagate the resulting value through the graph by the methods described in Section IV. The user, however, will be spared the labor of quantifying inherently linguistic variables and the guilt associated with issuing uncertain estimates.

This approach will undoubtedly raise objections from traditional analysts who may view the reliance on linguistic, rather than numerical, inputs as a regression toward the prescientific era of speculative alchemy and seat-of-the-pants decision making. However, the ultimate objective of decision analysis is to provide both formal and *valid* representations of the decision maker's experience. Forcing a person to produce numbers would not by itself make the representation more valid, especially when one's experience is encoded qualitatively. A more reasonable approach would be to incorporate into the formal model as many of these qualitative relations as possible, so as to make the end results insensitive to the exact magnitude assigned to each relation. We believe that the goal-directed structure is a step in this direction; it is made up of many detailed and cognitively clear relationships which render the exact quantification of each component less critical. We feel, for instance, that the statement, "Noise level and safety are two factors of roughly equal importance," conveys more reliable information than any reasonable numerical response to the query: "How many people seriously injured or killed per year, call that number x, make you indifferent between the option: [x injured or killed and 2500 persons subjected to high noise levels] and the option: [one person injured or killed and 1 500 000 subjected to high noise levels]?" (Slovic *et al.* [14], quotation from Keeney's analysis of "The Mexico City Airport").

Succinctly, our basic position on this issue can be summarized by the belief that qualitative relationships of many cognitively meaningful concepts can be made to produce more accurate results than can numerical quantification of few cognitively unmanageable relationships.

Although we have not performed systematic experiments for evaluating the merit of GODDESS it appears that the goal-directed structure offers several advantages over the traditional decision tree approach. Our personal experiences with the two types of decision support systems confirm earlier expectations that the goal-directed approach would offer superiority in both *clarity* and *purposefulness.*

We find it clear, natural, and pleasing to talk about the need to obtain a loan in order to build a house, to quantify the degree of this need, or to express directly the fact that refinancing a house would diminish one's spendable income. These options of expression are simply not provided by the decision tree approach, where only action-sequences and world-states are considered, while conditions, issues, and factors remain tacit.

Similarly, we have on several occasions noticed that the explicit mention of an objective by the program focuses the attention of the user on a host of related experiences and evokes a number of unconventional alternatives capable of realizing that objective. For example, the idea of refinancing one's house and using the funds to develop one's land is very common to anyone with a little experience in real estate. However, to a user with no previous exposure to real estate manuevers, this possibility either may not occur, or in the more common case, the prospects of entering into debts may be discarded from conscious attention by virtue of emotional barriers or unpleasant associations it may carry. The goal-directed method weakens the impact of such barriers by focusing on a single objective at any given time and instructing the user to ignore, for the moment, all side effects. It should be very hard for the user responding to the query: "List all possible action strategies that you can take toward the fulfillment of 'Get funds'" *not* to mention the possibility "Refinance my house," regardless of the adverse implications that such an alternative may carry.

Recent experiments by Pitz, Sachs, and Heerboth [12] confirm our belief in the potential of GODDESS to encourage the discovery of novel alternatives. Of several candidate procedures tested for evoking a wider variety of choices, the one based on subgoal elicitation was found to be most effective. Additional experiments undertaken at our laboratory further show that goal-directed structuring, more than decision-tree structuring, encourages subjects to deviate from habitual patterns of behavior [11].

Based on these preliminary results and observations, we cannot rule out the prospect that the goal-directed structure described in this report will develop into the standard architecture for next generation decision-support systems. It provides a personal consultation to the casual user in a user-defined terminology and retains a formal documentation of the planning process. It is capable of operating in a fully computerized mode as well in a "analyst's apprentice" capacity, wherein the program would merely assist a traditional decision-analyst in selecting the next issue to be discussed. Finally, it is conceptually appealing and permits both systematic and directional acquisition of knowledge.

References

[1] L. S. Armstrong, W. G. Denniston, Sr., and M. M. Gordon, "The use of the decomposition principle in making judgements," *Org. Behavior Human Performance*, vol. 14, pp. 257–263, 1975.

[2] G. W. Ernst and A. Newell, *GPS: A Case Study in Generality and Problem Solving.* New York: Academic, 1969.

[3] R. E. Fikes and N. J. Nilsson, "STRIPS: A new approach to the application of theorem proving to problem solving," *Artificial Intell.*, vol. 2, pp. 184–208, 1971.

[4] C. F. Gettys, C. Michel, J. H. Steiger, C. W. Kelly, III, and C. R. Peterson, "Multiple-stage probabilistic information processing," *Org. Behavior Human Performance*, vol. 10, pp. 379–387, 1973.

[5] A. Leal, "An interactive program for conversational elicitation of decision structures," Ph.D. dissertation, Sch. Eng. Appl. Sci., Univ. of California, Los Angeles, UCLA-ENG-REP-7666, June 1976.

[6] A. Leal and J. Pearl, "An interactive program for conversational elicitation of decision structures," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 368–376, May 1977.

[7] A. Leal, S. Levin, S. Johnston, M. Agmon, and G. Weltman, "An interactive computer aiding system for group decision making," Perceptronics, Woodland Hills, CA, Tech. Rep. PQTR-1046-78-2, Feb. 1978.

[8] M. W. Merkhofer, A. C. Miller, III, B. E. Robinson, and R. J. Korsan, "Decision structuring aid: Characterization and preliminary implementation," Stanford Res. Inst., Menlo Park, CA, Sept. 1977.

[9] N. J. Nilsson, *Problem Solving Methods in Artificial Intelligence.* New York: McGraw-Hill, 1971.

[10] J. Pearl, "A goal-directed approach to structuring decision problems," Sch. Eng. Appl. Sci., Univ. of California, Los Angeles, UCLA-ENG-7811, Feb. 1978.

[11] J. Pearl, J. Kim, and R. Fiske, "An experimental evaluation of the effectiveness of two decision structuring methods," UCLA-ENG-81-36, Oct. 1981.

[12] G. F. Pitz, N. J. Sachs, and J. Heerboth, "Procedures of eliciting choices in the analysis of individual decisions," *Org. Behavior Human Performance*, vol. 26, pp. 396–408, 1980.

[13] J. Saleh, "IDEA: An issue-driven elicitation algorithm for decision problem structuring," Ph.D. dissertation, Sch. Eng. Appl. Sci., Univ. of California, Los Angeles, June 1980.