Described is an experimental system that enables the user, through an intelligent graphics terminal, to construct, modify, analyze, and store decision trees. With this system, business decisions under uncertainty can be analyzed. This paper discusses the system and its capabilities. Included is a brief discussion of decision analysis, which represents an aspect of financial modeling.

An interactive graphics system for analysis of business decisions

by J. Ravin and M. Schatzoff

Decision tree analysis is a formal methodology by which complex decision problems can be decomposed into sequences of contemplated decisions (or acts) and their uncertain consequences (or events). The analysis is naturally and conveniently depicted in the form of a tree, where the branches emanating from a given node represent alternative acts or events. Anticipated cash flows and probabilities are associated with appropriate branches of the tree, and an optimal strategy may be defined as that path that maximizes expected monetary gain. The concept of risk aversion may be introduced by means of a preference function, which quantitatively represents the decision-maker's attitudes toward risks. These ideas will be explored in greater depth and clarified by example in ensuing sections.

Much of the original research and development of methodology for dealing analytically with problems of decision making under uncertainty has taken place at the Harvard Business School over the past dozen years under the leadership of Professors Howard Raiffa and Robert Schlaifer.[1-5] A pair of interesting expository articles by John Magee on the application of decision analysis to business problems appeared in the *Harvard Business Review*.[6,7] Other terminology that has been employed to describe this approach includes: Statistical Decision Theory, Analysis of Decisions under Uncertainty, and Bayesian Decision Theory.

The principal thrust of this paper is to describe the design features and functional capabilities of the decision tree graphics system, an experimental system for interactively constructing, modifying, and analyzing decision trees at a graphics terminal. Computer graphics provides a natural and convenient medium for decision analysis, since the tree diagram can be displayed on the screen of a graphics display device, and the tree structure itself can be easily manipulated by pointing to specific branches and nodes. This mode of operation should be contrasted to a nongraphics approach that would require a user to locate a particular node or branch by searching a list (or lists) of identifiers. An added advantage of the graphics approach is the interactive capability that allows a user to carry out sensitivity analyses easily and rapidly. For example, a user may wish to examine the implications of changes in his basic assumptions concerning prior probabilities of uncertain events, cash flows associated with particular acts or events, or the structure of the decision tree itself. The ability to interact with the computer by merely pointing to displayed objects with a light pen to change data values or structure, and reevaluating instantaneously renders complex sensitivity analyses almost trivial. These capabilities are of particular importance to the basically nontechnical class of users at which such a system is aimed, namely, executives and their staffs and business school students.

The following section provides an elementary discussion of the principal concepts underlying decision tree analysis. The reader who is already familiar with this methodology may wish to proceed directly to the succeeding sections that describe the functional capabilities of the decision tree graphics system and the design and implementation considerations.

Decision analysis

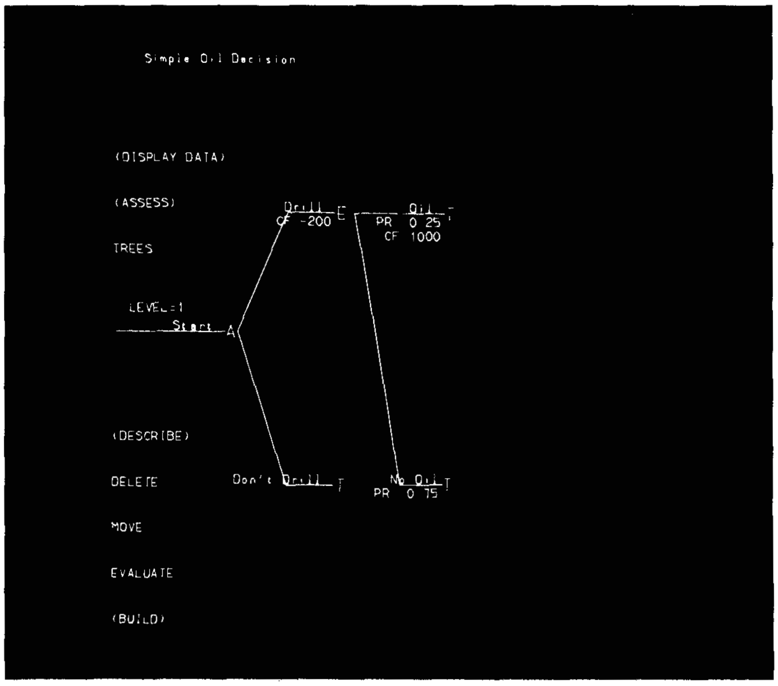
discrete problems

Virtually all important decisions are made under *uncertainty*, for it is impossible to predict the future with certainty and one can never know exactly what relevant events will occur subsequent to a particular choice or action. The responsible business executive is continually called upon to make decisions under such circumstances, and must choose a definite course of *action* among those available to him. Even not acting in a given situation represents a decision that may have uncertain consequences associated with it. The desirability of each alternative course of action contemplated by the decision maker may depend upon the *chances* that various possible events will occur subsequent to the act, the *consequences* that will result if certain events do occur, and the *desirability* (or undesirability) of such consequences. The decision maker must consider the above factors, at least implicitly, in arriving at any given decision. The aim of decision analysis is to systematically decompose the decision problem into its constituent parts, quantify the uncertainties and consequences associated with various elements of the problem, and evaluate in a logical and consistent manner the implications of one set of factors at a time. It provides a means for the decision maker to consider *explicitly* those factors that he must, of necessity, consider at least implicitly in arriving at any decision.

Let us consider a simple example to illustrate the basic ideas discussed above. Suppose that an oil company owns drilling rights to a particular parcel of land and the company must decide whether or not it should invest in undertaking a full-fledged drilling operation on that land. The decision depends upon the size of the required investment and the likely value of any oil that may be produced. A very simple representation of this problem is provided by the decision tree shown in Figure 1. The tree starts at an act node A with a choice of two acts, "drill" or "do not drill." The cost of drilling is assumed to be \$200,000, and since costs represent negative cash flows, the branch labeled drill has assigned to it a cash flow (CF) of −200. (All cash flows are shown herein as multiples of \$1000.) It is estimated that the chances of finding oil are about one in four (PR 0.25), and that if oil is found, it will be worth \$1,000,000. This information is displayed on the branches of the fork emanating from the event node E. Such estimates would most likely be provided by a geologist, based upon his knowledge of the geography and physical characteristics of the site. Note that even in the absence of this formal structuring of the problem, any prudent businessman confronted with this decision problem would have to consider all of these same factors, i.e., the costs, chances of success, and probable returns.

The notion of numerically assessing the probability of an uncertain event is one with which the decision maker may feel a bit uncomfortable, for the event of finding oil at the site is dissimilar in nature to the outcome of flipping a coin. If the coin is perfectly balanced, the frequency interpretation of probability theory asserts that the proportion of heads will approach one half as the number of flips increases. On a single flip, the result is either a head or a tail, and we resort to the frequency interpretation of probability theory to ascribe one half as the probability of a head on a single flip. In the oil drilling example, we do not have an experiment that can be repeated many times, but we assert that an expert's experience, based as it is on the outcome of many similar experiences, can be used to assess a subjective probability for the outcome of the drilling operation. The actual probability of oil existing is, of course, either one or zero, since oil either is or is not to be discovered at the site. However, until such determination can be made, the decision maker must have some rational basis for assessing the risks associated with each of the actions that he might decide to take. The numerical probabilities that he assigns to the uncertain events represent his *subjective beliefs* about the chances that these events will occur. If he were a betting man, the decision maker of our example would be indifferent between taking a three-to-one bet that there is oil, and

Figure 1 Diagram of oil decision problem



giving the same odds on the opposite proposition. To carry the argument one step further, he should be indifferent to the first gamble and one in which he receives three to one odds that a random selection of one ball from a bag known to contain one white and three red balls will produce the white ball.

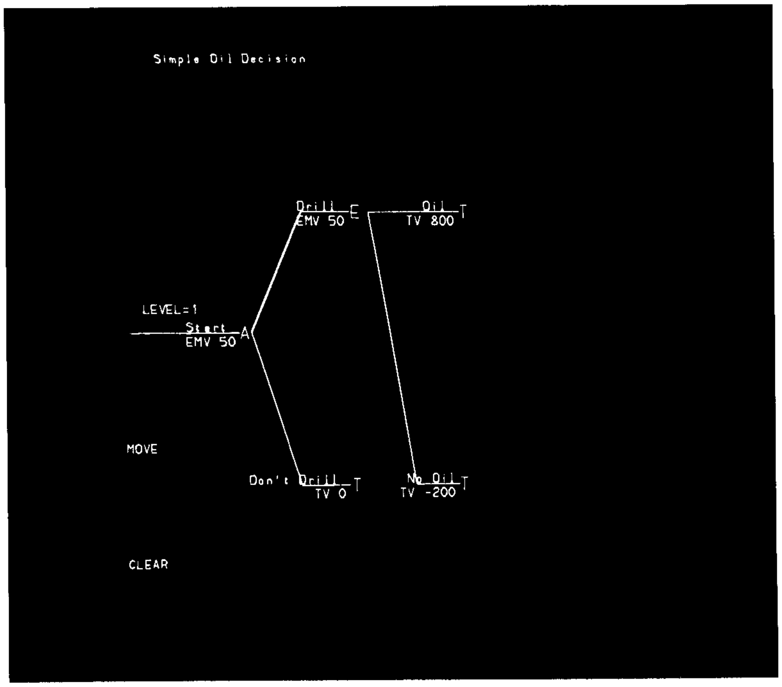
The formal analysis of our simple example begins by calculating the total cash flow or terminal value (TV) at each terminal node, as shown in Figure 2. The expected monetary value (EMV) of the decision to drill is $50,000, computed by averaging the terminal values:

$$\text{EMV} = (800) \times (0.25) + (-200) \times (0.75) = 50$$

Thus, even though there is a high chance (0.75) of losing $200,000, the large gain ($800,000) that can be realized if oil is found – an event of probability 0.25 – is sufficient to induce the decision maker to drill, since his net expected gain is $50,000. This example illustrates the basic algorithm, sometimes called "averaging out and folding back," used to evaluate decision trees. Starting at the terminal nodes of the tree, one works backwards, computing nodal values as follows:

1. The value of an event node is the expected (or average) value of the nodes corresponding to the branches emanating from the event node in question. It is computed by $\Sigma V_i P_i$,

Figure 2 Evaluation of oil decision problem



where V_i and P_i are the value and probability associated with the ith such branch and node.

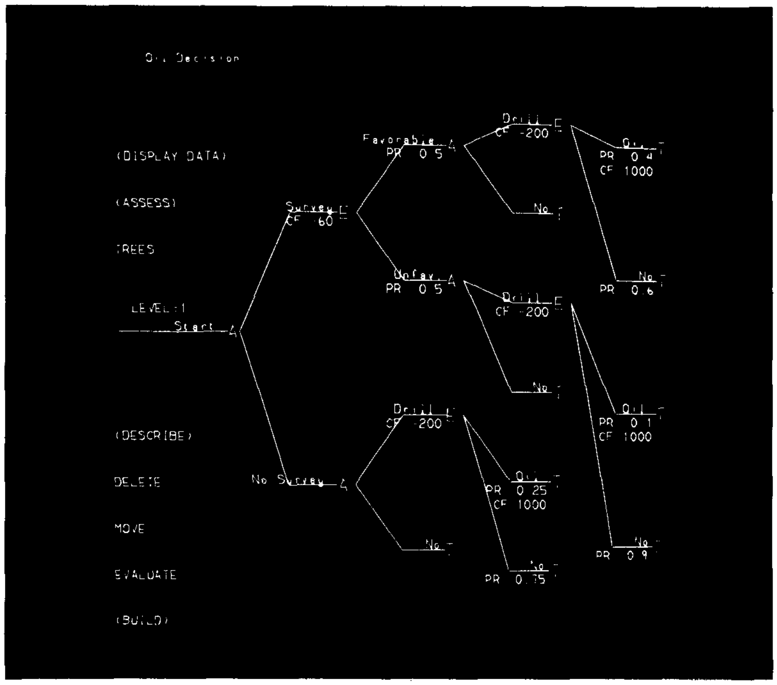
2. The value of an act node is the maximum of the values of the nodes corresponding to the branches emanating from the act node in question.

added factor

To further illustrate the nature of decision analysis, let us complicate the problem in a simple, but realistic, way by allowing for the possibility of a geological survey at a cost of $60,000, prior to deciding whether or not to proceed with the drilling operation. The revised problem is diagrammed in Figure 3. Note that the subjective, or prior probability, of finding oil, is dependent upon the results of the geological survey. That is, a favorable survey would lead the decision maker to increase his assessment of the likelihood of striking oil, whereas a negative result would have the opposite effect. Thus, the probability of oil existing, given the result of the survey, is a *conditional* probability. The survey merely provides information that modifies the decision maker's belief about the existence of oil. However, the *marginal* probability of oil existing cannot be affected by the survey, and this can be verified by a simple computation.

$$\begin{aligned}\text{Pr (oil)} &= \text{Pr (Oil} \mid \text{Favorable Survey)} \times \text{Pr (Favorable Survey)} \\ &\quad + \text{Pr (Oil} \mid \text{Unfavorable Survey)} \times \text{Pr (Unfavorable Survey)} \\ &= (0.4) \times (0.5) + (0.1) \times (0.5) \\ &= 0.25\end{aligned}$$

Figure 3 Revised oil decision problem



In the above formula, the notation Pr $(A|B)$ is read as "Probability of the event A, given the event B." It should be noted that it might be more convenient in a practical situation to assess the marginal probability of the second event and the conditional probability of the first event, given the second event. For example, in our oil-drilling problem, it might be natural for the geologist to assess the marginal probability of oil existing and the conditional probabilities of a favorable survey, given that oil did or did not exist at the site. There is then the problem of converting from the stated marginal and conditional probabilities to those needed for the analysis. This is accomplished by means of Bayes' Theorem; hence the use of the terminology "Bayesian Decision Theory." The computation is given by the formula

$$\Pr(A|B) = \Pr(B|A) \times [\Pr(A) / \Pr(B)]$$

which is readily illustrated by Table 1. Suppose that the geologist had quoted the following probabilities:

Pr (Oil) = 0.25
Pr (Favorable Survey|Oil) = 0.80
Pr (Favorable Survey|No Oil) = 0.40

This information is sufficient to construct Table 1. For example, the joint probability of a favorable survey and oil existing is given by the product Pr (Oil) × Pr (Favorable Survey|Oil) = 0.20, and the conditional probability of oil existing, given a favorable survey, is calculated from Bayes' formula as

Table 1. Conversion of probabilities

	Favorable survey	Unfavorable survey	
Oil	0.20	0.05	0.25
No oil	0.30	0.45	0.75
	0.50	0.50	1.00

$$\begin{aligned}\text{Pr (Oil}|\text{Favorable Survey)} &= \text{Pr (Favorable Survey}|\text{Oil)} \times \text{Pr} \\ &\quad \text{(Oil)/Pr (Favorable Survey)} \\ &= 0.40\end{aligned}$$

The evaluation of the revised oil decision problem shown in Figure 4 is obtained by "averaging out and folding back" as before. The optimal strategy, as before, is to drill without taking a survey. The interested reader can readily check that if the cost of the survey were \$40,000 instead of \$60,000, the optimal strategy would have been to take a survey and then drill if and only if the result were favorable, since this strategy would have an expected value of \$60,000.

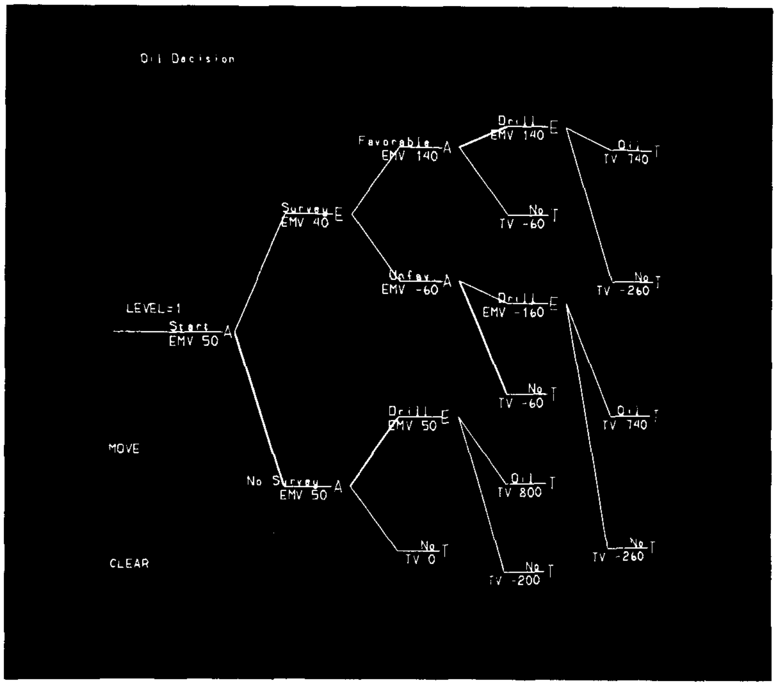
The worth of the survey may be seen to be \$50,000 for at this price the decision maker would be indifferent to acting directly or taking a survey and then acting—either strategy has an expected value of exactly \$50,000. It would be advantageous to take the survey at any cost below \$50,000 and disadvantageous to take it at any higher figure.

risks

Our discussion thus far has assumed that the decision maker desires to act in such a manner as to maximize expected monetary value. This criterion may not appeal to many businessmen since it fails to provide protection against possible large losses. In our example, for instance, we see that although the expected monetary value of the decision to drill is \$50,000, a high probability (0.75) exists of losing \$200,000. A conservative businessman might prefer to sell his drilling rights for a certain amount of money less than \$50,000 to avert the risk of losing \$200,000. We would say that such an individual is *risk averse*, and define the difference between the expected monetary value of the drilling rights and the amount for which he would sell it as a *risk premium*. An obvious analogy (in insurance parlance) is the insurance premium one would be willing to pay to avert a calamitous loss.

To integrate the decision-maker's attitudes toward risks into the overall analytic framework of decision analysis, he must stipu-

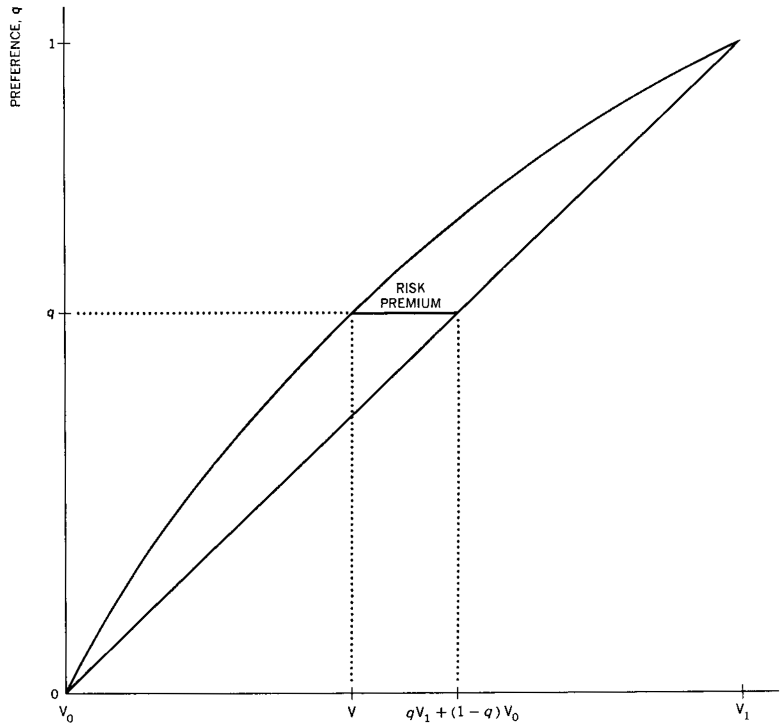
Figure 4 Evaluation of the revised oil decision problem



late the risk premium he would be willing to pay for every possible gamble. This specification is accomplished by means of a *preference* (or *utility*) function, which is defined over the range of all possible monetary values that may be realized in the given problem, and which is normalized to take values between zero and one. If the decision maker is indifferent to receiving the amount V with certainty, and receiving the amounts V_1 or V_0 with probabilities q or $1 - q$, respectively, ($V_1 \leq V \leq V_0$, $0 \leq q \leq 1$), we define q to be his *preference* for V relative to V_1 and V_0. The amount V that he would just be willing to accept in place of the above-referenced gamble is defined as his *certainty equivalent* (CE) for the gamble. The difference between the expected value of the gamble, $qV_1 + (1 - q)\, V_0$, and its certainty equivalent V, is then the *risk premium*. These concepts are illustrated in Figure 5.

A particular example of a preference function, which is incorporated in the decision tree graphics system, is that of constant proportional risk aversion. A person is said to have *positive decreasing risk aversion* if his risk premium is always positive and if it decreases with the addition of the same positive constant to both V_1 and V_0. A particular example of positive decreasing risk aversion is that of *constant proportional risk aversion*, where the individual would always pay the same fraction of his assets to

Figure 5 Preference function



insure against a fixed chance of losing a specified fraction of those assets. Such a function must be of the form

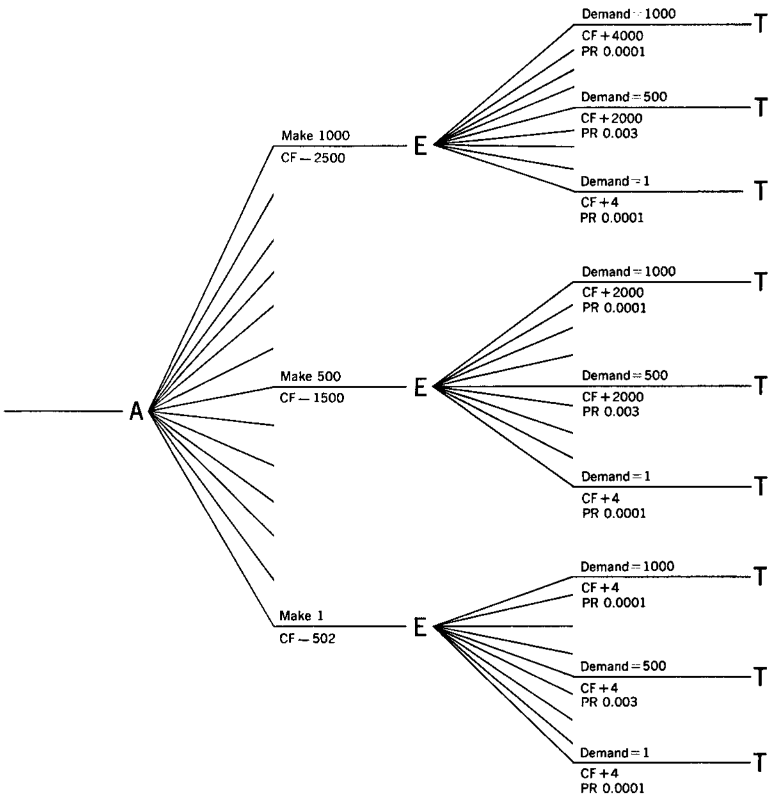
$$q(v;r) = \frac{1}{r}\,(1 - v^{-r}),\ v > 0$$

or a linear transformation thereof.[8]

If the decision maker regards this as a reasonable means of expressing his preferences for various risks, he can assess the function by specifying the reference values V_1 and V_0, the probability (q) of V_1, and the certainty equivalent V. Of course, many other kinds of preference functions may be employed. This particular one has been chosen to illustrate the concept of using preference functions in decision analysis.

The analysis of decision trees using preferences begins by replacing the calculated end-position values with the decision maker's preferences for those values, as determined from his preference function. From that point, the averaging out and folding back procedure is applied to the preferences, rather than to the monetary values. At any node in the tree, the *certainty equivalent* of the preference value may be computed from the inverse preference function. The overall effect of employing a

Figure 6 XYZ Corporation's decision problem, discrete version

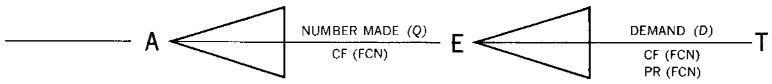


positive risk-averse preference function is that it tends to lead the decision maker away from acts that may result in large losses.

continuous problems

Although a large class of problems can be readily handled by the type of discrete analysis discussed above, many practical decision problems cannot be formulated conveniently without resorting to continuous representations of acts and/or events. For example, suppose that the XYZ Corporation manufactures valves. In a given time period, XYZ may produce up to 1000 valves at a fixed set-up cost of \$500, plus a variable cost of \$2 per valve. Let us assume that the going market price is \$4 per valve, and that XYZ's production is insufficient to affect the market price. Thus, XYZ can safely assume that if it manufactures Q valves in the face of a demand for D valves, its income for the period will be $\$4 \times Q$ (if Q is less than D), or $\$4 \times D$ (if D is less than Q). A strictly discrete representation (see Figure 6) of this problem would result in one million terminal nodes (i.e., 1000 values of demand corresponding to each of 1000 levels of production). It is obviously absurd to attempt to analyze this problem in this manner.

Figure 7 XYZ Corporation's decision problem, continuous version



Suppose, however, we consider the production (act) and demand (event) forks as essentially *continuous* objects, which we shall call act and event fans, respectively. Then we may depict the megabranched tree of Figure 6 very simply as in Figure 7, where the special symbol $\lessdot$ is a reminder that these substructures are fans representing many-branched forks.

The symbolic nature of a fan precludes the direct entry of cash flows and probabilities for each possible value of production and demand; rather, the symbolic decision diagram requires that cash flow *functions*, and probability *distributions*, be defined on the appropriate fans, each of which is represented by a specified variable.

In our example, if Q is the quantity manufactured and D the quantity demanded, then the production cost to the manufacturer is $-(500 + 2 \times Q)$ and the sales revenue is $4 \times \text{MIN } (D,Q)$.

In addition, we must specify a probability distribution for the demand D to perform the averaging out and folding back procedure; the initial act we are seeking to choose, of course, is the best possible choice for Q, the quantity to be produced.

These capabilities for incorporating fan structures and their associated cash flow and probability functions allow the user to tackle very large and complicated decision problems with relative ease.

Functional capabilities of the graphics system

From the examples presented in the previous section, it is easily seen that a computer-based decision tree graphics system should allow the user to:

1. Construct arbitrary tree structures.
2. Assign data values and descriptors (cash flows, probabilities, and labels) to each branch.
3. Modify the tree structure and its associated data.
4. Evaluate strategies (by EMV or preference).

5. Move the displayed image so as to display sections of trees that cannot be accommodated on the screen in their entirety.

From a human factors standpoint, the user interface must provide easily used facilities for carrying out these capabilities. This is accomplished in the decision tree graphics system by means of eight basic functions that can be independently invoked by light pen to enable the user to effect the five basic capabilities mentioned above. These eight functions are:

1. TREES – Permits retrieval of a stored tree, or definition of a new tree.
2. BUILD – Used to add structure to a node.
3. DELETE – Allows the user to remove portions of the tree.
4. DESCRIBE – Allows the user to associate data (cash flows, functions, probabilities, distributions, labels, etc.) with the branches and nodes.
5. EVALUATE – Allows the user to identify optimal paths and subpaths, together with computed node values.
6. MOVE – Enables the user to display different sections of a large tree by moving the viewing "window" over the tree.
7. DISPLAY DATA – Allows selective display of data that cannot be conveniently displayed as part of the tree itself, e.g., cash flow functions and nodes of probability distributions.
8. ASSESS – The user may assess preference functions or probability distributions.

Figure 8 Menu item command hierarchy

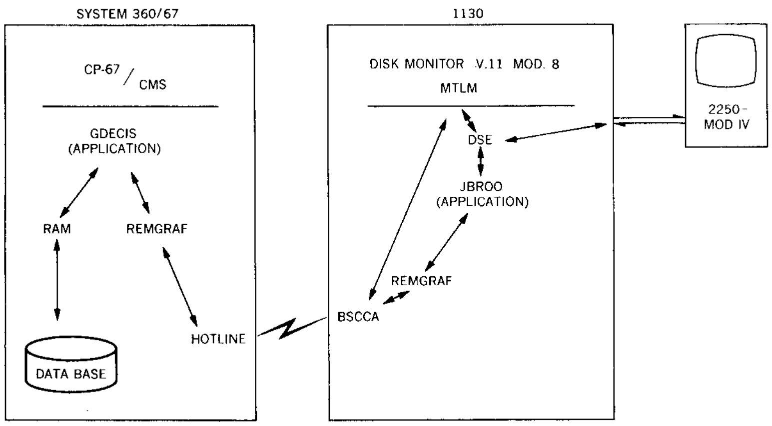
Main Menu	Submenus
(DISPLAY DATA)	(BACK) PROBABILITY NODE CASH FLOW PATH CASH FLOW RANGE
(ASSESS)	PREFERENCE FUNCTION PROBABILITY DISTRIBUTION
TREES	
(DESCRIBE)	(BACK) CASH FLOW PROBABILITY LABEL RANGE CHANGE TYPE
EVALUATE	
DELETE	
MOVE	
(BUILD)	(BACK) COPY BRANCH FORK FAN

Specific details of the use of each of these basic commands are given in the Appendix. Figure 8 depicts the menu-item structure of the decision tree graphics system. Note that all items appearing in parentheses in Figure 8 imply direct changes of menu when invoked. The submenus themselves are used to avoid a proliferation of functions at one level. The command (BACK), appearing under the (BUILD), (DESCRIBE), and (DISPLAY DATA) commands, allows the user to return to the main menu. Prompting messages are displayed at the top of the screen whenever a menu item requiring user action, such as MOVE or FORK is invoked. Thus, the user does not have to learn a great deal about the conventions for using the system, but rather can concentrate on solving problems.

System design and implementation

The decision tree graphics system was designed with two principal objectives in mind: first, to study the feasibility of using such a system for teaching, research, and the solution of actual decision problems, and second, to demonstrate the integration of

Figure 9 System hardware and software configuration



several experimental software subsystems written at the Cambridge Scientific Center into a single meaningful application. The system configuration is shown in Figure 9.

The display hardware used in the graphics system is an IBM 2250 Model IV display unit driven by an IBM 1130 computer. This subsystem is used as an intelligent graphics terminal for a CMS virtual machine running under CP-67.[9] It provides a message-switching capability by light-pen detection of displayed objects such as menu items and tree branches.

The application program for interpreting menu-item detects and sending messages is written in 1130 Assembler Language, and uses a display system executive package written at the Cambridge Scientific Center[10] to manage the graphic data base for the display terminal. Assembler Language allows access to an experimental subsystem called the Multi-Tasking Line Monitor (MTLM),[10] which provides the multi-tasking capability that the 1130 computer needs to support both graphics and teleprocessing. The teleprocessing capability is provided through an experimental program system called BSCCA/HOTLINE,[10] which allows the 1130 computer to communicate with a CMS virtual machine.

Thus, both the terminal system (through the MTLM) and the remote System/360 Model 67 (via CP-67) are time-shared and compute asynchronously in this application. The larger computer is used to perform the fundamental analysis and data management function for the decision tree graphics system and also to generate the topological structure of the display. Although it would be possible to implement a system like the one we are discussing on the 1130–2250 combination alone, the limited storage capacity and computational speed would be limiting factors.

The System/360 application program is written in PL/I because of the mixed computational/data-processing nature of the problem. It uses the RAM data base management system,[11] an experimental system based on the ideas of Feldman and Rovner,[12] to store and retrieve trees. This system was used because of its *relational* nature; the programmer need not store directly any pointer structure or other detailed low-level information about the data. The RAM system acts as a software associative memory, so that, for example, the programmer may ask for the successors of a given node directly, and RAM automatically returns the identifiers of nodes that fulfill the request. Directly callable from PL/I, it relieves the programmer of the detailed problems of data base management.

The current decision tree graphics system data base allows for any number of trees, up to a combined total of about 3900 nodes. The actual driving element in the system, as in all interactive graphics applications, is the light pen of the graphics display terminal; input of data (textual or numeric) is via the keyboard of the display terminal.

Summary

An experimental interactive graphics system for the analysis of decisions under uncertainty, using the "decision tree" technique, has been designed and implemented using an 1130–2250 combination as an intelligent graphics terminal communicating with a CMS virtual machine in a CP-67 environment. It provides the user with a highly interactive, easy-to-use facility for constructing, modifying, and evaluating complex decision trees, including capabilities for assessing and storing preference functions and probability distributions, and representation of continuous act and event nodes by fans (specialized branches). The methodology employed throughout is essentially that described by Schlaifer in his latest book.[5]

ACKNOWLEDGMENTS

The authors wish to express their gratitude to Professors Robert Glauber, Arthur Schleifer, and Robert Schlaifer of the Harvard Business School for their expert and valuable assistance in the design of the user interface and specification of the functional requirements for the decision tree graphics system.

CITED REFERENCES

1. R. O. Schlaifer, *Probability and Statistics for Business Decisions*, McGraw-Hill, New York, New York (1959).
2. H. Raiffa and R. O. Schlaifer, *Applied Statistical Decision Theory*, Division of Research, Graduate School of Business Administration, Harvard University, Boston, Massachusetts (1961).

3. J. W. Pratt, H. Raiffa, and R. O. Schlaifer, *Introduction to Statistical Decision Theory*, McGraw-Hill, New York, New York (1965).
4. H. Raiffa, *Decision Analysis*, Addison-Wesley, Reading, Massachusetts (1968).
5. R. O. Schlaifer, *Analysis of Decisions Under Uncertainty*, McGraw-Hill, New York, New York (1969).
6. J. F. Magee, "Decision trees for decision making," *Harvard Business Review* **42,** No. 4, 126–148 (July-August 1964).
7. J. F. Magee, "How to use decision trees in capital investment," *Harvard Businesss Review* **42,** No. 5, 79–96 (September-October 1964).
8. R. O. Schlaifer, *Computer Programs for Elementary Decision Analysis*, 157, Division of Research, Graduate School of Business Administration, Harvard University, Boston, Massachusetts (1971).
9. *CP-67/CMS Version 3.1 User's Guide*, Form No. GH20-0859, IBM Corporation, Data Processing Division, White Plains, New York.
10. These software subsystems are experimental programs written at the IBM Cambridge Scientific Center and are not available for distribution.
11. M.F.C. Crick et al., *A Data-Base System for Interactive Applications*, Report No. G320-2058, IBM Corporation, Scientific Center, Cambridge, Massachusetts.
12. J. A. Feldman and P. D. Rovner, "An Algol-based associative language," *Communications of the ACM* **12,** No. 8, 439–449 (August 1969).

Appendix

A detailed description of the system command structure follows:

1. TREES—This function enables the user to retrieve any stored tree from the data base, create a new tree, or terminate the current session with <END OF SESSION>. The light pen is used to point to the name of an existing tree, or (NEW TREE), in the first two cases. If an existing tree is selected for display, the detect causes the tree identifier (id) to be transmitted to a System/360; the System/360 retrieves the designated tree from the data base and generates the graphics display orders, which are then sent to an IBM 1130 computer.

The display is governed by these general rules: Up to five levels in the tree are displayed; at each level, up to 12 nodes (branches) are shown. If the tree is deeper than five levels, or has more than 12 branches at a given level, the existence of undisplayed structure is indicated by an asterisk after the appropriate node symbol. For example, the oil decision tree in Figure 3 is five levels deep; the first level has one branch, the second, two, the third, four, the fourth, six, and the fifth, four. Thus, the whole tree may be displayed in the window defined by the screen of the display device.

The display algorithm produces equally spaced branches at a given level, rather than trying to give a fixed or balanced format to the display. This allows the display of a maximum number of nodes with uniform legibility. By convention, when a tree is first called to the display screen, the left-most node displayed is the tree's root node, and LEVEL = 1 is displayed above this node.

Should the user choose to create a new tree, he is directed to enter a name for the tree at the display terminal keyboard (using the JUMP key to terminate the name). The system then creates the root node for him and displays it. The current tree name is always displayed in the upper left corner of the screen.

2. (BUILD) – This function allows the user to add a BRANCH, a FORK, or a FAN at any existing node, or to COPY a substructure from one node to another, by indicating the appropriate item in the submenu. The COPY command is particularly useful in constructing a large tree, since substructures of the tree tend to be repetitive in nature, as seen in Figure 3. The (BUILD) function is additive; adding a branch to an existing fork of three branches simply changes the fork to one of four branches. The user may thus easily make additions to an already constructed tree.

After the (BUILD) function is completed, an asterisk is displayed after the node symbol to which the structure was added, and the user is asked if he wants the change displayed. This is to avoid redisplaying the tree after each addition, which requires transmission of the entire display screen image from the System/360. If the user is making several additions at once, he may wish to display only the final representation, and thus avoid delays that would be incurred in retransmitting the display screen image after each individual change.

(BUILD) options

Following are options to the (BUILD) function.

(BACK) – Returns the system to the main menu.
BRANCH – Asks the user to point to the node to which the branch is to be added, and then the type of branch (ACT/EVENT/TERMINAL).
FORK – Same as BRANCH, except the user must first enter the number of branches to be generated by the system.
FAN – Same as BRANCH, except the user must also specify the name of the variable to be associated with the fan.
COPY – The system asks the user to point to the branch to be copied *from*, and then to the branch to be copied *to*.

An important feature here is the "HOOK," an imaginary node displayed in the upper left portion of the screen. It may be used as a temporary node to copy to and from when the two nodes involved in the COPY operation cannot be displayed at the same time. The HOOK is brightened when structure is copied to it. Such structure may then be copied directly to another node, usually after completion of a MOVE command, as described later. This feature is very handy in manipulating a large tree.

3. DELETE—An inverse, so to speak, of (BUILD); the user is asked to point to the node or branch from which structure is to be deleted. The system then displays the "pruned" tree and waits for the user to point to the (OKAY) command (in which case he is returned to the main menu) or to the (ABORT) command (in which case the tree is redisplayed with the deletion negated). This "second chance" feature is very helpful, especially in a large tree, in the event that an unintentional deletion has been attempted.

4. (DESCRIBE)—This invokes a submenu that the user may use to enter or change data in the tree. The changes are displayed immediately from the 1130 computer; no redisplay of the tree from the System/360 is necessitated by (DESCRIBE).

(DESCRIBE) options

Following are options to the (DESCRIBE) function.

(BACK)—Returns the decision tree graphics system to the main menu.
RANGE—The user is asked to point to a FAN, and then to enter the range for the FAN variable in the form < lower end > b < upper end >.
LABEL—The user is asked to point to a node and then enter a label.
CASH FLOW—The user is asked to point to a node and then enter a cash flow; the dollar sign is omitted. The number may be integer or decimal; the system displays only the integral dollar amount on redisplay of the tree. For fans, or successors of fans, the system asks for a cash flow function; any valid FORTRAN expression is accepted, as long as: (1) variables used are only those appearing as FAN variables along the path from the current node to the tree root, and (2) arithmetic library functions used in the expression are restricted to MIN, MAX, EXP, LOG, SQRT, and ABS.
PROBABILITY—The user is asked to point to a node and enter a probability. For an event FAN, the user is given the option of either choosing a distribution from all those currently stored or generating a new one. The procedure for constructing a probability distribution is described under item 8 (ASSESS).
CHANGE TYPE—The user may change any node in the tree from ACT to EVENT to TERMINAL, etc. This action is useful after a deletion.

5. EVALUATE—Determines the optimal strategy by means of the averaging out and folding back algorithm.

First, the user is asked if he wishes to SET a FAN variable. If he indicates YES, the SET/DONE/MOVE facility allows him to SET any FAN variable to any particular value within its range, or to MOVE the viewing window to find other FANs he may wish to

SET, or to tell the decision tree graphics system that he is DONE with this facility. At this point, or if he has indicated NO in response to the above inquiry, a list of all preference functions stored in the data base, along with < EXPECTED VALUE > and (NEW FUNCTION) is presented at the display screen. He then indicates the name of the preference function he wishes to use in the evaluation of the tree. (If he chooses (NEW FUNCTION), he is first asked to assess a new preference function to be used by EVALUATE.) The user is then asked to point to a node to fold back to, whereupon the decision tree graphics system evaluates the tree by Expected Monetary Value (EMV) (or preferences) and redisplays the tree, but with the EMVs (or expected preferences and their Certainty Equivalents (CE)) replacing the cash flows and probabilities. Also, at *each* act fork, the path corresponding to the best choice (in terms of EMV or CE) is brightened by the system, so that the user may readily see the optimal strategy. He may then MOVE or CLEAR the tree. The former (as described below) lets him follow strategies to the terminal branches of large trees; the latter clears the structure of calculated values and/or preferences, redisplays the tree with the cash flows and probabilities, and returns to the main menu.

6. MOVE—The user may move the viewing window represented by the display screen over the tree by using the MOVE main menu item. The system instructs him to point to the node to be moved and then to the node to which it is to be moved. This move is by LEVEL only; pointing to two nodes at the same LEVEL results in a null move. If a MOVE has been completed, the left-most branch will have LEVEL = N displayed above it, indicating the level of this node in the tree.

MOVE at the main menu level has an (ABORT)/(OKAY) feature similar to that of DELETE—and for similar reasons. (Note: This feature is not available for the MOVE subcommand provided under COPY and EVALUATE.)

7. (DISPLAY DATA)—Allows the user to display any of the data associated with any branch of the tree. This function is included because, for example, the character strings for cash flow functions could completely clutter the picture.

(DISPLAY DATA) options

Following are options to the (DISPLAY DATA) function.

(BACK)—As previously indicated with DESCRIBE options.
RANGE—Displays the range of the variable for the indicated fan.
NODE CASH FLOW—Displays the cash flow function or value associated with the indicated branch.
PATH CASH FLOW—Displays the total cash flow value or function for the path leading from the tree root to the indicated node.

PROBABILITY—Displays the probability value, or the name of the distribution associated with the indicated branch.

8. (ASSESS)—The user may assess or delete either a probability distribution or a preference function, and, in addition, he may also display a graph of the density and distribution functions for any currently stored probability distribution.

If he has chosen to assess a preference function, he is asked to enter a ruin point and a risk premium. The latter is elicted by the system, which requires him to indicate the amount he would accept for a 50-50 gamble on the extreme terminal cash flow values. The system then constructs a constant proportional risk averse preference function based on this data.

If he is assessing a probability distribution, he is asked to enter the number of points he wishes to specify on the cumulative distribution curve, the values of these points, and their corresponding fractiles. (These should include the zero and one fractiles.) The system then displays the distribution in the form of a quadratically smoothed cumulative function through the specified fractiles and the corresponding density function. If the shape of the distribution does not conform to the user's conception of what it should look like, he can erase it and construct a new one.

Hard copy of what appears on the display screen can be produced on the IBM 1627 plotter by use of a function key on the IBM 2250 display terminal.