AD-A092 129   DECISIONS AND DESIGNS INC MCLEAN VA   F/G 5/10

PILOT AUTOMATED INFLUENCE DIAGRAM DECISION AID.(U)

DEC 78 R J KORSAN, J E MATHESON   N00014-78-C-0100

UNCLASSIFIED   NL

1 OF 1

AD
A092129

END
DATE
FILMED
81-2-
DTIC

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

PILOT AUTOMATED INFLUENCE DIAGRAM DECISION AID

Final Report

December 1978

By: Robert J. Korsan
James E. Matheson

Prepared for:

Defense Advanced Research Projects Agency
Arlington, Virginia

Under Subcontract from Decisions and Designs, Inc.
McLean, Virginia

SRI Project 7078

50272-101

REPORT DOCUMENTATION PAGE

1. REPORT NO. AD-A092129

2.

3. Recipient's Accession No.

4. Title and Subtitle
Pilot Automated Influence Diagram Decision Aid

5. Report Date
Dec[illegible]78

6.

7. Author(s)
Robert J. Korsan James E. Matheson

8. Performing Organization Rept. No.
7078

9. Performing Organization Name and Address
SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.
(C) ANR-N00014-78-C0100
(G)

12. Sponsoring Organization Name and Address
Advanced Research Projects Agency
Cybernetics Technology Division
1400 Wilson Blvd.
Arlington, Virginia 22209

13. Type of Report & Period Covered
Technical

14.

15. Supplementary Notes

Final rept.

16. Abstract (Limit: 200 words)

1. Introduction

Influence diagrams were originally conceived as a way of visually representing dependencies among random variables. It was recognized that they provide an effective means of communicating probabilistic information in complex, uncertain situations. They were soon generalized to include decision as well as chance variables. An earlier SRI report, <u>Development of Automated Aids for Decision Analysis</u> (Miller, et al., May, 1976), describing influence diagrams and showed that they are a more fundemental way of capturing uncertain decision situations than the commonly used desicion tree methods. This recognition held the promise that they might provide a basis for more efficient modeling and solution of decision problems.

The report reviews the fundamentals of influence diagrams (adapted from Miller et al.) It illustrates with an example of toxic chemical testing how an analyst might use an automated influence diagram system. A first such system was implemented on the IBM 5110 mini-computer. This work allowed the development of computational algorithms in the APL language. Small decision problems are within the capabilities of this system. This development establishes the feasibility of implementing a comprehensive influence diagram system on a larger machine. Such a system could greatly assist decision makers and decision analysts. The final sections of this report explore guestions which arose out of our work.

17. Document Analysis a. Descriptors

b. Identifiers/Open-Ended Terms
Influence Diagrams, decision analysis, decision aids, decision making

c. COSATI Field/Group

18. Availability Statement:
Release Unlimited

19. Security Class (This Report)
Unclassified

20. Security Class (This Page)
Unclassified

21. No. of Pages
1

22. Price

(See ANSI–Z39.18) See Instructions on Reverse OPTIONAL FORM 272 (4–77) (Formerly NTIS–35) Department of Commerce

390664

DO NOT PRINT THESE INSTRUCTIONS AS A PAGE IN A REPORT

INSTRUCTIONS

Optional Form 272, Report Documentation Page is based on Guidelines for Format and Production of Scientific and Technical Reports, ANSI Z39.18–1974 available from American National Standards Institute, 1430 Broadway, New York, New York 10018. Each separately bound report—for example, each volume in a multivolume set—shall have its unique Report Documentation Page.

1. Report Number. Each individually bound report shall carry a unique alphanumeric designation assigned by the performing organization or provided by the sponsoring organization in accordance with American National Standard ANSI Z39.23–1974, Technical Report Number (STRN). For registration of report code, contact NTIS Report Number Clearinghouse, Springfield, VA 22161. Use uppercase letters, Arabic numerals, slashes, and hyphens only, as in the following examples: FASEB/NS–75/87 and FAA/RD–75/09.
2. Leave blank.
3. Recipient's Accession Number. Reserved for use by each report recipient.
4. Title and Subtitle. Title should indicate clearly and briefly the subject coverage of the report, subordinate subtitle to the main title. When a report is prepared in more than one volume, repeat the primary title, add volume number and include subtitle for the specific volume.
5. Report Date. Each report shall carry a date indicating at least month and year. Indicate the basis on which it was selected (e.g., date of issue, date of approval, date of preparation, date published).
6. Sponsoring Agency Code. Leave blank.
7. Author(s). Give name(s) in conventional order (e.g., *John R. Doe, or J. Robert Doe*). List author's affiliation if it differs from the performing organization.
8. Performing Organization Report Number. Insert if performing organization wishes to assign this number.
9. Performing Organization Name and Mailing Address. Give name, street, city, state, and ZIP code. List no more than two levels of an organizational hierarchy. Display the name of the organization exactly as it should appear in Government indexes such as Government Reports Announcements & Index (GRA & I).
10. Project/Task/Work Unit Number. Use the project, task and work unit numbers under which the report was prepared.
11. Contract/Grant Number. Insert contract or grant number under which report was prepared.
12. Sponsoring Agency Name and Mailing Address. Include ZIP code. Cite main sponsors.
13. Type of Report and Period Covered. State interim, final, etc., and, if applicable, inclusive dates.
14. Performing Organization Code. Leave blank.
15. Supplementary Notes. Enter information not included elsewhere but useful, such as: Prepared in cooperation with . . . Translation of . . . *Presented at conference* of . . . To be published in . . . When a report is revised, include a statement whether the new report supersedes or supplements the older report.
16. Abstract. Include a brief (200 words or less) factual summary of the most significant information contained in the report. If the report contains a significant bibliography or literature survey, mention it here.
17. Document Analysis. (a). Descriptors. Select from the Thesaurus of Engineering and Scientific Terms the proper authorized terms that identify the major concept of the research and are sufficiently specific and precise to be used as index entries for cataloging.

    (b). Identifiers and Open-Ended Terms. Use identifiers for project names, code names, equipment designators, etc. Use open-ended terms written in descriptor form for those subjects for which no descriptor exists.

    (c). COSATI Field/Group. Field and Group assignments are to be taken from the *1964 COSATI Subject Category List*. Since the majority of documents are multidisciplinary in nature, the primary Field/Group assignment(s) will be the specific discipline, area of human endeavor, or type of physical object. The application(s) will be cross-referenced with secondary Field/Group assignments that will follow the primary posting(s).
18. Distribution Statement. Denote public releasability, for example "Release unlimited", or limitation for reasons other than security. Cite any availability to the public, with address, order number and price, if known.

19. & 20. Security Classification. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED).

21. Number of pages. Insert the total number of pages, including introductory pages, but excluding distribution list, if any.
22. Price. Enter price in paper copy (PC) and/or microfiche (MF) if known.

CONTENTS

ILLUSTRATIONS

1. SUMMARY

Influence diagrams were originally conceived as a way of visually representing dependencies among random variables. It was recognized that they provide an effective means of communicating probabilistic information in complex, uncertain situations. They were soon generalized to include decision as well as chance variables. An earlier SRI report[1] described influence diagrams and showed that they are a more fundamental way of capturing uncertain decision situations than the commonly used decision tree methods. This recognition held the promise that they might provide a basis for more efficient modeling and solution of decision problems.

This report reviews the fundamentals of influence diagrams (adapted from Miller et al.[1]). It illustrates, with an example of toxic chemical testing, how an analyst might use an automated influence diagram system. The first such system was implemented on the IBM 5110 mini-computer. This work allowed the development of computational algorithms in the APL language. Small decision problems are within the capabilities of this system. This development establishes the feasibility of implementing a comprehensive influence diagram system on a larger machine. Such a system could greatly assist decisionmakers and decision analysts. The final sections of this report explore questions that arose out of our work.

---

[1] A.C. Miller, M.W. Merkhofer, R.A. Howard, J.E. Matheson, T.R. Rice, "Development of Automated Aids for Decision Analysis," SRI International, Menlo Park, CA (May 1976).

2. BACKGROUND

2.1 Influence Diagrams

An influence diagram is a way of describing the dependencies among state variables and decision variables. An influence diagram can be used to visualize the probabilistic dependencies in a decision analysis, to specify the states of information for which independencies can be assumed to exist, and to describe what state and decision information is available to the decisionmaker at each decision point.

Figure 2.1-1 shows how influence diagrams represent the dependencies among state variables and decision variables. A state variable is represented by a circle containing its name or number. An arrow pointing from chance variable A to chance variable B means that the outcome of A can influence the probabilities associated with B. An arrow pointing to a decision node from either another decision or state variable means that the decision is made with the knowledge of the outcome of the other decision or state variable. A connected set of squares and circles is called an influence diagram because it shows how state variables and decision variables are logically related.

The influence diagram in Figure 2.1-2(a) states that the probability distribution assigned to x <u>may</u> depend on the value of y, while the influence diagram in Figure 2.1-2(b) asserts that x and y are probabilistically independent for the state of information with which the diagram was drawn. Note that the diagram of Figure 2.1-2(a) really makes no assertion about the probabilistic relationship of x and y since, as we know, any joint probability $\{x,y|S\}$ can be represented in the form $\{x,y|S\} = \{x|y,S\}\ \{x|S\}$. However, since $\{x,y|S\} = \{y|x,S\}\ \{x|S\}$, the influence diagram of Figure 2.1-2(a) can be redrawn as shown in Figure 2.1-2(c); both are completely general representations requiring no independence assertions. While the direction of the arrow does not limit the form of the joint distribution for this simple example, it is used in more complicated problems to specify the states of information upon which independence assertions are made.

Similarly, with the three variables x,y,z, there are six possible influence diagrams of complete generality, one corresponding to each of the possible expansions. They are shown in Figure 2.1-3. While all of these representations are logically equivalent, they again differ in their suitability for probability assessment purposes. In large decision problems, the influence diagrams can display the needed assessments in a very useful way.

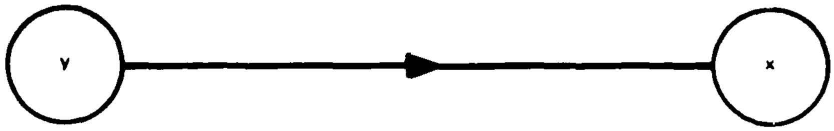
THE PROBABILITES ASSOCIATED WITH CHANCE VARIABLE B DEPEND ON THE OUTCOME OF CHANCE VARIABLE A

THE PROBABILITY OF CHANCE VARIABLE D DEPENDS ON DECISION C

THE DECISION MAKER KNOWS THE OUTCOME OF CHANCE VARIABLE E WHEN DECISION F IS MADE

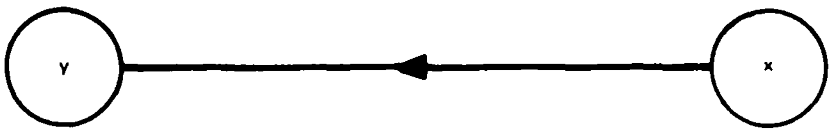
THE DECISION MAKER KNOWS DECISION G WHEN DECISION H IS MADE

FIGURE 2.1-1 DEFINITIONS USED IN INFLUENCE DIAGRAMS



(a) A SIMPLE INFLUENCE DIAGRAM

(b) AN EVEN SIMPLER INFLUENCE DIAGRAM



(c) AN ALTERNATE REPRESENTATION

FIGURE 2.1-2 TWO NODE INFLUENCE DIAGRAMS

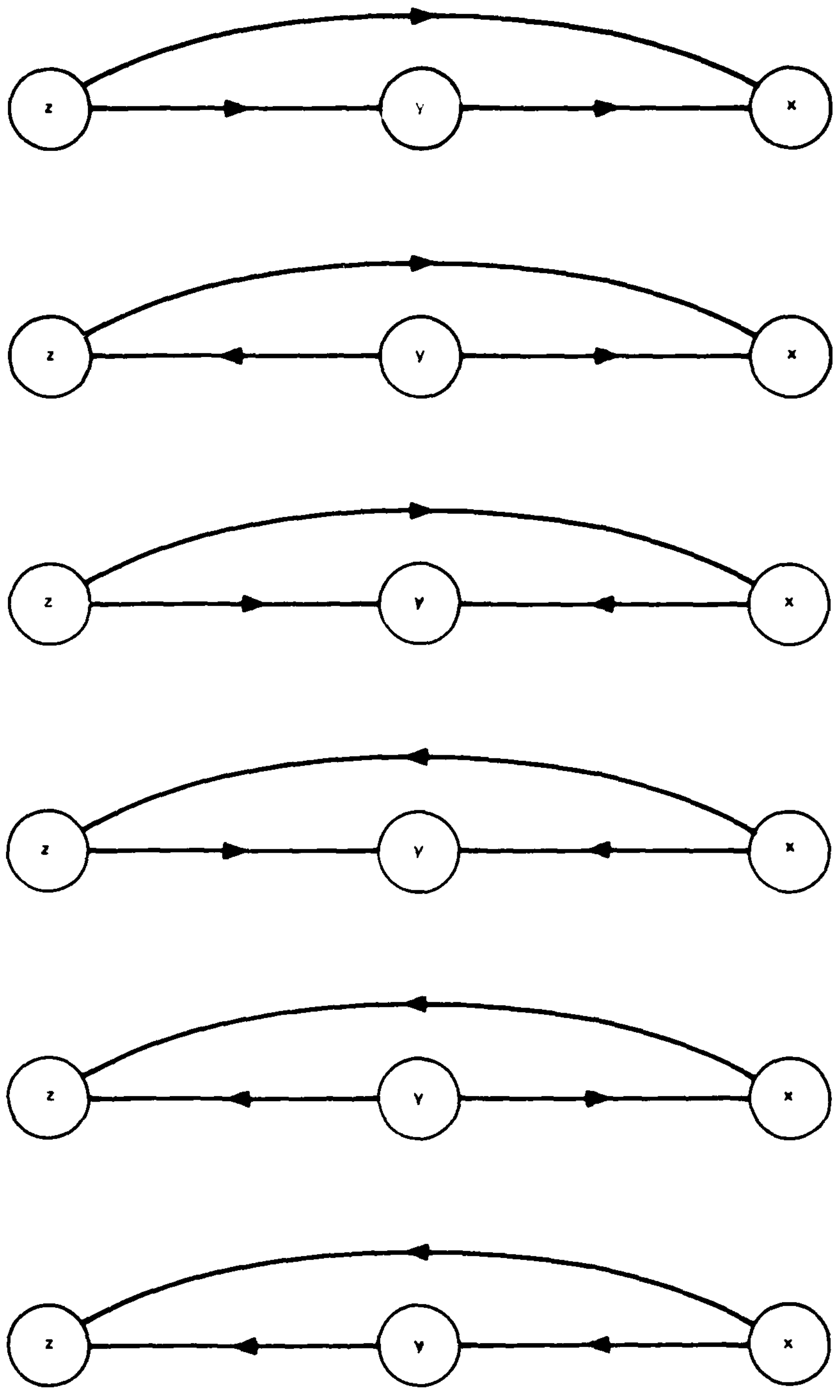


FIGURE 2.1-3 ALTERNATE INFLUENCE DIAGRAMS FOR x, y, z S

2.2 Formal Definition of an Influence Diagram

An influence diagram is a directed graph having no loops. It contains two types of nodes:

- Decision nodes represented by boxes (□)
- Chance nodes represented by circles (○)

Arrows between node pairs indicate influences of two types:

- Informational influences are represented by arrows leading into a decision node. These show exactly which variables will be known by the decisionmaker at the time that the decision is made.

- Conditioning influences are represented by arrows leading into a chance node. These show the variables on which the probability assignment to the chance node variable will be conditioned.

The informational influences on a decision node represent a basic cause-effect ordering, whereas the conditional influences into a chance node represent, as we have seen, a somewhat arbitrary order of probabilistic conditioning, which may not correspond to any cause-effect notion and may be changed by application of the laws of probability (e.g. Bayes' Rule).

Figure 2.2-1 is an example of an influence diagram. Chance node variables a, b, c, e, f, g, h, i, j, k, l, m, and o all indicate chance variables whose probabilities must be assigned, given their respective conditioning influences. Decision node variables d and n represent decision variables that must be set as a function of their respective informational influences. For example, the probability assignment to variable i is conditioned upon variables f, g, and i, and only these variables. In inferential notation, this assignment is {i|f, g, l, E}, where E represents a special S, the initial state of information upon which the construction of the entire diagram is based. As another example, the decision variable d is set with knowledge of variables a and c, and only these variables. Thus, d is a function of a and c.

One of the most important, but most subtle, aspects of an influence diagram is the set of possible additional influences that are not shown on the diagram. An influence diagram asserts that these missing influences do not exist.

In order to illustrate this characteristic of influence diagrams more clearly, we must make a few more definitions.

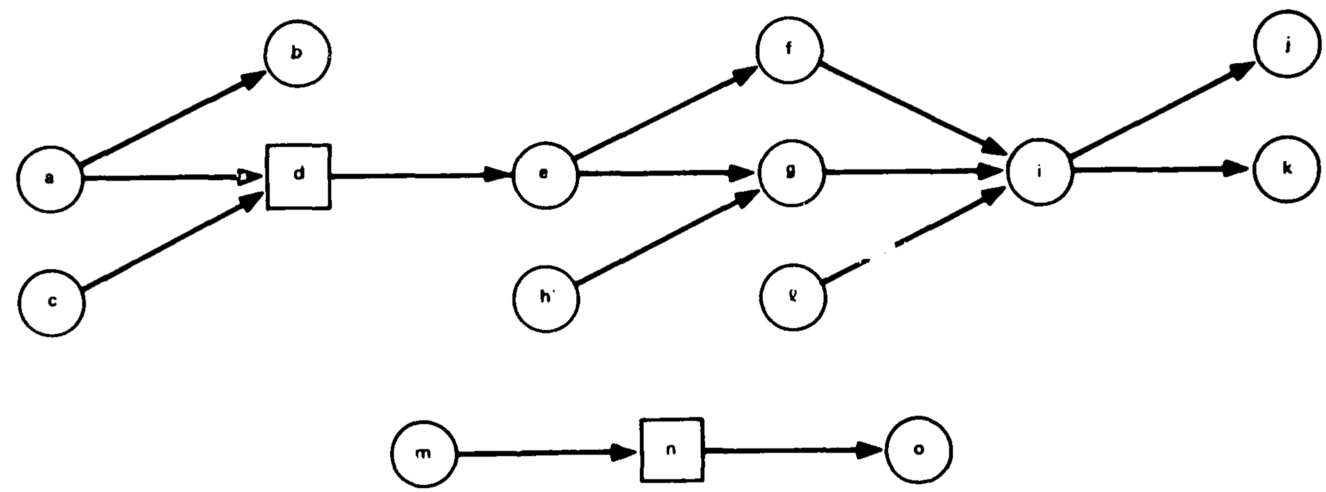


FIGURE 2.2-1 AN INFLUENCE DIAGRAM WITH DECISION NODES

- A path from one node to another node is a set of influence arrows, connected head to tail, that forms a directed line from one node to another.

With respect to any given node we make the following definitions:

- The predecessor set of a node is the set of all nodes having a path leading to the given node.

- The direct predecessor set of a node is the set of nodes having an influence arrow leading directly to the given node.

- The indirect predecessor set of a node is the set formed by removing from its predecessor set all elements of its direct predecessor set.

- The successor set of a node is the set of all nodes having a path leading from the given node.

- The direct successor set of a node is the set of nodes having an influence arrow leading directly from the given node.

- The indirect successor set of a node is the set formed by removing from its successor set all elements of its direct successors.

We refer to members of these sets as predecessors, direct predecessors, indirect predessors, successors, direct successors, and indirect successors. Figure 2.2-2 shows the composition of each of these sets in relation to node g.

2.3 Relationship of Influence Diagrams to Decision Trees

Some influence diagrams do not have corresponding decision trees. As in a decision tree, all probability assignments in an influence diagram -- including the assignment limitations represented by the structure -- must be based on a base state of information, E. Unlike a decision tree, the nodes in an influence diagram do not have to be totally ordered, nor do they have to depend directly on all predecessors. The freedom from total ordering allows convenient probabilistic assessment and computation. The freedom from dependence on all predecessors allows decisions to be based on informational event sets that are incompatible with a "single decisionmaker" point of view. If a single decisionmaker is assumed not to forget information, then the direct predecessor set of one decision must be a subset of the direct predecessor set of any subsequent decision. In the influence diagram of

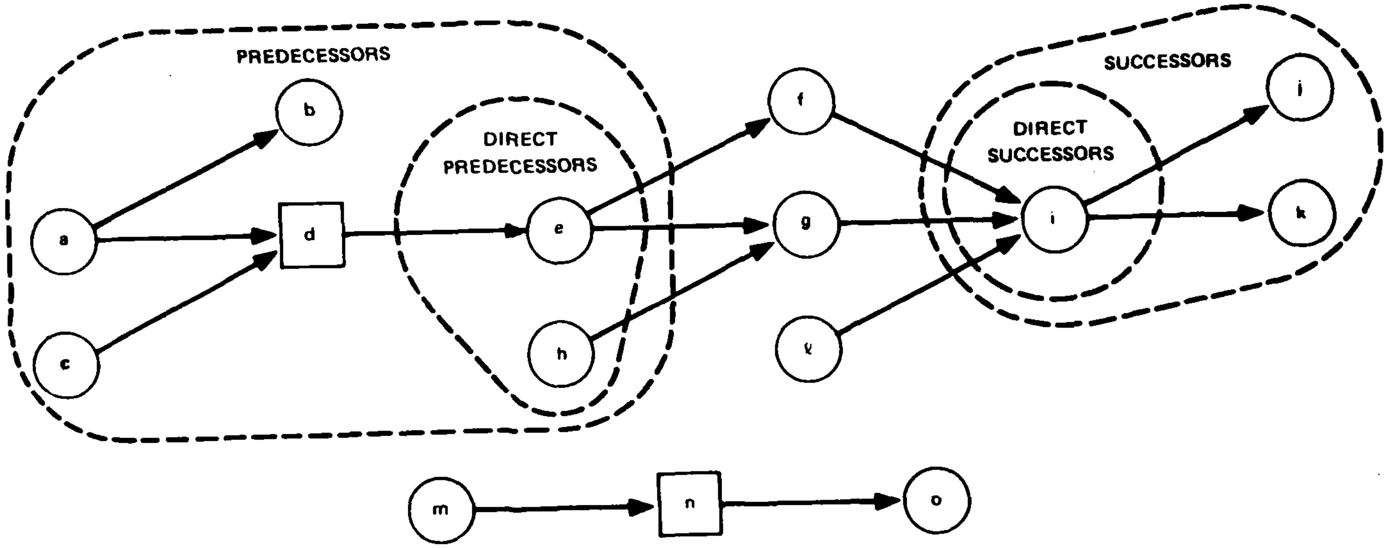


FIGURE 2.2-2 SOME SETS DEFINED BY THE NODE g

Figure 2.2-2, decisions d and n have mutually exclusive direct predecessor sets, (a,c) and (m). This situation could not be represented with a decision tree.

If the informational arrows shown as dashed lines in Figure 2.3-1 are added to Figure 2.2-2, then the influence diagram can be represented by a decision tree. Many different valid decision trees can be constructed from this new influence diagram. The only conditions are that they must (1) preserve the ordering of the influence diagram and (2) not allow a chance node to be a predecessor of a decision node for which it is not a direct predecessor. For example, the chance node m must not appear ahead of decision node d in a decision tree, because this would imply that the decision rule for d could depend on m, which is not the case.

The situation becomes more complex when we add a node such as p in Figure 2.3-2. If we were to construct a decision tree beginning with chance node p, it would imply that the decision rules at nodes d and n could depend on p, which is not the actual case. Node p represents a variable that is used in the probability assignment model but is not observable by the decisionmaker at the time he makes his decisions. In this situatiion, we would normally use the laws of probability (e.g. Bayes' Rule) to eliminate the conditioning of c on p. This process would lead to a new influence diagram reflecting a change in the sequence of conditioning. This would result in the inclusion of additional influences.

In Figure 2.3-3, the dashed arrow represents an influence that has been "turned around" by Bayes' Rule. The resulting diagram can be developed into a decision tree without further processing of probabilities. Also note that the change in the influence diagram required only information already specified by the original influence diagram (Figure 2.3-2) and its associated numerical probability assignments. Thus, it can be carried out by a routine procedure.

The foregoing considerations motivate two new definitions:

- A decision network is an influence diagram:

  (1) that implies a total ordering among decision nodes,

  (2) where each decision node and its direct predecessors directly influence all successor decision nodes.

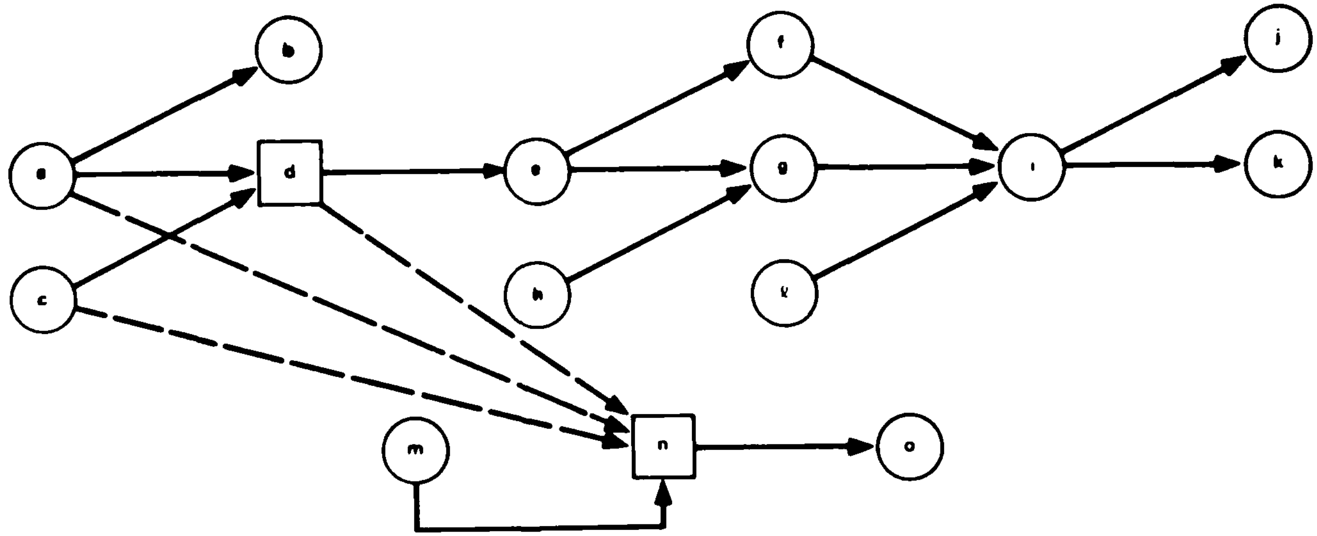


FIGURE 2.3-1 AN INFLUENCE DIAGRAM REPRESENTABLE BY A DECISION TREE

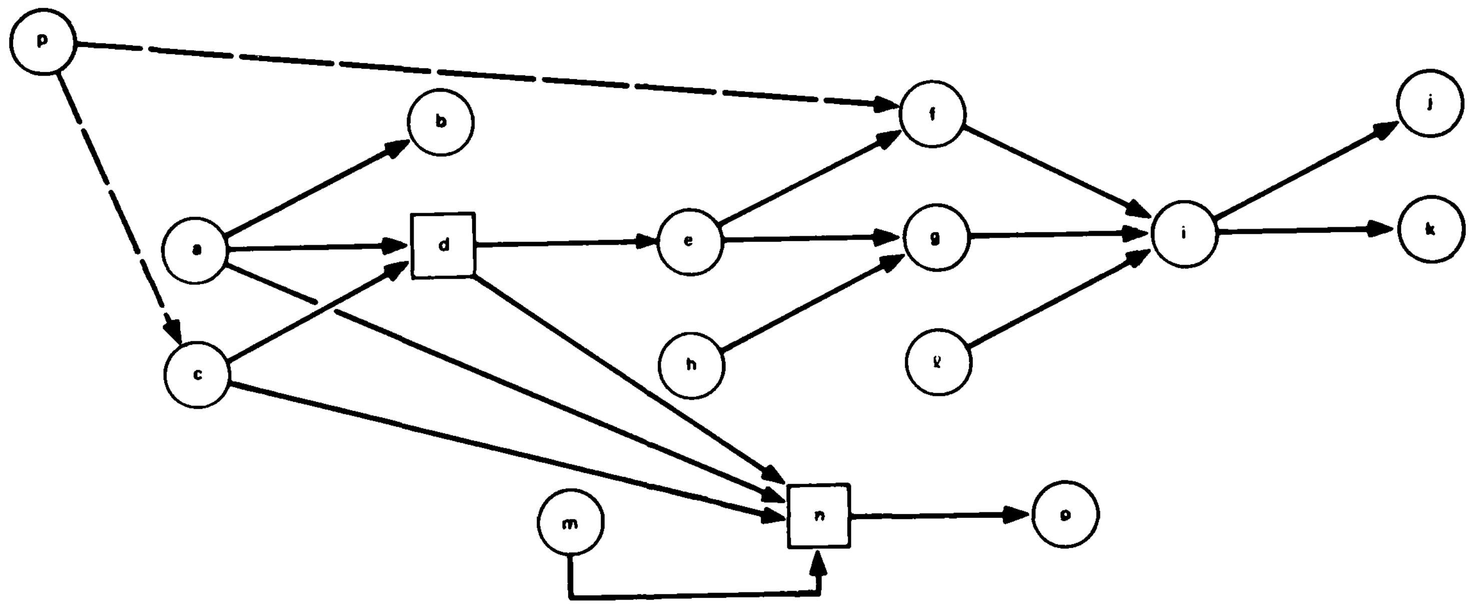


FIGURE 2.3-2 INFLUENCE DIAGRAM REQUIRING PROBABILISTIC MANIPULATION BEFORE DECISION TREE CONSTRUCTION

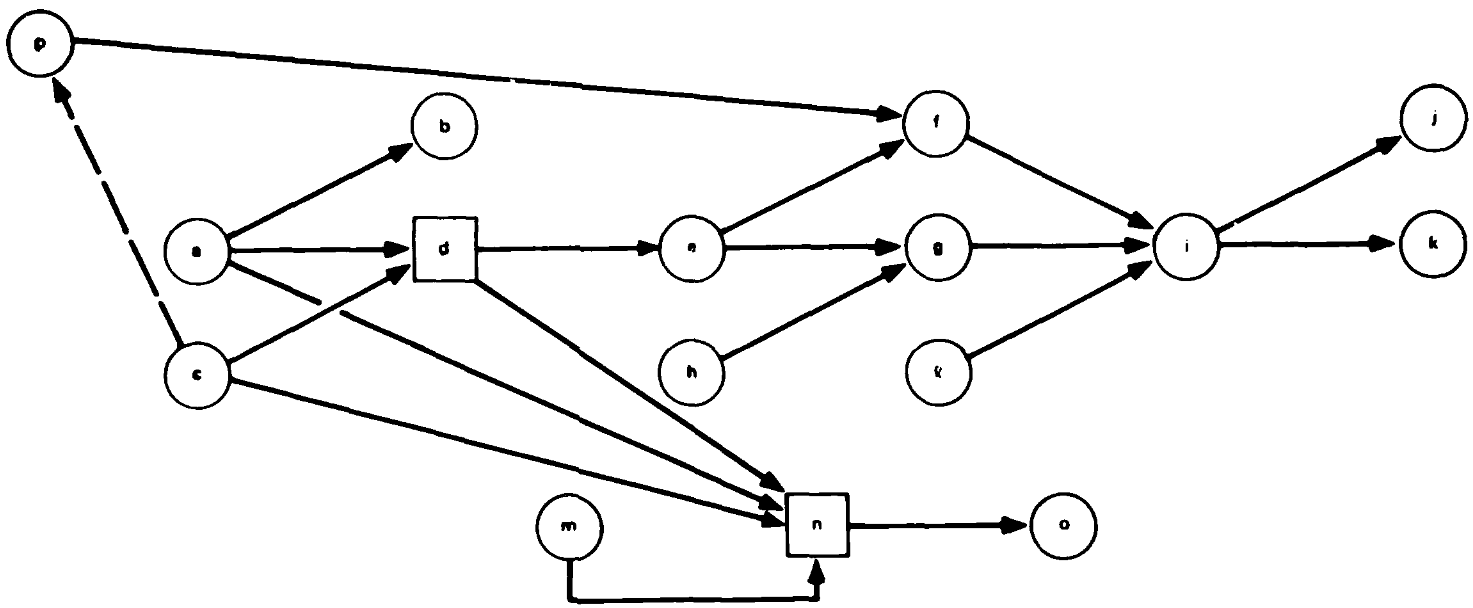


FIGURE 2.3-3 INFLUENCE DIAGRAM READY FOR DEVELOPMENT INTO A DECISION TREE

- A <u>decision tree network</u> is a decision network:
  (3) where all predecessors of each decision node are direct predecessors.

Requirement (1) is the "single decisionmaker" condition and requirement (2) is the "no forgetting" condition. These two conditions guarantee that a decision tree can be constructed, possibly after some probabilistic processing. Requirement (3) assures that no probabilistic processing is needed, so that a decision tree can be constructed in direct correspondence with the influence diagram.

As an example, consider the standard inferential decision problem represented by the decision network of Figure 2.3-4(a). As we discussed earlier, this influence diagram cannot be used to generate a decision tree directly, because the decision node c has a non-direct predecessor that represents an unobservable chance variable. To convert this decision network to a suitable decision tree network, we simply reverse the arrow from a to b, which is permissible because they have only common predecessors, namely none.* We thus achieve the decision tree network of Figure 2.3-4(b), and with redrawing, we arrive at Figure 2.3-4(c).

Specifying the limitations on possible conditioning by drawing the influence diagram may be the most significant step in probability assignment. The remaining task is to specify the numerical probability of each chance node variable conditioned on its direct predecessor variable.

---

*See graphical manipulation in Miller et al., "Development of Automated Aids for Decision Analysis," SRI International, Menlo Park, CA, p. 126 (May 1976).

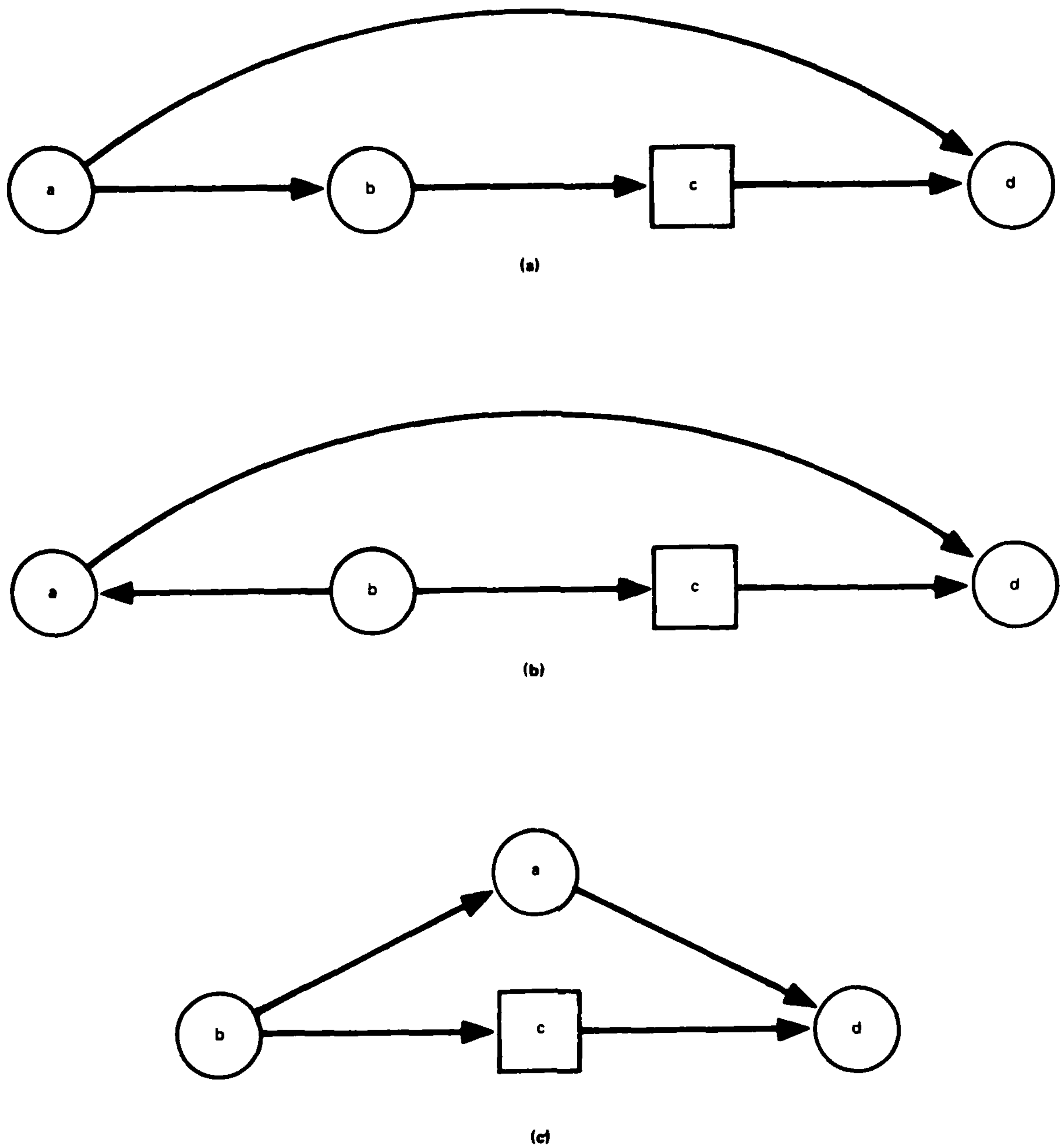


FIGURE 2.3-4 THE PROCESS OF CONVERTING A DECISION NETWORK TO A DECISION TREE NETWORK

3. THE INFLUENCE DIAGRAM AID

An automated decision aid based on the concept of influence diagrams has been implemented on the IBM 5110. This aid consists of programs to assist 4 major phases of the analysis of a decision problem. These phases are (1) influence diagram construction and problem definition, (2) transformation of the influence diagram to a decision tree network, (3) probability and value elicitation, and, finally, (4) tree generation and policy evaluation. The following sections describe the purpose, major capabilities, and limitations of this experimental aid. Section 4 gives a detailed description of the use of this aid.

3.1 Purpose of the Influence Diagram Aid

The influence diagram aid was developed to assist the analysis of decision problems. The fundamental structure used to coordinate all phases of the analysis is the influence diagram. The aid can be used to streamline all phases of analysis.

Initially, the aid allows the user to interactively develop the graphical representation of the decision problem structure on the screen of the IBM 5110. This phase includes questions designed to increase problem complexity in levels. All important areas of the decision problem are developed uniformly.

The aid will take the problem structure that is implicit in the influence diagram and interactively construct a decision tree network. The aid attempts to streamline the elicitation process by coordinating the original influence diagram and the decision tree network constructed from it. When independence has been implied, the program will check that this was indeed intended during elicitation. The purpose of this check is to minimize the information that needs to be supplied to analyze the decision problem. This allows the user to catch any structural errors contained in the original influence diagram. Once the influence diagram has been elicited and the probability distributions assessed, the aid constructs and analyzes the decision tree associated with this problem. This finally results in an expected value, a lottery, and an optimal state contingent policy for the decision problem.

3.2 Major Capabilities

The major capabilities of the aid are:

- Guided assistance in analysis of a decision problem.
- Interactive elicitation and modification of an influence diagram.
- Analysis of an influence diagram and guidance on the structure needed to create a decision tree network.
- Elicitation of decision alternatives and probability distributions necessary to solve the decision problem.
- Construction, rollback, and policy evaluation based on the information collected.
- Value of information computations.
- Library facilities that allow the user to save the problem (or versions of it) at any phase of the analysis.

3.3 Limitations

The limitations of the decision aid, as currently implemented are:

- Linear additive value functions.
- Utility functions unavailable.
- Access to APL models of outcomes not available.
- Normal approximation to continuous probability distributions.
- A total of approximately 7-10 state variables, decisions, and outcomes in any particular problem.

Some of these limitations could be easily overcome in a future design. Others are dependent upon the physical capabilities of the IBM 5110 and could be remedied by implementation of a different system.

4. USING THE INFLUENCE DIAGRAM PROGRAMS

The influence diagram programs are a set of four APL workspaces (WS). These are stored on a diskette that has been specially prepared with space for five problems. All of the workspaces are designed around using this file format. There is a common directory that allows access to all problems. The major phases of problem analysis are performed by:

1. *WS DIAGRAM* - interactive elicitation of influence diagram.

2. *WS XFORM* - analyze the influence diagram and detail the steps necessary to create a decision tree network.

3. *WS PROBS* - based on the influence diagram, elicit the joint probability distribution over the problem variables. Also, determine the value trade-offs among the outcome variables.

4. *WS SOLVE* - determine the expected value, value lottery, and state contingent policy for this problem.

The reader is referred to the IBM 5110 System Library for details on machine installation, maintenance, serial I/O interface operation, etc. The current aid is designed to be used on a 5110 with at least one disk drive.

4.1 Preparing a Diskette

The influence diagram aid is a portable system. Copies of the aid can be made for distribution to others. The steps involved are simple:

1. Initialize the diskette -- see the IBM 5110 customer support functions reference manual for a description of the diskette initialization function.

2. Place the MASTER tape cartridge in the tape reader.

3. Execute the APL system command

)PROC 1001 *PREPARE*

4. Execute the APL system command

)LOAD 11001 *DIAGRAM*

Step 3 performs the diskette preparation. Step 4 places you at the beginning of the analysis and solution process.

4.2 The *WS DIAGRAM*

The purpose of the *WS DIAGRAM* is to interactively build an influence diagram. The user is questioned about the structure of the problem. This portion of the aid is designed to have the user think back from values to current decisions. The information provided is used to guide the user into thinking about the problem in uniform layers of complexity.

4.2.1 Conventions

The screen of the 5110 is a restrictive graphical medium. In order to maximize the use of the screen, we have introduced some conventions. All problem variables are represented by a single symbol. We make a distinction between state variables (chance variables that do not enter directly into the value function) and outcome variables (chance variables that do directly enter into the value function). The symbols available are:

Outcome variables - the digits 0 through 9

State variables - the alphabet *A* through *Z*

Decision variables - the underlined alphabet *A̲* through *Z̲*

The names associated with the symbol are limited to 18 characters. Variables are connected by directed arrows (→↔ etc.). When arrows cross but do not connect, the crossing point is denoted by a "jot" (∘). If arrows cross and connect, the junction is denoted by a "plus" (+). See Figure 4.2.1-1 (a) and (b) for examples of these conventions. When describing influences, variable symbols are separated by either → or ←, which denotes the direction of the influence. Hence, *A*→*B* is read as "*A* influences *B*."

```
        B
        ↓
A → → → o → → → B
        ↓       ↑
        ↓       ↑
        ↓ → → → C
```

(a) Arrows crossing but not connecting.

```
        B
        ↓
        ↓
A → → → + → → → B
        ↓       ↑
        ↓       ↑
        ↓ → → → C
```

(b) Arrows crossing and connecting.

Figure 4.2.1-1(a). Graphic Conventions for Connecting Variables

4.2.2 Retrieving Old Problems/Starting New Ones

The aid, upon loading any of the four workspaces, will automatically start execution. The user is presented with the following:

OPTIONS
1 = *START A NEW PROBLEM FROM SCRATCH*
2 = *RETRIEVE AN OLD PROBLEM FROM DISK AND CONTINUE*
PLEASE ENTER THE NUMBER OF THE DESIRED OPTION AND PRESS EXECUTE

Entering a "1" starts a fresh problem, which may be saved at the end of the session. Entering a "2" will cause the problem directory to be shown and a problem that is partially solved may be retrieved, for example --

THE PROBLEMS CURRENTLY STORED ARE:
1. *ARPA*
2. *TEST*
3. *ONR*
4. *SHALE*
5. *UNUSED*

PLEASE ENTER THE NUMBER OF THE PROBLEM DESIRED.

Entering a "1" will retrieve the problem that previously has been called ARPA. This problem saving/retrieval is common to the four workspaces that make up the aid.

4.2.3 Available Options

The following set of options is presented upon entry to the *VS DIAGRAM*.

OPTIONS

1 = *CONTINUE STRUCTURE ELICITATION*
2 = *MAKE CORRECTIONS TO EXISTING DIAGRAM*
3 = *DISPLAY VARIABLE DICTIONARY*
4 = *DISPLAY DIAGRAM*
5 = *DISPLAY DISK DIRECTORY*
6 = *EXIT AND STORE PROBLEM*
7 = *EXIT*
PLEASE ENTER THE NUMBFR OF THE DESIRED OPTION AND PRESS EXFCUTE

This is presented after starting/retrieving a problem and at the completion of any option. Option 1 is a process whereby the user is questioned about the next variables to be entered into the problem. The user places the symbol on the screen, names the symbol, and describes the influences. The user is first asked about outcome variables (variables that directly enter into the value function). This is followed by the variables that influence these outcomes, etc. During this process, any of the following errors may be produced:

ERROR: INVALID INFLUENCE SYMBOL. EDIT LINE AND PRESS EXECUTE:
ERROR: UNDEFINED VARIABLE. EDIT LINE AND PRESS EXECUTE:
ERROR: INVALID DELIMITER. EDIT LINE AND PRESS EXECUTE:
ERROR: UNDEFINED VARIABLE. EDIT LINE AND PRESS EXECUTE:
ERROR: INVALID INFLUENCE. EDIT LINE AND PRESS EXECUTE:

These are straightforward, except for invalid influence, which may occur in response to two situations. First, a loop may be created, in which case you are notified and forced to repeat the process. Second, given the limitations of the characters used for drawing influences, the physical layout of the 5110 screen may not allow the influences to be drawn. In either case, you must edit the description and/or the screen and re-specify the influences.

The description of influences is done symbolically. The '→' is used to designate influence. Hence, in this language, $A \rightarrow B$ is read as "A influences B". Multiple statements may be placed on a single line, separating them by semi-colons, eg. $A \rightarrow B$; $A \rightarrow C$; $B \rightarrow C$.

Option 2 allows you to interact with the screen of the 5110 and modify the diagram that has been drawn. The correction options are:

1 = *REMOVE A PROBLEM VARIABLE (AND ITS INFLUENCES)*
2 = *REMOVE A SPECIFIC INFLUENCE*
3 = *MOVE THE SCREEN POSITION OF A VARIABLE*
4 = *ADD NEW INFLUENCES*
5 = *ADD PREVIOUSLY FORGOTTEN VARIABLES*

Option 3 displays a table that has three columns, which lists the symbols and names used as state variables, decision variables, and outcome variables. A sample of this output is given by:

STATE VARIABLES	*DECISION VARIABLES*	*OUTCOMES*
R = RED RESPONSE	*A = TASK FORCE ACT*	1 = *WAR RISK*
		2 = *MATERIALS*
		3 = *NEUTRALIZATION*

Option 4 displays the influence diagram, which might look like:

```
→ → → → → → → + → → → → → → → 1
↑             ↑
↑ → → → → → → + → → → → → → → 2
↑             ↑
A → → → → → → R
↓             ↓
↓             ↓
↓ → → → → → → + → → → → → → → 3
```

Options 5, 6, and 7 are used to control the problems that are added to the set already saved on the current diskette.

4.3 The *WS XFORM*

The *WS XFORM* is used to analyze a previously developed influence diagram. This results in a list of conditions that are sufficient to convert the current diagram into a decision tree network. These conditions are presented so that the user may add them to the current diagram or explicitly reject the analysis of the problem as a decision tree.

If probabilities have been previously elicited, the functions will perform the necessary probabilistic manipulations wherever possible (e.g. Bayes' Rule when reversing the direction of an influence). Probabilities will be elicited once again, with consistency checking when no other action is available. The conventions used are the same as the previous *WS*'s.

4.3.1 Available Options

The following set of options are presented upon entry to the *WS XFORM*.

OPTIONS:
1 = *CONVERT DIAGRAM TO DECISION TREE NETWORK*
2 = *PRINT DIAGRAM ON SIO DEVICE*
3 = *DISPLAY VARIABLE DICTIONARY*
4 = *DISPLAY DIAGRAM*
5 = *DISPLAY DISK DIRECTORY*
6 = *EXIT AND STORE PROBLEM*
7 = *EXIT*
PLEASE ENTER THE NUMBER OF THE DESIRED OPTION AND PRESS EXECUTE

Option 1 displays the list of conditions that must be met in order to have the influence diagram become a decision tree network. The system allows the user to specify if all these changes are to be made. If not, an option is presented that allows the user to manipulate the diagram and show the changes needed.

Option 2 allows a hard copy output of the influence diagram when the IBM 5110 is configured without a printer (assuming the SIO option). The remaining options are the same as those of *WS DIAGRAM*, Section 4.2.3.

4.4 The *WS PROBS*

The *WS PROBS* is used to elicit the minimum number of probability elicitations based on the influence diagram. Discrete and continuous distributions are allowed. The "normal" approximation is used when obtaining continuous distributions. The conventions used are the same as previous *WS*'s.

4.4.1 Available Options

The following set of options are presented upon entry to the *WS PROBS*:

OPTIONS:
1 = *CONTINUE PROBABILITY ELICITATION*
2 = *EDIT PROBABILITIES*
3 = *DISPLAY VARIABLE DICTIONARY*
4 = *DISPLAY DIAGRAM*
5 = *DISPLAY DISK DIRECTORY*
6 = *EXIT AND STORE PROBLEM*
7 = *EXIT*
PLEASE ENTER THE NUMBER OF THE DESIRED OPTION AND PRESS EXECUTE

Option 1 uses the influence diagram to present a series of questions that elicit the distribution needed from the user. In the case of discrete distributions, the user must provide the levels and probabilities of each level, based on appropriate conditioning. In the case of a continuous distribution, the simple normal approximation is used. Probabilities of .25, .5 and .25 are associated with the 10, 50 and 90 fractiles. The first two moments of this distribution are used to specify the approximate normal distribution. An example of such an elicitation is:

```
PROBABILITY ELICITATION FOR VARIABLE 2 = MATERIALS
CONDITIONAL ON VARIABLES          VALUE

    A = TASK FORCE ACT             AIR
    R = RED RESPONSE              ATTACK

    PLEASE ENTER THE '2' VALUES CORRESPONDING TO THE
    10, 50, AND 90% POINTS FOR THE CUMULATIVE '2'
    DISTRIBUTION (OR TO GET THE MENU, ENTER: MENU).
        10% (OR MENU):  3750   50%:  12250  90%

        ARE THE ABOVE VALUES CORRECT?  (Y OR N):
```

Option 2 allows the user to change some of the values previously input. These changes are incorporated and the appropriate conditioned inputs queried. These data are used to update the joint distribution fundamental to the solution of the problem. The remaining options are the same as those of *WS DIAGRAM*, section 4.2.3.

4.5 The *WS SOLVE*

This *WS* takes the data gathered previously and elicits a value function. Currently, value functions are restricted to linear combinations of outcome variables. Once elicited, the value lottery is computed and printed along with a state dependent optimal policy and expected value.

5. TOXIC CHEMICAL TESTING EXAMPLE

To illustrate the power of influence diagrams to solve complex problems of decisionmaking and information acquisition, we shall apply this method to a problem of toxic chemical testing. Let us suppose that a chemical having some benefits also is suspected of possible carcinogenicity. We wish to determine whether to ban, restrict, or permit its use, and also whether to undertake any information gathering regarding cancer-producing activity of the chemical or its degree of exposure to humans.

The primary decision problem can be formulated by drawing the influence diagram of Figure 5-1. This figure tells us that the economic value of the product and the cancer cost attributed to it both depend on the decision regarding usage of the chemical. The (probability assignment on) economic value given the usage decision is independent of the human exposure, carcinogenic activity, and the cancer cost. However, the cancer cost is dependent upon the usage decision, as well as both the carcinogenic activity and human exposure levels of the chemical. The net value of the chemical, given the economic value and the cancer cost, is independent of the other variables. Also, human exposure and carcinogenic activity are independent.

These relationships are not necessarily obvious ones; they depend on knowledge of the problem at hand. For example, the economic value of a particular chemical might depend on its chemical activity, which in turn might be closely related to its carcinogenic activity. In such a case, an arrow might have to be added from carcinogenic activity to economic value.

The next step would be to obtain probability and value assessments corresponding to the influence diagram. For example, an automated influence diagram system might ask for a list of usage decision alternatives. In this case, they are BAN, RESTRICT, or PERMIT. Next, it might ask for the economic value given each of these alternatives. In this case, the permit alternative is considered to have a reference value of zero, the restrict alternative a substitute process cost of $1 million, and the ban alternative a substitute process cost of $5 million.

The next question might be to assess possible outcomes for human exposure and carcinogenic activity, along with their corresponding (unconditional) probabilities. The probability trees of Figure 5-2 illustrate these assignments. Next, we might be asked for the cancer

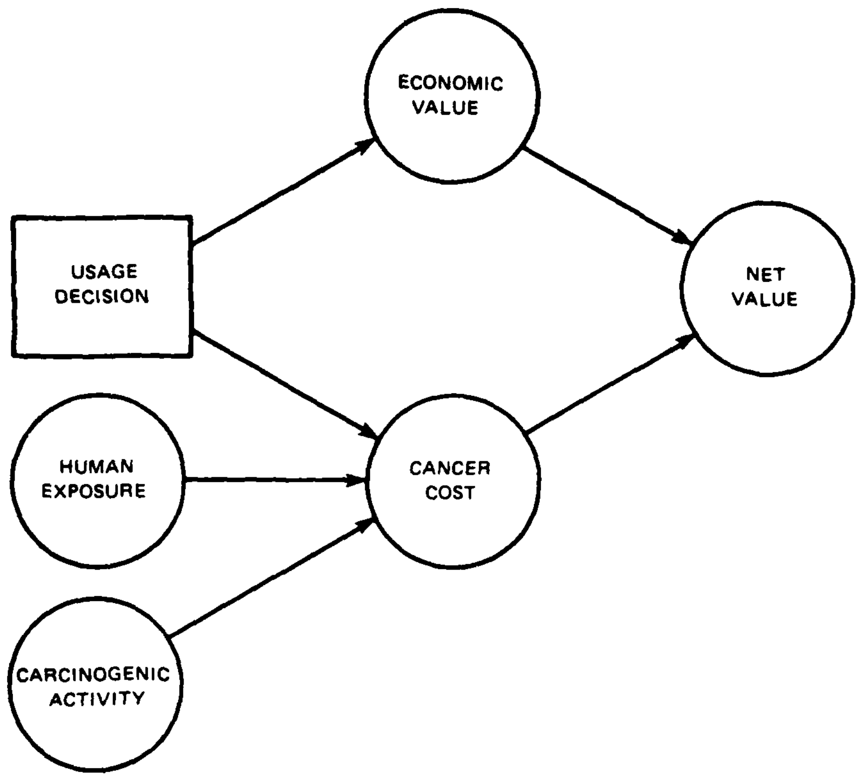


FIGURE 5-1 INFLUENCE DIAGRAM FOR PRIMARY DECISION

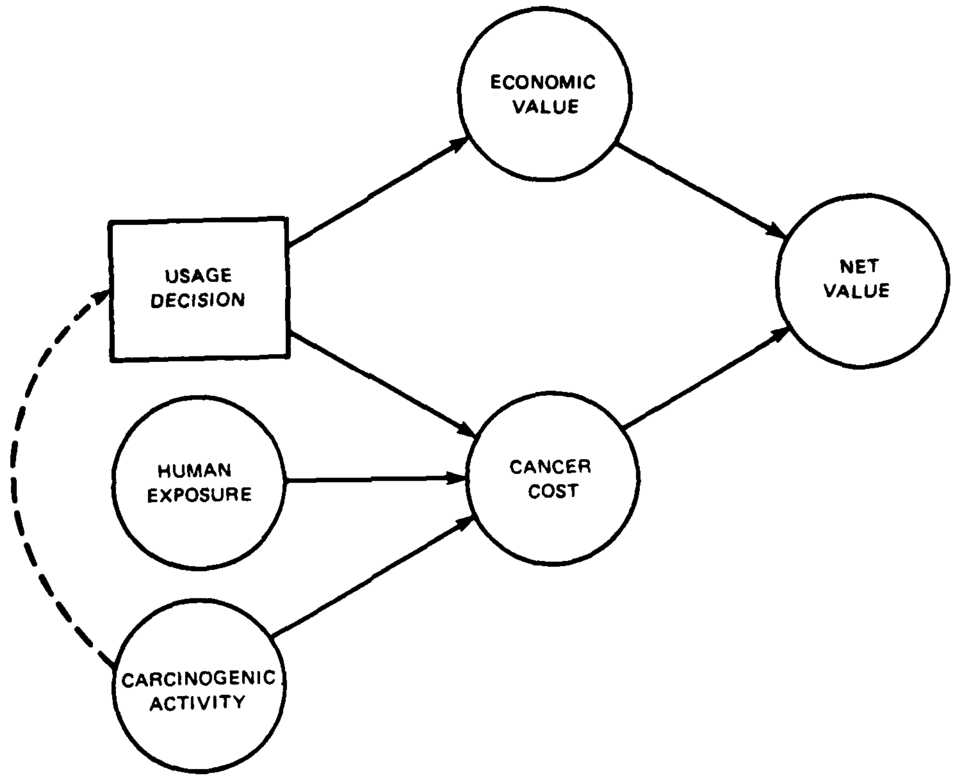


FIGURE 5.1-1 INFLUENCE DIAGRAM MODIFICATION TO DETERMINE THE VALUE WITH PERFECT INFORMATION ON CARCINOGENIC ACTIVITY

cost, given human exposure and carcinogenic activity levels, as well as the disposition decision. This cost is deterministic and is given in Table 5-1. Finally, the net value is stated to be simply the sum of the economic value and cancer cost.

All of this information would come out of detailed modeling and expert judgment regarding the decision situation. Once it has been captured with the influence diagram, analysis can proceed. In this case, an automated influence diagram procedure could generate the appropriate decision tree, display it if desired by the user, and determine that the best decision is to restrict usage. The expected value given this decision is a cost of $2.2 million. In this example, we will consider only the expected value or risk neutral case, although the case of risk aversion can be treated with little difficulty.

Table 5-1

CANCER INCIDENCE
(Valued at $100,000 each)

	Human Exposure								
	PERMIT Alternative			RESTRICT Alternative			BAN Alternative		
Carcinogenic Activity	Low	Med	High	Low	Med	High	Low	Med	High
Inactive	0	0	0	0	0	0	0	0	0
Moderate	5	50	500	0.5	5	50	0	0	0
Very active	100	1,000	10,000	10	100	1,000	0	0	0

5.1 The Value of Perfect Information

Before investigating actual information-gathering alternatives, the usual decision analysis practice is to determine the value of perfect information (clairvoyance) on the uncertain variables. The value of clairvoyance furnishes an upper limit on the value of real information gathering.

With an automatic influence diagram procedure, these calculations are trivial. For example, to calculate the value with perfect information on carcinogenic activity, we need only add the influence arrow indicated in dotted lines on Figure 5.1-1. This modification states that the decisionmaker knows the degree of carcinogenic activity when he makes the usage decision. The result is an expected cost of $1.1 million and a decision rule to permit if inactive, restrict if moderate, and ban if very active. This means that the expected value of perfect information is the original $2.2 million minus the $1.1 million expected cost, which is $1.1 million. Figure 5.1-2 shows a complete display of the decision tree for this case, which could be automatically generated upon request of the user.

5.2 The Value of Imperfect Information

In order to place a value on imperfect information, we must model the information source. To be useful, the informational report must depend probabilistically on one or more of the uncertain variables in the problem. In order to incorporate this dependence, we augment the influence diagram with a model of the information-gathering activity.

In the example at hand, it might be possible to carry out a laboratory test of the carcinogenic activity of the chemical. In this case, we begin by adding a chance node to represent the report from the activity test. In Figure 5.2-1 we have added an activity test node; we have drawn an arrow to it from the carcinogenic activity node (showing that the test result depends on the actual carcinogenic activity of the chemical), and we have drawn an arrow from the activity test to the usage decision (showing that the decisionmaker will know the test result when he makes the usage decision). We must also check the logic of each probabilistic statement represented in the diagram, because additional knowledge, in principle, could change the probabilistic dependence elsewhere in the diagram.

An automated system would now ask us to define the test results. We would reply that there are three test results, called "INACTIVE," "MODERATELY ACTIVE," and "VERY ACTIVE," corresponding to the possibilities for the actual activity. However, unlike perfect information, these test result indications may be wrong. The system would now ask us to supply the probabilities of these test results for each state of carcinogenic activity (the likelihood function). Figure 5.2-2 shows a possible display with the assigned probabilities.

All of the information needed to determine the value of the carcinogenic activity test has now been supplied. However, the influence diagram of Figure 5.2-1 is what we term a decision network, rather than a decision tree network, so it must be manipulated into decision tree network form before a decision tree can be generated and evaluated. The problem is that the carcinogenic activity node precedes the usage decision node, but activity is unknown to the decisionmaker when he makes the usage decision. A decision tree beginning with resolution of carcinogenic

FIGURE 5-2 INITIAL PROBABILITY ASSIGNMENTS

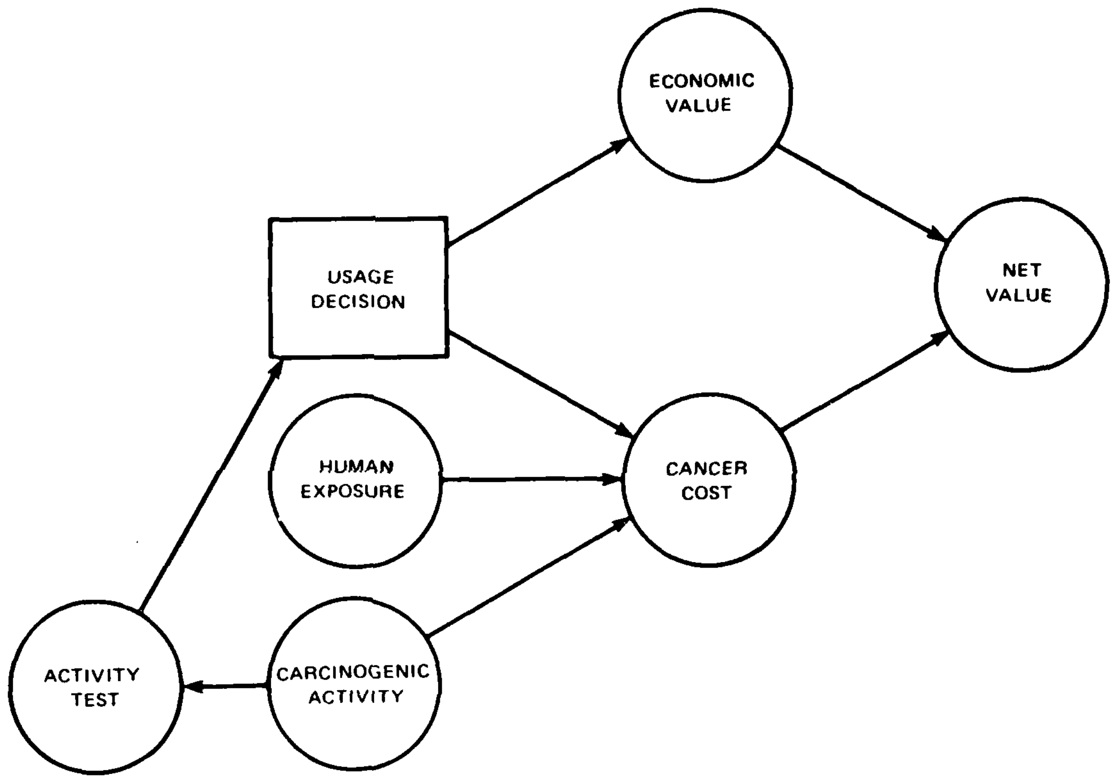


FIGURE 5.2-1 INFLUENCE DIAGRAM TO DETERMINE THE VALUE WITH IMPERFECT INFORMATION ON CARCINOGENIC ACTIVITY

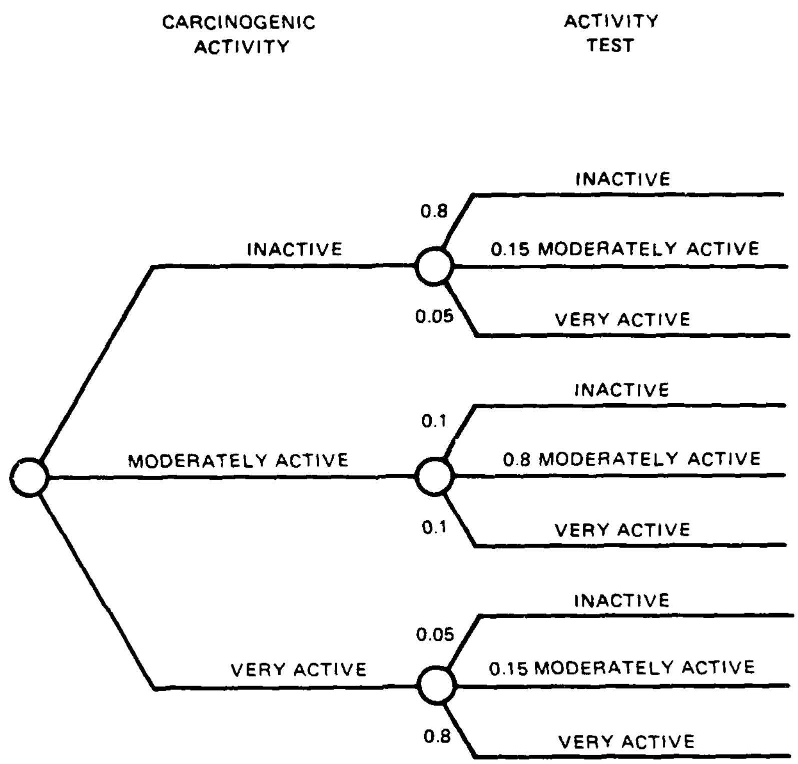


FIGURE 5.2-2 ACTIVITY TEST PROBABILITY ASSIGNMENTS

activity would incorrectly give this information to the decisionmaker. The problem is resolved by turning around the influence arrow between carcinogenic activity and the activity test. This manipulation requires the application of Bayes' rule to determine from the original probability assignments a new set conditional in the opposite order. Carrying out this manipulation would be straightforward for an automated system and will result in the desired decision tree network. In fact, a sophisticated system could determine that this manipulation was required and carry it out without being asked by the user.

Evaluation of this network yields an expected cost, given the activity test option, of $1.96 million. Subtracting this cost from the original cost of $2.20 million yields an expected value of $0.24 million from a free activity test. This is the upper limit on the price the decisionmaker should pay for the actual test.

A test of the degree of human exposure also could be treated by a similar modification of the influence diagram. Finally, the value of testing both carcinogenic activity and human exposure could be determined by making both modifications as illustrated in Figure 5.2-3.

We have shown in this example how influence diagrams can be used to model the primary decision problem, to determine the value of perfect information on the uncertain variables, and, finally, to determine the value of actual, but imperfect, information. The latter calculation usually requires the application of Bayes' law. Decision tree methods require the user to apply Bayes' law and supply the answers, or at least the formula, for the appropriate probabilities on the decision tree. Because the influence diagram captures the logic of the problem in a more fundamental way, the user need only supply the initial probabilities that represent his model of the information-gathering activity, and an automated system can carry out the rest of the analysis. This example shows how influence diagrams can greatly simplify the probabilistic modeling and decisionmaking process.

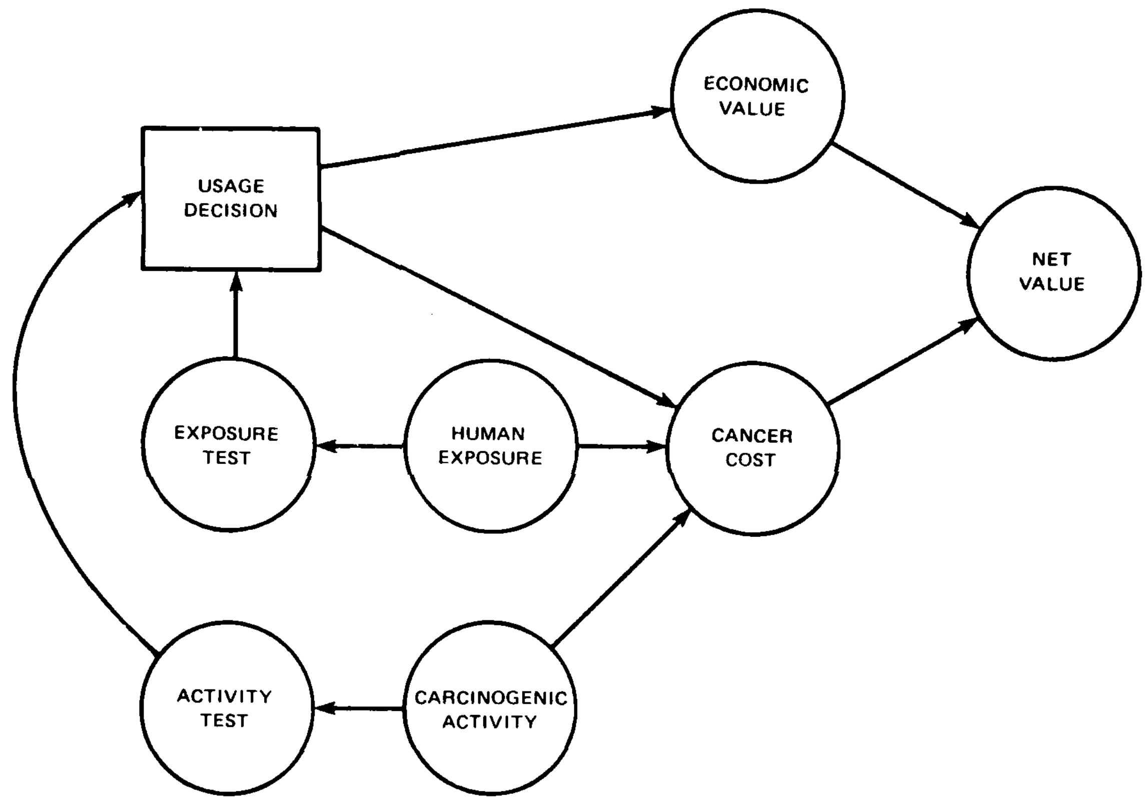


FIGURE 5.2-3 INFLUENCE DIAGRAM TO DETERMINE THE VALUE OF IMPERFECT INFORMATION ON BOTH CARCINOGENIC ACTIVITY AND HUMAN EXPOSURE

6. CONCLUSIONS

In this work, we have developed the first implementation of an automated, interactive process for constructing influence diagrams. We have also illustrated how an interactive session with an automated system might be used to analyze an example problem of a toxic chemical decision.

A logical next step is the thorough testing of this pilot system during actual analysis. The SRI Decision Analysis Group now routinely uses influence diagrams as a modeling tool. However, almost all analysis is still carried out using older automated procedures having greater computational capacity. The small size of the IBM-5110 computer places a severe limit on the complexity of problems that can be handled by the pilot system. Nevertheless, actual tests with small problems should provide a critique of both the modes of computer interaction and the algorithms used.

This pilot implementation has demonstrated the feasibility of automated influence diagrams as a basic new tool for decision analysis and model building. The high value of faster and more accurate problem-solving capability dictates that rapid progress to practical, full-scale implementation of automated systems should be made.

6.1 Directions for Further Development

A recent dissertation by Daniel Owen[1] explored the use of influence diagrams as a quantitative tool to provide guidance to model building. In brief, he developed procedures for determining the best place to expand or refine a decision model throughout the model construction process. Owen also developed a powerful matrix method for approximate analysis and solution of decision problems. Both of these ideas could provide valuable additions to a comprehensive automated system.

In the earlier SRI work[2], we developed a separate graphical system for deterministic analysis. A deterministic analysis is, of course, a mathematically trivial case of a probabilistic analysis and therefore can, in principle, be treated with an automated influence diagram procedure. Practically, however, the treatment of deterministic relations with the probabilistic procedure presented here is cumbersome and computationally inefficient. Expansion of influence diagram modeling and computational procedures to unify these two important aspects of analysis should be a fruitful direction for further work.

The work we performed during implementation has pointed out a neglected area of research. The development of an influence diagram is itself an iterative and interactive process. Many decisions are made regarding the development and evaluation of the influence diagram through the process of user-computer interaction; for example, where to continue expansion of the influence diagram, when to access numerical values, and when to stop expanding. Our ongoing use of influence diagrams has led to a set of intuitive criteria for making these model-building decisions. The pilot system uses these criteria in an implicit manner or explicitly asks for the user judgment. Research on this process and the model-building decisions (as opposed to the "decisionmakers" decisions) may lead to a more effective and efficient process and quantitative criteria for judging the authenticity of the resulting value lotteries. Such results may provide a key to the intelligent use of influence diagrams on a broad scale.

---

[1] "The Concept of Influence and Its Use in Structuring Complex Decision Problems," Ph.D. dissertation, Engineering-Economic Systems Department, Stanford University (October 1978).

---

[2] A.C. Miller, M.W. Merkhofer, R.A. Howard, J.E. Matheson, T.R. Rice, "Development of Automated Aids for Decision Analysis," SRI International, Menlo Park, CA (May 1976).

APPENDIX I-1

The Workspace DIAGRAM

```
       ∇START[⎕]∇
     ∇ START;⎕IO
[1]    INITIALIZE
[2]    →EXEC IF 1=1↑DΔSTATUS
[3]    →0 IF 3=1↑DΔSTATUS
[4]   LOOP:GETΔOPTION
[5]   EXEC:⍎MENUΔEXEC[OPTION;]
[6]    →LOOP IF∧/ 6 7 ≠OPTION
     ∇
       ∇INITIALIZE[⎕]∇
     ∇ INITIALIZE
[1]    INITIALIZEΔCONST
[2]    INITΔDATA
[3]    →(NEW,OLD)[DΔSTATUS←GETΔSTATUS]
[4]   NEW:INITIALIZEΔVARS
[5]    GETΔINIT
[6]    →0
[7]   OLD:FETCHΔPROBLEM
     ∇
       ∇GETΔOPTION[⎕]∇
     ∇ GETΔOPTION
[1]    0 1024 PUT 1024⍴' '
[2]   PRΔMENU:MENU
[3]    →PRΔMENU IF∧/ 1 2 3 4 5 6 7 ≠OPTION←(,⎕)[1]
     ∇
       MENU
OPTIONS:
1 = CONTINUE STRUCTURE ELICITATION
2 = MAKE CORRECTIONS TO EXISTING DIAGRAM
3 = DISPLAY VARIABLE DICTIONARY
4 = DISPLAY DIAGRAM
5 = DISPLAY DISK DIRECTORY
6 = EXIT AND STORE PROBLEM
7 = EXIT
PLEASE ENTER THE NUMBER OF THE DESIRED OPTION AND PRESS EXECUTE
       ∇INITIALIZEΔCONST[⎕]∇
     ∇ INITIALIZEΔCONST
[1]    TURNΔMAT← 8 2 ⍴'↑→↑←↓→↓←→↑→↓←↑←↓'
[2]    UTURNΔMAT← 4 2 ⍴'→←←→↑↓↓↑'
[3]    A←'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[4]    A←'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[5]    N←'1234567890'
[6]    ERRΔSYM←' ', 6 1 ⍴'∧↑|↑⍋Δ'
[7]    SCRΔSYM←'↑↓→←∘+⎕~'
[8]    ⎕IO←1
[9]    LOOPΔPAT←'I∨M∨.∧('
     ∇
       ∇INITΔDATA[⎕]∇
     ∇ INITΔDATA
[1]    ⍎(,DATAΔNAMES,'←'),'⍳0'
[2]    STATUS←'NEW'
[3]    ESCAPE←0
     ∇
```

```
        ∇GETΔSTATUS[⎕]∇
      ∇ STATUS←GETΔSTATUS
[1]   MENU:INITΔMENU
[2]    →MENU IF∧/ 1 2 ≠STATUS←1↑⎕
      ∇
        INITΔMENU
OPTIONS:
1 = START A NEW PROBLEM FROM SCRATCH
2 = RETRIEVE AN OLD PROBLEM FROM DISK AND CONTINUE
PLEASE ENTER THE NUMBER OF THE DESIRED OPTION AND PRESS EXECUTE
        ∇INITIALIZEΔVARS[⎕]∇
      ∇ INITIALIZEΔVARS
[1]    INFΔIND←INFΔLENGTH←INFΔCH←VARΔIND←VARS←⍳0
[2]    DELΔINFΔLIST←INFΔLIST← 0 2 ⍴NAMES← 0 18 ⍴' '
[3]    OLDΔSCR←1024⍴' '
[4]    OPTION←2×NEWΔSTART←1+POINTER←0
      ∇
        ∇GETΔINIT[⎕]∇
      ∇ GETΔINIT
[1]    BLK
[2]   GETΔSCR:'PLACE ON THE SCREEN THE OUTCOMES OF ULTIMATE INT
       EREST.'
[3]    'REMEMBER TO USE NUMERALS 0-9 ONLY.'
[4]    PAUSE
[5]    0 1024 PUT 1024⍴' '
[6]    OLDΔSCR←GET 0 1024
[7]    →GETΔSCR IF~∧/(VARS←NEWΔVARS←(OLDΔSCR≠' ')/OLDΔSCR)∊N
[8]    VARΔIND←OLDΔSCR⍳VARS
      ∇
        ∇FETCHΔPROBLEM[⎕]∇
      ∇ FETCHΔPROBLEM
[1]    GETΔDIRECTORY
[2]    →0 IF SELECTΔPROBLEM
[3]    GETΔPROBLEM
      ∇
        ∇GETΔDIRECTORY[⎕]∇
      ∇ GETΔDIRECTORY;IN
[1]    BLK
[2]    ⎕WA←1 ⎕SVO 'IN'
[3]    IN←'IN 11005'
[4]    ⎕WA←IN
[5]   GETΔDIR:DIRECTORY←IN
[6]    ⎕WA←⎕SVR 'IN'
      ∇
        ∇SELECTΔPROBLEM[⎕]∇
      ∇ Z←SELECTΔPROBLEM
[1]   PROBS:'THE PROBLEMS STORED ON THIS DISK ARE:'
[2]    (⍕ 6 1 ⍴⍳6),(6 3 ⍴' = '),DIRECTORY,[1] 'PROBLEM NOT ON T
       HIS DISK'
[3]    'ENTER THE NUMBER OF THE DESIRED PROBLEM.'
[4]    →PROBS IF∧/ 1 2 3 4 5 6 ≠PROB←(,⎕)[1]
[5]    →PROBS IF∧/'UNUSED'=6↑(DIRECTORY,[1] 24↑' ')[PROB;]
[6]    →Z←0 IF 6≠PROB
[7]    'PLEASE INSERT THE CORRECT DISK AND RESTART.'
[8]    ⎕WA←⎕SVR 'IN'
[9]    DΔSTATUS←3
[10]   →0×Z←1
      ∇
```

```
      ∇GETΔPROBLEM[⎕]∇
    ∇ GETΔPROBLEM;IN;RETΔCODE
[1]    BLK
[2]    ⎕WA←1 ⎕SVO 'IN'
[3]    IN←'IN 1100',(⍕5↑PROB)
[4]    →OK IF∧/0=RETΔCODE←IN
[5]    'WARNING: I/O ERROR'
[6]    'RETURN CODE = '⍕RETΔCODE
[7]    'OPERATION WILL BE RETRIED'
[8]  OK:GETΔDATA
[9]    ⎕WA←⎕SVR 'IN'
[10]   REDRAW
    ∇
      ∇GETΔDATA[⎕]∇
    ∇ GETΔDATA;I
[1]    I←1
[2]  OUTΔLOOP:⍎'→OK IF ⎕AV[256]≠1↑,',DATAΔNAMES[I;],'←IN'
[3]    ⍎DATAΔNAMES[I;],'←⍳0'
[4]    DATAΔNAMES[I;],' IS AN EMPTY VECTOR'
[5]  OK:→OUTΔLOOP IF(1↑⍴DATAΔNAMES)≥I←I+1
    ∇
      DATAΔNAMES
XΔSTATUS
INFΔIND
INFΔLENGTH
INFΔCH
VARΔIND
NAMES
VARS
INFΔLIST
POINTER
IMPΔINFΔLIST
MARΔVALS
MARΔLGN
ELICITΔLIST
CΔVALS
CONΔPROBS
CPΔLGN
CΔVALΔLGN
ESCAPE
VARΔPTR
CONΔCTR
NUMΔPREDS
CONΔPTR
UVF
MATRIX
ELICITΔVALS
TYPES
CΔCOMP
NUMΔCONS
STATUS
NAMEΔLIST
VALΔBUF
```

```
ULTIMATE
JOINT
        MENUΔEXEC
CONTINUE
EDIT
DICTIONARY
DIAGRAM
DISΔDIR
SAVEΔDIAG
→0
        ∇CONTINUE[⎕]∇
     ∇ CONTINUE
[1]     BLK
[2]   QUES:QUESTIONS[((V∊A),(V∊A̲),(V∊N))/⍳3;],V←VARS[POINTER←
        POINTER+1]
[3]     PAUSE
[4]     EDIT
     ∇
        QUESTIONS
PLACE ON THE SCREEN THE VARIABLES REQUIRED TO PREDICT:
PLACE ON THE SCREEN THE VARIABLES INFLUENCING DECISION:
PLACE ON THE SCREEN THE VARIABLES REQUIRED TO PREDICT OUTCOME:
        ∇PAUSE[⎕]∇
     ∇ PAUSE
[1]     960 38 PUT ' PRESS EXECUTE WHEN READY TO CONTINUE.'
[2]     ⎕WA←GET 960 1
[3]     960 38 PUT 38↑' '
     ∇
        ∇BLK[⎕]∇
     ∇ BLK
[1]     0 1024 PUT 1024⍴' '
[2]     ⎕WA←1 ⎕CC 0
     ∇
        ∇EDIT[⎕]∇
     ∇ EDIT
[1]     NEWΔINFΔLIST← 0 2 ⍴' '
[2]     →NEW IF NEWΔSTART
[3]     →PUTΔSCR IF OPTION≠2
[4]     BLK
[5]     'DO YOU WANT TO EDIT THE SCREEN? (Y OR N):'
[6]     →NEW IF 'N'=1↑⍞
[7]   PUTΔSCR: 0 1024 PUT OLDΔSCR
[8]     NEWΔSCR←GET 0 1024
[9]     →PUTΔSCR IF CHECKΔSCR
[10]    DELETE((ΔSCR[;1]∊A,A̲,N)∧ΔSCR[;2]='~')/ΔSCR
[11]    MOVE((ΔSCR[;1]∊A,A̲,N̄)∧ΔSCR[;2]='⎕')/ΔSCR
[12]    INSERT((ΔSCR[;1]=' ')∧(ΔSCR[;2]∊A,A̲,N)∧~ΔSCR[;2]∊VARS)/
        ΔSCR
[13]  NEW:NEWΔSTART←0
[14]    GETΔNAMES
[15]    GETΔINF
[16]    INSERTΔINF
     ∇
```

```
        ∇PUT[⎕]∇
      ∇ POS PUT SCR;N;CTL;DAT
[1]     ⎕WA←1 ⎕SVO N← 2 3 ⍴'CTLDAT'
[2]     CTL←'DISPLAY'
[3]     DAT←SCR
[4]     CTL←1,POS
[5]     ⎕WA←⎕SVR N
      ∇
        ∇GET[⎕]∇
      ∇ SCR←GET POS;N;CTL;DAT
[1]     ⎕WA←1 ⎕SVO N← 2 3 ⍴'CTLDAT'
[2]     CTL←'DISPLAY'
[3]     CTL←0,POS
[4]     SCR←DAT
[5]     ⎕WA←⎕SVR N
      ∇
        ∇CHECKΔSCR[⎕]∇
      ∇ Z←CHECKΔSCR;ERR;CORΔMAT;ALLΔVARS
[1]     OLDΔSCR MATCH NEWΔSCR
[2]     →ERROR IF(×ERR←1)∧∨/ΔSCR[;1]∊SCRΔSYM
[3]     →ERROR IF(×ERR←2)∧∨/(⍳⍴ALLΔVARS)≠ALLΔVARS⍳ALLΔVARS←(~
        NEWΔSCR∊SCRΔSYM,' ')/NEWΔSCR
[4]     →ERROR IF(×ERR←3)∧∨/(ΔSCR[;2]='⎕')∧1025=NEWΔSCR⍳ΔSCR[;1]
[5]     →ERROR IF(×ERR←4)∧∨/(ΔSCR[;2]='~')∧1025≠NEWΔSCR⍳ΔSCR[;1]
[6]     →ERROR IF(×ERR←5)∧∨/(ΔSCR[;1]∊A,A̲,N)∧ΔSCR[;2]∊A,A̲,N,' '
[7]     OLDΔSCR←NEWΔSCR
[8]     →Z←0
[9]    ERROR:BEEP
[10]    2 64 ⍴128↑SCRΔERR[ERR;]
[11]    PAUSE
[12]    Z←1
      ∇
        ∇MATCH[⎕]∇
      ∇ OLD MATCH NEW
[1]     ΔSCR←⍉(ΔSCRN/OLD),[0.1](ΔSCRN←OLD≠NEW)/NEW
[2]     ΔSCRN←ΔSCRN/⍳⍴ΔSCRN
      ∇
        SCRΔSYM
↑↓→←∘+⎕~
        SCRΔERR
ERROR: YOU HAVE ERASED PART OF AN INFLUENCE. TRY AGAIN.

ERROR: A VARIABLE APPEARS MORE THAN ONCE ON THE SCREEN. TRY
        AGAIN.
A ⎕ WAS PLACED OVER A VARIABLE, BUT THE VARIABLE WAS NOT MOVED.
        TRY AGAIN.
ERROR: THE ~ SYMBOL WAS USED TO MOVE A VARIABLE. TRY AGAIN.

ERROR: ONE VARIABLE WAS REPLACED BY ANOTHER OR A BLANK.
        TRY AGAIN.
```

```
      ∇DELETE[⎕]∇
    ∇ DELETE ΔLIST;DELΔINFΔLIST;COMP1;COMP2
[1]   →0 IF 0=×/ρΔLIST
[2]   DELΔINFΔLIST←(∨/INFΔLIST∊ΔLIST[;1])/⍳1↑ρINFΔLIST
[3]   DELETEΔINF
[4]   VARS←(COMP2←~VARS∊ΔLIST[;1])/VARS
[5]   VARΔIND←COMP2/VARΔIND
[6]   NAMES←COMP2⌿NAMES
[7]   POINTER←POINTER-+/POINTER↑COMP2
    ∇
      ∇DELETEΔINF[⎕]∇
    ∇ DELETEΔINF;POS;COMP;INFΔPOINT;NEWΔCOMP
[1]   →0 IF 0=1↑ρDELΔINFΔLIST
[2]   →EXPAND IF 0=1↑ρNEWΔCOMP←DELΔINFΔLIST>1↑ρINFΔLIST
[3]   COMP←(1↑ρNEWΔINFΔLIST)ρ1
[4]   COMP[(-1↑ρINFΔLIST)+NEWΔCOMP/DELΔINFΔLIST]←0
[5]   NEWΔINFΔLIST←COMP⌿NEWΔINFΔLIST
[6]   DELΔINFΔLIST←(~NEWΔCOMP)/DELΔINFΔLIST
[7]  EXPAND:EXPANDΔLISTS
[8]   COMP←(ρINFΔIND)ρCOMP1←(1↑ρINFΔLIST)ρI←1
[9]   COMP[FINDΔIND DELΔINFΔLIST]←0
[10]  COMP1[DELΔINFΔLIST]←0
[11]  INFΔIND←COMP/INFΔIND
[12]  INFΔCH←COMP/INFΔCH
[13]  INFΔLIST←COMP1⌿INFΔLIST
[14]  INFΔLENGTH←COMP1/INFΔLENGTH
[15]  DELΔINFΔLIST←⍳0
    ∇
      ∇EXPANDΔLISTS[⎕]∇
    ∇ EXPANDΔLISTS;EXPΔDEL
[1]   POINTS←INFΔIND[(INFΔCH[INDS]∊'∘+')/INDS←FINDΔIND
      DELΔINFΔLIST]
[2]   EXPΔDEL←1++/~(INFΔIND ALLΔIOTA POINTS)∘.≥+\INFΔLENGTH
[3]   NEWΔINFΔLIST←INFΔLIST[(EXPΔDEL←(~EXPΔDEL∊DELΔINFΔLIST)/
      EXPΔDEL);],[1] NEWΔINFΔLIST
[4]   DELΔINFΔLIST←DELΔINFΔLIST,EXPΔDEL
    ∇
      ∇MOVE[⎕]∇
    ∇ MOVE ΔLIST;DELΔINFΔLIST
[1]   →0 IF 0=×/ρΔLIST
[2]   NEWΔINFΔLIST←NEWΔINFΔLIST,[1](COMP←∨/INFΔLIST∊ΔLIST[;1])
      ⌿INFΔLIST
[3]   DELΔINFΔLIST←(COMP)/⍳ρCOMP
[4]   DELETEΔINF
[5]   VARΔIND[VARS⍳ΔLIST[;1]]←OLDΔSCR⍳ΔLIST[;1]
    ∇
      ∇INSERT[⎕]∇
    ∇ INSERT ΔLIST
[1]   →0 IF 0=×/ρΔLIST
[2]   VARS←VARS,NEWΔVARS←ΔLIST[;2]
[3]   VARΔIND←VARΔIND,OLDΔSCR⍳ΔLIST[;2]
    ∇
```

```
      ∇GETΔNAMES[⎕]∇
    ∇ GETΔNAMES;I
[1]    →0 IF 0=ρNEWΔVARS
[2]    BLK
[3]    I←1
[4]   ENTER:'ENTER THE NAME OF VARIABLE ',NEWΔVARS[I]
[5]    NAMES←NAMES,[1] 18↑⍞
[6]    →ENTER IF(ρNEWΔVARS)≥I←I+1
[7]    NEWΔVARS←0ρ' '
    ∇
      ∇GETΔINF[⎕]∇
    ∇ GETΔINF;BUF;BUF1
[1]   START: 0 1024 PUT 1024ρ' '
[2]    BUF1← 0 2 ρ' '
[3]   DESC:'DESCRIBE THE NEW INFLUENCES:'
[4]    →CHKΔLOOPS IF 0=ρBUF←⍞
[5]    BUF←PARSE BUF
[6]    BUF1←(~PREV←∨/BUF∧.=⍉INFΔLIST,[1] NEWΔINFΔLIST)⌿BUF
[7]    →CHKΔLOOPS IF 0=∨/PREV
[8]    BEEP
[9]    'THE FOLLOWING INFLUENCE(S) HAVE PREVIOUSLY BEEN INSERTE
       D:'
[10]   ,(PREV⌿BUF)[;,1],'→',(PREV⌿BUF)[;,2],';'
[11]   PAUSE
[12]  CHKΔLOOPS:→START IF CHECKΔLOOPS
[13]   ⎕WA←1 ⎕CC 0
[14]   NEWΔINFΔLIST←NEWΔINFΔLIST,[1] BUF1
    ∇
      ∇PARSE[⎕]∇
    ∇ LIST←PARSE BUF;POS
[1]   START:LIST←SQUASH BUF
[2]    →START IF~ 3 4 CHKΔMEM ';'
[3]    →START IF~ 1 2 CHKΔMEM '→←~'
[4]    →START IF~ 4 1 CHKΔMEM VARS,NEWΔVARS
[5]    →START IF~ 2 3 CHKΔMEM VARS,NEWΔVARS
[6]    →OK IF∧/POS←≠/LIST[; 1 3]
[7]    BUF←5 EDITΔINF POS
[8]    →START
[9]   OK:→END IF∧/POS←~((LIST[;1]∊N)∧(~LIST[;3]∊N)∧LIST[;
       2]='→')∨(~LIST[;1]∊N)∧(LIST[;3]∊N)∧LIST[;2]='←'
[10]   BUF←5 EDITΔINF POS
[11]   →START
[12]  END:LIST← 0 ¯1 ↓LIST
[13]   DELΔINF
[14]   →COMPRESS IF 0=1↑IND←(LIST[;2]≠'→')/⍳1↑ρLIST
[15]   LIST[IND;]←⌽LIST[IND;]
[16]  COMPRESS:LIST←LIST[; 1 3]
    ∇
      ∇SQUASH[⎕]∇
    ∇ LIST←SQUASH BUF
[1]    LIST←((1↑ 100 4 ⍴(ρBUF)[1]),4)ρBUF←(BUF≠1⌽BUF)/BUF←((BUF
       ≠' ')/BUF),';'
    ∇
```

```
      ∇CHKΔMEM[⎕]∇
    ∇ Z←CODE CHKΔMEM SET;POS
[1]   →0 IF Z←∧/POS←LIST[;CODE[2]]∊SET
[2]   BUF←CODE[1] EDITΔINF POS
    ∇
      ∇EDITΔINF[⎕]∇
    ∇ Z←ERRΔCODE EDITΔINF ERRΔPOS
[1]   ⎕WA←2 ⎕CC 2
[2]   ERRORS[ERRΔCODE;]
[3]   ,LIST
[4]   ERRΔSYM[ERRΔCODE;1+((4×ρERRΔPOS)ρ(-ERRΔCODE)⌽ 1 0 0 0)\~
      ERRΔPOS]
[5]   Z←GET 768 64
[6]   ⎕WA←1 ⎕CC 0
    ∇
      ERRORS
ERROR: INVALID INFLUENCE SYMBOL. EDIT LINE AND PRESS EXECUTE:
ERROR: UNDEFINED VARIABLE. EDIT LINE AND PRESS EXECUTE:
ERROR: INVALID DELIMITER. EDIT LINE AND PRESS EXECUTE:
ERROR: UNDEFINED VARIABLE. EDIT LINE AND PRESS EXECUTE:
ERROR: INVALID INFLUENCE. EDIT LINE AND PRESS EXECUTE:
      ∇DELΔINF[⎕]∇
    ∇ DELΔINF;DELΔINFΔCOMP;DELΔINFΔLIST
[1]   →0 IF 0=1↑ρDIΔLIST←(COMP←LIST[;2]='~')⌿LIST
[2]   LIST←(~COMP)⌿LIST
[3]   DELΔINFΔCOMP←∨⌿(DIΔLIST[; 1 3]∧.=IΔLIST)∨DIΔLIST[;
      3 1]∧.=IΔLIST←⍉INFΔLIST,[1] NEWΔINFΔLIST
[4]   DELΔINFΔLIST←DELΔINFΔCOMP/⍳¯1↑ρIΔLIST
[5]   DELETEΔINF
[6]   ⍝
    ∇
      ∇BEEP[⎕]∇
    ∇ BEEP
[1]   ⎕WA←2 ⎕CC 2
    ∇
      ∇CHECKΔLOOPS[⎕]∇
    ∇ Z←CHECKΔLOOPS
[1]   →(Z←0) IF∨/ 0 1 =ρVARS
[2]   ⎕WA←1 ⎕CC 0
[3]   M←INFΔMAT BUF1
[4]   →(Z←0) IF 0=LOOPS M
[5]   BEEP
[6]   'THE FOLLOWING VARIABLES ARE INVOLVED IN ONE OR MORE LOO
      PS:'
[7]   (' ',VARS)[1+((2×ρLOOPΔVARS)ρ 1 0)\LOOPΔVARS]
[8]   'PLEASE REDEFINE THE INFLUENCES.'
[9]   Z←1
[10]  PAUSE
    ∇
      ∇INFΔMAT[⎕]∇
    ∇ M←INFΔMAT BUF1;IΔLIST
[1]   M←((ρVARS)*2)ρ0
[2]   M[(ρVARS)⊥⍉((ρIΔLIST)ρ ¯1 0)+VARS⍳IΔLIST←INFΔLIST,[1]
      NEWΔINFΔLIST,[1] BUF1]←1
[3]   M←(2ρρVARS)ρM
    ∇
```

```
      ∇INSERTΔINF[⎕]∇
    ∇ INSERTΔINF;I;CONFLICTS;ΔSCR
[1]   REDRAW
[2]   →0 IF 0=×/ρNEWΔINFΔLIST
[3]   REDRAW
[4]   CONFLICTS←⍳0×I←1
[5]  DRW:KOMPUTELINE 16 64 LTA OLDΔSCR⍳NEWΔINFΔLIST[I;]
[6]   OLDΔSCR[LINE] MATCH CH
[7]   →OK IF∧/ΔSCR[;1]=' '
[8]   →OK IF FIXΔCON
[9]   CONFLICTS←CONFLICTS,I
[10]  →LOOPΔEND
[11] OK:STOREΔDATA
[12]  REDRAW
[13] LOOPΔEND:→DRW IF(1↑ρNEWΔINFΔLIST)≥I←I+1
[14]  REPORTΔCONFLICTS
[15]  NEWΔINFΔLIST← 0 2 ρ' '
    ∇
      ∇REDRAW[⎕]∇
    ∇ REDRAW
[1]   OLDΔSCR←1024ρ' '
[2]   OLDΔSCR[INFΔIND,VARΔIND]←INFΔCH,VARS
    ∇
      ∇KOMPUTELINE[⎕]∇
    ∇ KOMPUTELINE RC;T;Δ;SG;⎕IO;V:H
[1]   →(0∧.=SG←×Δ←-/[⎕IO←1] RC)/ERROR
[2]   Δ←|Δ
[3]   LINE←RC[1;2]+64×¯1+RC[1;1]+(-SG[1])×⍳T←Δ[1]-|SG[1]
[4]   CH←TρSG[1]↑'↑↓'
[5]   LINE←LINE,(64×¯1+RC[2;1])+RC[1;2]+(SG[2]×|SG[1])+(-SG[
      2])×⍳T←Δ[2]+¯1+|SG[1]
[6]   CH←CH,Tρ(-SG[2])↑'→←'
[7]   →0
[8]  ERROR:SΔCOMPUTELINE←2+1↑⎕LC
[9]   'POSITIONAL ERROR'
[10]  →1
    ∇
      ∇FIXΔCON[⎕]∇
    ∇ Z←FIXΔCON
[1]   →(Z←0) IF(∨/ΔSCR[;1]∊A,A̲,N)∨∨/,UTURNΔMAT∧.=⍉ΔSCR
[2]  TEST1:→TEST2 IF~∨/TURNS←∨⌿TURNSΔMAT←TURNΔMAT∧.=⍉ΔSCR
[3]   FIXΔCON1
[4]  TEST2:→TEST3 IF~∨/PLUSSES←ΔSCR[;1]='+'
[5]   FIXΔCON2
[6]  TEST3:→TEST4 IF~∨/JOTS←(ΔSCR[;1]='∘')∧ΔSCR[;2]∊'↑↓'
[7]   FIXΔCON3
[8]  TEST4:→END IF~∨/JOTS←(ΔSCR[;1]='∘')∧ΔSCR[;2]∊'→←'
[9]   FIXΔCON4
[10] END:Z←1
    ∇
```

```
      ∇FIXΔCON1[⎕]∇
    ∇ FIXΔCON1;PPOS;TPOS
[1]   TPOS←LINE[TURNS/ΔSCRN]
[2]   PPOS←TPOS∊INFΔIND[FINDΔIND INFΔLIST[;1] ALLΔIOTA
      NEWΔINFΔLIST[I;1]]
[3]   VTURNS←∨⌿TURNSΔMAT[⍳4;]
[4]   PPOS←TURNS/(TURNS\PPOS)∨VTURNS\('↑',CH)[VTURNS/ΔSCRN]∊'↑
      ↓'
[5]   INFΔCH[INFΔIND ALLΔIOTA PPOS/TPOS]←'+'
[6]   CH[LINE⍳PPOS/TPOS]←'+'
[7]   INFΔCH[INFΔIND ALLΔIOTA(~PPOS)/TPOS]←'∘'
[8]   CH[LINE⍳(~PPOS)/TPOS]←'∘'
    ∇
      ∇ALLΔIOTA[⎕]∇
    ∇ Z←LIST1 ALLΔIOTA LIST2
[1]   Z←(∨/(,LIST1)∘.=,LIST2)/⍳⍴,LIST1
    ∇
      ∇FIXΔCON2[⎕]∇
    ∇ FIXΔCON2
[1]   CH[PLUSSES/ΔSCRN]←'+'
    ∇
      ∇FIXΔCON3[⎕]∇
    ∇ FIXΔCON3
[1]   CH[JOTS/ΔSCRN]←'∘'
    ∇
      ∇FIXΔCON4[⎕]∇
    ∇ FIXΔCON4
[1]   PLUSSES←('↑',CH)[CHΔIND←JOTS/ΔSCRN]∊'↑↓'
[2]   CH[PLUSSES/CHΔIND]←'+'
[3]   INFΔCH[INFΔIND ALLΔIOTA LINE[PLUSSES/ΔSCRN]]←'+'
[4]   CH[(~PLUSSES)/CHΔIND]←'∘'
    ∇
      ∇STOREΔDATA[⎕]∇
    ∇ STOREΔDATA
[1]   INFΔIND←INFΔIND,LINE
[2]   INFΔLENGTH←INFΔLENGTH,⍴LINE
[3]   INFΔCH←INFΔCH,CH
[4]   INFΔLIST←INFΔLIST,[1] NEWΔINFΔLIST[I;]
    ∇
      ∇REPORTΔCONFLICTS[⎕]∇
    ∇ REPORTΔCONFLICTS
[1]   →0 IF 0=⍴CONFLICTS
[2]   BLK
[3]   'THE FOLLOWING INFLUENCES CANNOT BE DRAWN AND SO HAVE'
[4]   'NOT BEEN INSERTED. VARIABLES MUST BE MOVED AND/OR'
[5]   'OTHER INFLUENCES MUST BE REMOVED BEFORE THESE'
[6]   'INFLUENCES CAN BE ENTERED AGAIN.'
[7]   1⌽(⌽NEWΔINFΔLIST[CONFLICTS;]),'→'
[8]   PAUSE
    ∇
```

```
      ∇DICTIONARY[⎕]∇
    ∇ DICTIONARY;MAT;STMAT;STCOMP;DECMAT;DECOMP;OCMAT;OCCOMP
[1]   BLK
[2]   HEADER← 1 60 ρ60↑'STATE VARIABLES      DECISION VARIABLES
        OUTCOMES'
[3]   STMAT← 14 20 ↑(((+/STCOMP),1)ρSTCOMP/VARS),'=',(STCOMP←
      VARS∊A)≠NAMES
[4]   DECMAT← 14 20 ↑(((+/DECOMP),1)ρDECOMP/VARS),'=',(DECOMP←
      VARS∊A)≠NAMES
[5]   OCMAT← 14 20 ↑(((+/OCCOMP),1)ρOCCOMP/VARS),'=',(OCCOMP←
      VARS∊N)≠NAMES
[6]   0 896 PUT, 15 64 ↑HEADER,[1] STMAT,DECMAT,OCMAT
[7]   PAUSE
    ∇
      ∇DIAGRAM[⎕]∇
    ∇ DIAGRAM
[1]   REDRAW
[2]   0 1024 PUT OLDΔSCR
[3]   ⎕WA←GET 0
    ∇
      ∇DISΔDIR[⎕]∇
    ∇ DISΔDIR;DIRECTORY
[1]   GETΔDIRECTORY
[2]   (⍕ 5 1 ρ⍳5),(5 3 ρ' = '),DIRECTORY
[3]   PAUSE
    ∇
      ∇GETΔDIRECTORY[⎕]∇
    ∇ GETΔDIRECTORY;IN
[1]   BLK
[2]   ⎕WA←1 ⎕SVO 'IN'
[3]   IN←'IN 11005'
[4]   ⎕WA←IN
[5]  GETΔDIR:DIRECTORY←IN
[6]   ⎕WA←⎕SVR 'IN'
    ∇
      ∇SAVEΔDIAG[⎕]∇
    ∇ SAVEΔDIAG
[1]   GETΔDIRECTORY
[2]   SELECTΔFILE
[3]   PUTΔPROBLEM
    ∇
      ∇SELECTΔFILE[⎕]∇
    ∇ SELECTΔFILE
[1]  PROBS:'THE PROBLEMS STORED ON THIS DISK ARE:'
[2]   (⍕ 5 1 ρ⍳5),(5 3 ρ' = '),DIRECTORY
[3]   'ENTER THE NUMBER OF THE DESIRED PROBLEM.'
[4]   →PROBS IF∧/ 1 2 3 4 5 ≠PROB←(,⎕)[1]
[5]   'ENTER THE NAME OF THE PROBLEM (SEVEN CHARACTERS ONLY).'
[6]   PROBΔNAME←7↑⍞
[7]   BLK
[8]   CHGΔDIRECT
    ∇
```

```
      ∇CHGΔDIRECT[⎕]∇
    ∇ CHGΔDIRECT;OUT
[1]   ⎕WA←1 ⎕SVO 'OUT'
[2]   DIRECTORY[PROB;]←24↑PROBΔNAME
[3]   ⎕WA←1 ⎕SVO 'OUT'
[4]   OUT←'OUT 11005 ID=(DIRECT)'
[5]   OUT←DIRECTORY
[6]   ⎕WA←⎕SVR 'OUT'
    ∇
      ∇PUTΔPROBLEM[⎕]∇
    ∇ PUTΔPROBLEM;OUT;NUM
[1]   BLK
[2]  START:⎕WA←1 ⎕SVO 'OUT'
[3]   OUT←'OUT 1100',NUM,' ID=(SYS000',(NUM←⍕5+PROB),')'
[4]   →DONE IF PUTΔDATA
[5]   'WARNING: I/O ERROR.'
[6]   'RETURN CODE = ',⍕RETΔCODE
[7]   'OPERATION WILL BE RETRIED.'
[8]   ⎕WA←⎕SVR 'OUT'
[9]   →START
[10] DONE:⎕WA←⎕SVR 'OUT'
    ∇
      ∇PUTΔDATA[⎕]∇
    ∇ Z←PUTΔDATA;I
[1]   I←1
[2]  OUTΔLOOP:⍎'→STORE IF 0≠⍴,',DATAΔNAMES[I;]
[3]   ⍎DATAΔNAMES[I;],'←⎕AV[256]'
[4]  STORE:⍎'OUT←',DATAΔNAMES[I;]
[5]   →OK IF∧/0=1↑RETΔCODE←OUT
[6]   →Z←0
[7]  OK:→OUTΔLOOP IF(1↑⍴DATAΔNAMES)≥I←I+1
[8]   Z←1
    ∇
```

APPENDIX I-2

The Workspace XFORM

```
        ∇START[⎕]∇
      ∇ START
[1]     →INIT IF~REΔENTER
[2]     REΔENTER←0
[3]     →LOOP
[4]    INIT:INITIALIZE
[5]     FETCHΔPROBLEM
[6]     →0 IF∧/'QUIT'=4↑STATUS
[7]    LOOP:GETΔOPTION
[8]     ⍎MENUΔEXEC[OPTION;]
[9]     →LOOP IF∧/ 6 7 ≠OPTION
      ∇
        ∇REΔSTART[⎕]∇
      ∇ REΔSTART
[1]     REΔENTER←1
[2]     START
      ∇
        ∇INITIALIZE[⎕]∇
      ∇ INITIALIZE
[1]     ⎕IO←1
[2]     A←'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[3]     A←'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[4]     N←'0123456789'
[5]     IMPΔINFALIST← 0 2 ⍴' '
[6]     STATUS←'OK'
[7]     MAT←INFΔMAT INFALIST
      ∇
        MENU
OPTIONS:
1 = CONVERT DIAGRAM TO DECISION NETWORK
2 = CONVERT DIAGRAM TO DECISION TREE NETWORK
3 = DISPLAY VARIABLE DICTIONARY
4 = DISPLAY DIAGRAM
5 = DISPLAY DISK DIRECTORY
6 = EXIT AND STORE PROBLEM
7 = EXIT
PLEASE ENTER THE NUMBER OF THE DESIRED OPTION AND PRESS EXECUTE
        MENUΔEXEC
CONVΔDN
CONVΔDTN
DICTIONARY
DIAGRAM
DISΔDIR
SAVEΔDIAG
→0
```

```
        ∇CONVADN[⎕]∇
      ∇ CONVADN
[1]     BLK
[2]   ADD:ADDADDAIND
[3]     →ADD IF ADDADDAARB
[4]     BLK
[5]     'THE DIAGRAM IS NOW A DECISION NETWORK.'
[6]     WAIT 2
      ∇
        ∇ADDADDAIND[⎕]∇
      ∇ ADDADDAIND
[1]     →0 IF∨/ 0 1 =⍴DECALIST←(VARS∊A̲)/VARS
[2]     I←1
[3]   DECΔLOOP:→ENDΔLOOP IF(⍴DIR←(DIR∊A̲)/DIR←DIRΔPRED DECALIST[
        I])=⍴ALL←(ALL∊A̲)/ALL←ALLΔPRED DECALIST[I]
[4]     IMPΔINFALIST←IMPΔINFALIST,[1]⍉(COMP/ALL),[
        0.1](+/COMP←~ALL∊DIR)⍴DECALIST[I]
[5]   ENDΔLOOP:→DECΔLOOP IF(I←I+1)≥⍴DECALIST
      ∇
        ∇ALLΔPRED[⎕]∇
      ∇ Z←ALLΔPRED NODES;ALLΔSUC1;N
[1]     Z←(AP VARS⍳NODES)/VARS
      ∇
        ∇AP[⎕]∇
      ∇ Z←AP N;T
[1]     →(Z←0) IF 0=⍴,N
[2]     Z←T∨AP(T←DP N)/⍳1↑⍴MAT
      ∇
        ∇DIRΔPRED[⎕]∇
      ∇ Z←DIRΔPRED NODES
[1]     Z←(DP VARS⍳NODES)/VARS
      ∇
        ∇ADDADDAARB[⎕]∇
      ∇ Z←ADDADDAARB
[1]     DECΔMAT←COMP⌿(COMP←VARS∊A̲)/INFΔMAT INFALIST,[1]
        IMPΔINFALIST
[2]     DECΔVARS←COMP/VARS
[3]     →(Z←0) IF 1=+/COMP
[4]     →(Z←0) IF∧/,DECΔMAT
[5]     DECΔPAIR←DECΔVARS[1 1 +(⍴DECΔMAT)⊤¯1+(,DECΔMAT)⍳0]
[6]     'FOR A DECISION NETWORK THERE MUST BE AN INFLUENCE'
[7]     'BETWEEN DECISIONS ',PAIR[1],' AND ',PAIR[2],'.'
[8]   ENTER:'ENTER 1 FOR ',PAIR[1],'→',PAIR[2],', OR 2 FOR ',
        PAIR[1],'←',PAIR[2],':'
[9]     →ENTER IF∧/ 1 2 ≠DIRECTION←⎕
[10]    IMPΔINFALIST←IMPΔINFALIST,[1] DIRECTION⌽PAIR
[11]    →(Z←0) IF 1=+/~,DECΔMAT
[12]    Z←1
      ∇
```

```
     ∇WAIT[⎕]∇
    ∇ WAIT X;I
[1]    I←1
[2]   LOOP:→LOOP IF(6×X)≥I←I+1
    ∇
     ∇CONVΔDTN[⎕]∇
    ∇ CONVΔDTN
[1]    BLK
[2]    ADDΔSDΔIND
[3]    ADDΔDSΔIND
[4]    'THE DECISION NETWORK IS NOW A DECISION TREE NETWORK.'
[5]    WAIT 2
    ∇
     ∇ADDΔSDΔIND[⎕]∇
    ∇ ADDΔSDΔIND;I;DIR;ALL;COMP;DECALIST
[1]    DECΔLIST←(VARS∈A̲)/VARS
[2]    I←1
[3]   DECΔLOOP:→ENDΔLOOP IF(⍴DIR←DIRΔPRED DECALIST[I])=⍴ALL←
       ALLΔPRED DECALIST[I]
[4]    IMPΔINFΔLIST←IMPΔINFΔLIST,[1]⍉(COMP/ALL),[
       0.1](+/COMP←~ALL∈DIR)⍴DECΔLIST[I]
[5]   ENDΔLOOP:→DECΔLOOP IF(⍴DECΔLIST)≥I←I+1
    ∇
     ∇ADDΔDSΔIND[⎕]∇
    ∇ ADDΔDSΔIND;I;DIR;ALL;COMP;STΔLIST
[1]    STΔLIST←(VARS∈A,N)/VARS
[2]    I←1
[3]   STΔLOOP:→ENDΔLOOP IF(⍴DIR←(DIR∈A̲)/DIR←DIRΔPRED STΔLIST[I]
       )=⍴ALL←(ALL∈A̲)/ALL←ALLΔPRED STΔLIST[I]
[4]    IMPΔINFΔLIST←IMPΔINFΔLIST,[1]⍉(COMP/ALL),[
       0.1](+/COMP←~ALL∈DIR)⍴STΔLIST[I]
[5]   ENDΔLOOP:→STΔLOOP IF(⍴STΔLIST)≥I←I+1
    ∇
```

APPENDIX I-3

The Workspace PROBS

```
     ∇START[⎕]∇
   ∇ START
[1]    →INIT IF~REΔENTER
[2]    REΔENTER←0
[3]    →LOOP
[4]   INIT:INITIALIZE
[5]    FETCHΔPROBLEM
[6]    →0 IF∧/'QUIT'=4↑STATUS
[7]    →LOOP IF ESCAPE
[8]    →LOOP IF∧/'OLD'=3↑STATUS
[9]    NEWΔINIT
[10]  LOOP:GETΔOPTION
[11]   ⍎MENUΔEXEC[OPTION;]
[12]   →LOOP IF∧/ 6 7 ≠OPTION
   ∇
     ∇INITIALIZE[⎕]∇
   ∇ INITIALIZE
[1]    ⎕IO←1
[2]    CONΔCTR←VARΔPOINTER←1
[3]    A←'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[4]    A←'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[5]    N←'0123456789'
   ∇
     ∇NEWΔINIT[⎕]∇
   ∇ NEWΔINIT
[1]    BUILDΔLIST
[2]    NAMEΔLIST←NAMES[VARS⍳ELICITΔLIST;]
[3]    VARΔPTR←0
   ∇
     ∇BUILDΔLIST[⎕]∇
   ∇ BUILDΔLIST;COMP;MASKEDΔMAT
[1]    ELICITΔLIST←⍳0
[2]    MATRIX←INFΔMAT IMPΔINFΔLIST
[3]   LOOP:MASKEDΔMAT←MATRIX∧⍉(2⍴⍴VARS)⍴~VARS∊ELICITΔLIST
[4]    LISTΔBUF←((~VARS∊ELICITΔLIST)∧0=∨/MASKEDΔMAT)/VARS
[5]    LISTΔBUF←⎕←((~COMP)/LISTΔBUF),(COMP←LISTΔBUF∊A,N)/LISTΔBUF
[6]    ELICITΔLIST←ELICITΔLIST,LISTΔBUF
[7]    →LOOP IF(⍴ELICITΔLIST)≠⍴VARS
[8]    MATRIX←MATRIX[PERM←VARS⍳ELICITΔLIST;]
[9]    MATRIX←MATRIX[;PERM]
   ∇
     MENU
OPTIONS:
2 = ELICIT ULTIMATE VALUE COEFFICIENTS
2 = EDIT PROBABILITIES
3 = DISPLAY VARIABLE DICTIONARY
4 = DISPLAY DIAGRAM
5 = DISPLAY DISK DIRECTORY
6 = EXIT AND STORE PROBLEM
7 = EXIT
PLEASE ENTER THE NUMBER OF THE DESIRED OPTION AND PRESS EXECUTE
```

```
        ∇START[⎕]∇
      ∇ START
[1]     →INIT IF~REΔENTER
[2]     REΔENTER←0
[3]     →LOOP
[4]    INIT:INITIALIZE
[5]     FETCHΔPROBLEM
[6]     →0 IF∧/'QUIT'=4↑STATUS
[7]     →LOOP IF ESCAPE
[8]     →LOOP IF∧/'OLD'=3↑STATUS
[9]     NEWΔINIT
[10]   LOOP:GETΔOPTION
[11]    ⍎MENUΔEXEC[OPTION;]
[12]    →LOOP IF∧/ 6 7 ≠OPTION
      ∇
        ∇INITIALIZE[⎕]∇
      ∇ INITIALIZE
[1]     ⎕IO←1
[2]     CONΔCTR←VARΔPOINTER←1
[3]     A←'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[4]     A←'A̲B̲C̲D̲E̲F̲G̲H̲I̲J̲K̲L̲M̲N̲O̲P̲Q̲R̲S̲T̲U̲V̲W̲X̲Y̲Z̲'
[5]     N̲←'0̲1̲2̲3̲4̲5̲6̲7̲8̲9̲'
      ∇
        ∇NEWΔINIT[⎕]∇
      ∇ NEWΔINIT
[1]     BUILDΔLIST
[2]     NAMEΔLIST←NAMES[VARS⍳ELICITΔLIST;]
[3]     VARΔPTR←0
      ∇
        ∇BUILDΔLIST[⎕]∇
      ∇ BUILDΔLIST;COMP;MASKEDΔMAT
[1]     ELICITΔLIST←⍳0
[2]     MATRIX←INFΔMAT IMPΔINFΔLIST
[3]    LOOP:MASKEDΔMAT←MATRIX∧⍉(2⍴⍴VARS)⍴~VARS∊ELICITΔLIST
[4]     LISTΔBUF←((~VARS∊ELICITΔLIST)∧0=∨/MASKEDΔMAT)/VARS
[5]     LISTΔBUF←⍞←((~COMP)/LISTΔBUF),(COMP←LISTΔBUF∊A,N)/LISTΔBUF
[6]     ELICITΔLIST←ELICITΔLIST,LISTΔBUF
[7]     →LOOP IF(⍴ELICITΔLIST)≠⍴VARS
[8]     MATRIX←MATRIX[PERM←VARS⍳ELICITΔLIST;]
[9]     MATRIX←MATRIX[;PERM]
      ∇
        MENU
OPTIONS:
1 = CONTINUE PROBABILITY ELICITATION
2 = ELICIT ULTIMATE VALUE COEFFICIENTS
3 = DISPLAY VARIABLE DICTIONARY
4 = DISPLAY DIAGRAM
5 = DISPLAY DISK DIRECTORY
6 = EXIT AND STORE PROBLEM
7 = EXIT
PLEASE ENTER THE NUMBER OF THE DESIRED OPTION AND PRESS EXECUTE
```

```
        MENUΔEXEC
 CONTINUE
 GETΔUVF
 DICTIONARY
 DIAGRAM
 DISΔDIR
 SAVEΔPROBLEM
 →0
        ∇CONTINUE[⎕]∇
     ∇ CONTINUE
[1]    BLK
[2]    →INIT IF~ESCAPE
[3]    ESCAPE←0
[4]    →ELICITΔLOOP
[5]   INIT:VARΔINIT
[6]   ELICITΔLOOP:→(0×ESCAPE←1) IF∧/'ESCAPE'=6↑⍎GETΔFNS[TYPES[VARΔPTR];
       ]
[7]    ESCAPE←0
[8]    →ELICITΔLOOP IF NUMΔCONS≥CONΔCTR←CONΔCTR+1
[9]    CALCΔMAR
[10]   STOREΔPROBS
[11]   BLK
[12]   'PROBABILITY ELICITATION FOR VARIABLE ',ELICITΔLIST[VARΔPTR],' C
       OMPLETE.'
[13]   WAIT 1
     ∇
        ∇VARΔINIT[⎕]∇
     ∇ VARΔINIT
[1]    VARΔPTR←VARΔPTR+CONΔCTR←1+ESCAPE←0
[2]    NUMΔPREDS←+/COMP←MATRIX[;VARΔPTR]
[3]    CONΔPTR←COMP/⍳⍴COMP
[4]    CONΔLGN←MARΔLGN[CONΔPTR]
[5]    CONΔLGN[(TYPES=2)[CONΔPTR]/CONΔPTR]←3
[6]    NUMΔCONS←×/CONΔLGN
[7]    VALΔBUF←⍳0
[8]    BLK
[9]    →GETΔTYPE IF~ELICITΔLIST[VARΔPTR]∊A̲
[10]   TYPES←TYPES,3
[11]   →DD
[12]  GETΔTYPE:'IS VARIABLE ',ELICITΔLIST[VARΔPTR],'=',NAMEΔLIST[
       VARΔPTR;]
[13]   '1 = A DISCRETE VARIABLE'
[14]   '2 = A CONTINUOUS VARIABLE'
[15]   'ENTER 1 OR 2:'
[16]   →GETΔTYPE IF∧/ 1 2 ≠¯1↑TYPES←TYPES,1↑⎕
[17]   →DD IF∧/~((MATRIX DIRΔPRED(TYPES=2)[CONΔPTR]/CONΔPTR)/
       ELICITΔLIST)∊A̲
[18]   CALCΔCOMP
[19]   →0
[20]  DD:DDΔINIT
     ∇
```

```
        ∇CALCΔCOMP[⎕]∇
      ∇ CALCΔCOMP
[1]     RCΔCOMP←,CΔCOMP←((NUMΔPREDS,⍴ELICITΔLIST)⍴MATRIX[;VARΔPTR]∧
        ELICITΔLIST∊A)∧⍉MATRIX[;CONΔPTR]
[2]     RCΔCOMP[((⍴ELICITΔLIST),1)+.×(¯1+⍳NUMΔPREDS),[
        0.5] CONΔPTR]←1
[3]     CΔCOMP←(⍴CΔCOMP)⍴RCΔCOMP
      ∇
        ∇DDΔINIT[⎕]∇
      ∇ DDΔINIT
[1]     BLK
[2]     →0 IF 2=TYPES[VARΔPTR]
[3]     'PLEASE ENTER THE POSSIBLE VALUES FOR VARIABLE:'
[4]     ELICITΔLIST[VARΔPTR],' = ',NAMES[VARS⍳ELICITΔLIST[VARΔPTR];],'IN
         ONE LINE:'
[5]    NEW:NEWΔVALS←⎕
[6]     'ARE THESE VALUES CORRECT? (Y OR N)'
[7]     →NEW IF 'Y'≠1↑⍞
[8]     MARΔVALS←MARΔVALS,NEWΔVALS
[9]     MARΔLGN←MARΔLGN,⍴NEWΔVALS
      ∇
        GETΔFNS
GETΔDD
GETΔCONT
GETΔDD
        ∇GETΔDD[⎕]∇
      ∇ Z←GETΔDD
[1]     LOOPΔINIT
[2]     PUTΔDDΔSCR
[3]     Z←GETΔDDΔSCR
      ∇
        ∇LOOPΔINIT[⎕]∇
      ∇ LOOPΔINIT;ELV;INDS
[1]     →CALC IF NUMΔPREDS≠0
[2]     ELICITΔVALS←⍳0
[3]     →0
[4]    CALC:ELICITΔVALS←MARΔVALS[INDS←((+\MARΔLGN)-MARΔLGN[1])[CONΔPTR]+
        CTRS←COUNTERS CONΔCTR]
[5]     →0 IF 0=×/⍴DECS←(TYPES=3)[CONΔPTR]/CONΔPTR
[6]     →0 IF∧/~COMP←(MATRIX DIRΔSUCC DECS)∧ELICITΔLIST∊ELICITΔLIST[
        CONΔPTR]
[7]     CΔCTR←1
[8]    CΔLOOP:ELV←MARΔVALS[INDS[(COMP/CONΔPTR)[CΔCTR]]+1+(CΔCOMP[CΔCTR;]
        /MARΔLGN[CONΔPTR])⍳¯1+CΔCOMP[CΔCTR;]/CTRS]
[9]     ELICITΔVALS[(COMP/⍳⍴COMP)[CΔCTR]]←ELV
[10]    →CΔLOOP IF(+/COMP)≥CΔCTR←CΔCTR+1
      ∇
```

```
    ∇COUNTERS[⎕]∇
  ∇ Z←COUNTERS CTR;BASE
[1]   BASE←,MARALGN[CONAPTR]
[2]   BASE[(TYPES=2)[CONAPTR]/CONAPTR]←3
[3]   Z←((⍴BASE)⍴1)+BASE⊤CTR-1
  ∇
    ∇PUTADDASCR[⎕]∇
  ∇ PUTADDASCR;Z;ELICIT
[1]   PC←⎕AV[173]
[2]   BLK
[3]   0 64 PUT 64↑'PROBABILITY ELICITATION FOR ',ELICITALIST[VARAPTR],'
       = ',NAMEALIST[VARAPTR;]
[4]   128 64 PUT 64↑'CONDITIONAL ON VARIABLES                VALUES'
[5]   Z←((NUMAPREDS,1)⍴ELICIT),((NUMAPREDS,3)⍴' = '),(NUMAPREDS,
      32)↑NAMEALIST[VARS[ELICIT←ELICITALIST[CONAPTR]];]
[6]   Z←Z,(NUMAPREDS,28)↑2⍕(NUMAPREDS,1)⍴ELICITAVALS
[7]   192 384 PUT,Z
[8]   576 64 PUT 'PLEASE ENTER THE PROBABILITIES CORRESPONDING TO THE'
[9]   640 64 PUT 'FOLLOWING ',ELICITALIST[VARAPTR],' VALUES:'
[10]  704 64 PUT '(OR, TO GET THE MENU, ENTER: MENU).'
[11]  832 64 PUT⍕MARAVALS[((+\MARALGN)-MARALGN[1])[VARAPTR]+⍳MARALGN[
      VARAPTR]]
[12]  960 64 PUT 'ARE THE ABOVE VALUES CORRECT? (Y OR N): '
[13]  ⎕WA←1 ⎕CC 1
  ∇
    ∇GETADDASCR[⎕]∇
  ∇ Z←GETADDASCR
[1]  START:VALABUFFER←GET 896 64
[2]   →(0×⍴Z←'ESCAPE') IF∧/'MENU'=4↑VALABUFFER
[3]   VALABUFFER←⍎VALABUFFER
[4]   Z←'CONTINUE'
[5]   →START IF 'Y'≠GET 999 1
[6]   VALABUF←VALABUF,VALABUFFER
  ∇
    ∇GETACONT[⎕]∇
  ∇ Z←GETACONT
[1]   LOOPAINIT
[2]   PUTACOASCR
[3]   Z←GETACOASCR
  ∇
    ∇PUTACOASCR[⎕]∇
  ∇ PUTACOASCR;Z;ELICIT
[1]   PC←⎕AV[173]
[2]   BLK
[3]   0 64 PUT 64↑'PROBABILITY ELICITATION FOR ',ELICITALIST[VARAPTR],'
       = ',NAMEALIST[VARAPTR;]
[4]   128 64 PUT 64↑'CONDITIONAL ON VARIABLES                VALUES'
[5]   Z←((NUMAPREDS,1)⍴ELICIT),((NUMAPREDS,3)⍴' = '),(NUMAPREDS,
      32)↑NAMES[VARS[ELICIT←ELICITALIST[CONAPTR]];]
[6]   Z←Z,(NUMAPREDS,28)↑2⍕(NUMAPREDS,1)⍴ELICITAVALS
[7]   192 384 PUT,Z
[8]   576 64 PUT 'PLEASE ENTER THE ',ELICITALIST[VARAPTR],' VALUES COR
      RESPONDING TO THE'
[9]   640 64 PUT '10, 50, AND 90',PC,' POINTS FOR THE CUMULATIVE ',
      ELICITALIST[VARAPTR],' DISTRIBUTION'
[10]  704 64 PUT '(OR, TO GET THE MENU, ENTER: MENU).'
[11]  832 64 PUT '      10',PC,' (OR MENU): ',(10⍴' '),'50',PC,': ',(
      10⍴' '),'90',PC,': '
[12]  960 64 PUT 'ARE THE ABOVE VALUES CORRECT? (Y OR N): '
[13]  ⎕WA←1 ⎕CC 1
```

```
      ∇GETAΔCOΔSCR[⎕]∇
    ∇ Z←GETAΔCOΔSCR
[1]  START:VALABUFFER←GET 852 8
[2]   →(0×ρZ←'ESCAPE') IF∧/'MENU'=4↑VALABUFFER
[3]   VALABUFFER←(⍳VALABUFFER),⌽⍳(GET 882 8),' ',GET 867 8
[4]   Z←'CONTINUE'
[5]   →START IF 'Y'≠GET 999 1
[6]   VALABUF←VALABUF,VALABUFFER
    ∇
      ∇CALCAMAR[⎕]∇
    ∇ CALCAMAR
[1]   →0 IF 2≠TYPES[VARAPTR]
[2]   CAVALS←CAVALS,VALABUF
[3]   CAVALALGN←CAVALALGN,ρVALABUF
[4]   →CALC IF 0≠NUMAPREDS
[5]   MARAVALS←MARAVALS,VALABUF
[6]   MARALGN←MARALGN,3
[7]   →0
[8]  CALC:MARALGN←MARALGN,LGN←3××/MARALGN[(TYPES=3)[CONAPTR]/CONAPTR]
[9]   MARAVALS←MARAVALS,LGNρ(¯1 0 1 ×(0.5×-/EXT)÷
      1.8132)+0.5×+/EXT←(1,((¯2+ρVALABUF)ρ0),1)/VALABUF←VALABUF[⍋
      VALABUF]
    ∇
      ∇STOREAPROBS[⎕]∇
    ∇ STOREAPROBS
[1]   →0 IF 2=TYPES[VARAPTR]
[2]   CONAPROBS←CONAPROBS,VALABUF
[3]   CPALGN←CPALGN,ρVALABUF
    ∇
      ∇SAVEAPROBLEM[⎕]∇
    ∇ SAVEAPROBLEM
[1]   GETADIRECTORY
[2]   SELECTAFILE
[3]   PUTAPROBLEM
    ∇
```

APPENDIX I-4

The Workspace SOLVE

```
        ∇START[⎕]∇
      ∇ START
[1]     →INIT IF~REΔENTER
[2]     REΔENTER←0
[3]     →LOOP
[4]    INIT:FETCHΔPROBLEM
[5]     →0 IF∧/'QUIT'=4↑STATUS
[6]    LOOP:GETΔOPTION
[7]     ⍎MENUΔEXEC[OPTION;]
[8]     →LOOP IF∧/ 6 7 ≠OPTION
      ∇
        MENU
OPTIONS:
1 = CALCULATE JOINT AND ULTIMATE VALUE
2 = DISPLAY VARIABLE DICTIONARY
3 = DISPLAY DIAGRAM
4 = DISPLAY DISK DIRECTORY
5 = EXIT AND STORE PROBLEM
6 = EXIT
PLEASE ENTER THE NUMBER OF THE DESIRED OPTION AND PRESS EXECUTE
        MENUΔEXEC
CALCULATE
DICTIONARY
DIAGRAM
DISΔDIR
SAVEΔPROBLEM
→0
        ∇CALCULATE[⎕]∇
      ∇ CALCULATE
[1]     CALCΔJOINT
[2]     CALCΔULTIMATE
      ∇
```

```
      ∇CALCΔJOINT[⎕]∇
    ∇ CALCΔJOINT;DIM;IVAL;IJOINT;COMP;CONΔCOMP;CONΔVALS;BINS;
      LIMΔDEL;PROB
[1]   NΔDIM←ρDIM←MARALGN
[2]   PROB←1
[3]   DIM[(TYPES=2)/⍳ρTYPES]←3
[4]   JOINT←DIMρ0
[5]   NUMΔVALS←×/DIM
[6]   IΔJOINT←1
[7]  JOINTΔLOOP:DDΔCTRS←((COMP←TYPES≠2)/CTRS←CALCΔCTRS IΔJOINT
      )+(+\CPΔLGN)-CPΔLGN[1]
[8]   INDS←COMP/⍳ρCOMP
[9]  ⍝LOOP OVER DD VARS WHICH HAVE DIRECT PREDECESSORS
[10]  →CON IF 0=ρINDS
[11]  IVAL←1
[12] DDΔLOOP:→ENDΔDD IF∧/~CONΔCOMP←MATRIX DIRΔPRED IND←INDS[
      IVAL]
[13]  DDΔCTRS[IND]←DDΔCTRS[IND]+MARALGN[IND]×(CONΔCOMP/DIM)⊥((
      +/CONΔCOMP)ρ¯1)+CONΔCOMP/CTRS
[14] ENDΔDD:→DDΔLOOP IF(+/COMP)≥IVAL←IVAL+1
[15]  →ENDΔJOINT IF 0=PROB←×/CONΔPROBS[DDΔCTRS]
[16] CON:→ENDΔJOINT IF 0=ρINDS←(~COMP)/⍳ρCOMP
[17] ⍝LOOP OVER CONTINUOUS VARIABLES WHICH HAVE PREDECESSORS
[18]  IVAL←1
[19] CONΔLOOP:→ENDΔCON IF∧/~CONΔCOMP←MATRIX DIRΔPRED INDS[IVAL
      ]
[20]  DECΔCOMP←(ELICITΔVALS∊A)∧CONΔCOMP
[21]  CONΔVALS←CΔVALS[(-/(+\ĈΔVALΔLGN)[IVAL,1]+⍳3+3×(CONΔCOMP/
      DIM)⊥((+/CONΔCOMP)ρ¯1)+CONΔCOMP/CTRS]
[22]  MARΔVAL←MARΔVALS[(-/(+\MARΔLGN)[IND,1])+CTRS[IND←INDS[
      IVAL]]+3×(DECΔCOMP/DIM)⊥((+/DECΔCOMP)ρ¯1)+DECΔCOMP/CTRS
[23]  LIMΔDEL← ¯1 1 ×1.8132×-/CONΔVALS[3 2]
[24]  PROB←PROB× 0 0.25 0.5 0.25 0[1++/MARΔVAL>,CONΔVALS[
      1 3 ∘.+LIMΔDEL]
[25] ENDΔCON:→CONΔLOOP IF(+/COMP)≥IVAL←IVAL+1
[26]  ⍎'JOINT[',(,((NΔDIM,5)ρ'CTRS['),((NΔDIM,1)ρ⍳NΔNIM),(
      NΔDIM,2)ρ'];'),']←PROB'
[27]  →ENDΔJOINT:JOINTΔLOOP IF NUMΔVALS≥IΔJOINT←IΔJOINT+1
    ∇
```

```
      ∇CALCΔULTIMATE[⎕]∇
    ∇ CALCΔULTIMATE;DIM;IVAL;IULT;COMP;VΔCTRS
[1]    DIM←MARΔLGN
[2]    DIM[(TYPES=2)/⍳⍴TYPES]←3
[3]    ULTIMATE←DIM⍴0
[4]    NUMΔVALS←×/DIM
[5]    IULT←1
[6]   ULTΔLOOP:VΔCTRS←CTRS←CALCΔCTRS IVAL
[7]    VΔCTRS←VΔCTRS+(+\MARΔLGN)-MARΔLGN[1]
[8]    IVAL←1
[9]    →CALC IF 0=⍴INDS←(COMP←TYPES=2)/⍳⍴TYPES
[10]  VALΔLOOP:→ENDΔVAL IF∧/~DECΔCOMP←(ELICITΔVALS∊A̲)∧MATRIX
       DIRΔPRED INDS[IVAL]
[11]   VΔCTRS[INDS[IVAL]]←VΔCTRS[INDS[IVAL]]+3×(DECΔCOMP/DIM)⍳(
       (+/DECΔCOMP)⍴¯1)+DECΔCOMP/CTRS
[12]   →VALΔLOOP IF(+/COMP)≥IVAL←IVAL+1
[13]  CALC:ULTIMATE←+/UVF×(ELICITΔLIST∊0)/MARΔVALS[VΔCTRS]
[14]   →ULTΔLOOP IF NUMΔVALS≥IULT←IULT+1
    ∇
```

```
      ∇TREE[⎕]∇
    ∇ EV←TREE G
[1]   EV←⍎(,'+⌈'[1+C[G]],[1.1](⍴G)⍴'/'),'(⌽G)⍉VP'
    ∇
      ∇TREE∆POLICY[⎕]∇
    ∇ TREE∆POLICY G;V;Y;K;N;M;I;L;D;H;P;B
[1]   V←(H←⌽G)⍉VP
[2]   Y←(×/⍴V)⍴1
[3]   V TVAL D←C[G]
[4]   ⎕←B← 2 1 ⍴' '
[5]   ⎕←LOT←Y/[1](,V÷P),[1.1],P←H⍉PR
[6]   K←1-D[1+(⍴C)-⍳⍴C]⍳1
[7]   N←1⌈×/M←K↑⍴V
[8]   I←((L←+/Y)⍴1=⍳N)/Y/⍳⍴Y
[9]   POL←⍉(K,0)↓1+(⍴V)⊤¯1+I
[10]  Z←(¯1+K+⍴C)↓'+⌈'[1+D]
[11]  Z←,⍉Z,[0.1](⍴Z)⍴'/'
[12]  POL←POL,[2](,(⍎Z,'V')÷⍎Z,'P')[⌈I÷×/(K-1)↑⍴V]
[13]  B
[14]  ,⍉(0,K)↓(((⍴G)⍴' '),[0.1] 'SD'[D+1]),[1] 1 0 ⍕G
[15]  ' '
[16]  (((2×K+⍴G)⍴ 3 0), 10 2)⍕POL
    ∇
      ∇SHOWTREE[⎕]∇
    ∇ SHOWTREE G;V;P;X;R;N;M;K;L;Q;A;B;I;H
[1]   K←8
[2]   L←10
[3]   N←×/⍴V←(H←⌽G)⍉VP
[4]   P←H⍉PR
[5]   X←(N,1)⍴' '
[6]   I←⍴G
[7]  L1:M←×/I↑⍴V
[8]   R←⍎(Q←'+⌈'[C[G[I]]+1]),'/P'
[9]   B←(N,K)⍴,((M,K)⍴((K,2)⍕V÷P)),[2](M,K×¯1+N÷M)⍴' '
[10]  V←⍎Q,'/V'
[11]  A←(N,L)⍴,((M,L)⍴((L,3)⍕,P÷(⍴P)⍴⍉((¯1↑⍴P),×/⍴R)⍴R)),[
      2](M,L×¯1+N÷M)⍴' '
[12]  X←A,[2] B,[2] X
[13]  →(I=1)⍴L2
[14]  P←R
[15]  I←I-1
[16]  →L1
[17] L2:V
[18]  ' '
[19]  X
    ∇
```

```
      ∇LOTT[⎕]∇
    ∇ LOTT;II;B;K;J;W;M;L;N;CP
[1]    II←1+⌊(AA[⍋AA]−MIN)÷DEL
[2]    K←II⍳J←⍳W←1+¯1↑II
[3]    M←(L←(K≤⍴II))/K
[4]    M←(¯1+1↓M),⍴AA
[5]    II←0⍴0
[6]    N←L/J
[7]    CP←(+\PR[⍋AA])[M]
[8]    Y←(⍴J)⍴0
[9]    Y[N]←CP−(0,¯1↓CP)
[10]   X←MIN+DEL×¯1+⍳W
[11]   Z←+\Y
[12]   ⎕←+/AA×PR
[13]   ⎕← 5 0 8 3 8 3 ⍕⍉(3,⍴X)⍴X,Z,Y
    ∇
```

CONTRACT DISTRIBUTION LIST
(Unclassified Technical Reports)

Director Advanced Research Projects Agency Attention: Program Management Office 1400 Wilson Boulevard Arlington, Virginia 22209	2 copies
Office of Naval Research Attention: Code 455 800 North Quincy Street Arlington, Virginia 22217	3 copies
Defense Documentation Center Attention: DDC-TC Cameron Station Alexandria, Virginia 22314	12 copies
DCASMA Baltimore Office Attention: Mr. K. Gerasim 300 East Joppa Road Towson, Maryland 21204	1 copy
Director Naval Research Laboratory Attention: Code 2627 Washington, D.C. 20375	6 copies

Revised August 1978

SUPPLEMENTAL DISTRIBUTION LIST
(Unclassified Technical Reports)

Department of Defense

Director of Net Assessment
Office of the Secretary of Defense
Attention: MAJ Robert G. Gough, USAF
The Pentagon, Room 3A930
Washington, DC 20301

Assistant Director (Net Technical Assessment)
Office of the Deputy Director of Defense Research and Engineering (Test and Evaluation)
The Pentagon, Room 3C125
Washington, DC 20301

Director, Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

Director, Cybernetics Technology Office
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

Director, ARPA Regional Office (Europe)
Headquarters, U.S. European Command
APO New York 09128

Director, ARPA Regional Office (Pacific)
Staff CINCPAC, Box 13
APO San Francisco 96610

Dr. Don Hirta
Defense Systems Management School
Building 202
Ft. Belvoir, VA 22060

Chairman, Department of Curriculum Development
National War College
Ft. McNair, 4th and P Streets, SW
Washington, DC 20319

Defense Intelligence School
Attention: Professor Douglas E. Hunter
Washington, DC 20374

Vice Director for Production
Management Office (Special Actions)
Defense Intelligence Agency
Room 1E863, The Pentagon
Washington, DC 20301

Command and Control Technical Center
Defense Communications Agency
Attention: Mr. John D. Hwang
Washington, DC 20301

Department of the Navy

Office of the Chief of Naval Operations (OP-951)
Washington, DC 20450

Office of Naval Research
Assistant Chief for Technology (Code 200)
800 N. Quincy Street
Arlington, VA 22217

Office of Naval Research (Code 230)
800 North Quincy Street
Arlington, VA 22217

Office of Naval Research
Naval Analysis Programs (Code 431)
800 North Quincy Street
Arlington, VA 22217

Office of Naval Research
Operations Research Programs (Code 434)
800 North Quincy Street
Arlington, VA 22217

Office of Naval Research
Information Systems Program (Code 437)
800 North Quincy Street
Arlington, VA 22217

Director, ONR Branch Office
Attention: Dr. Charles Davis
536 South Clark Street
Chicago, IL 60605

Director, ONR Branch Office
Attention: Dr. J. Lester
495 Summer Street
Boston, MA 02210

Director, ONR Branch Office
Attention: Dr. E. Gloye
1030 East Green Street
Pasadena, CA 91106

Director, ONR Branch Office
Attention: Mr. R. Lawson
1030 East Green Street
Pasadena, CA 91106

Office of Naval Research
Scientific Liaison Group
Attention: Dr. M. Bertin
American Embassy - Room A-407
APO San Francisco 96503

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps (Code RD-1)
Washington, DC 20380

Headquarters, Naval Material Command
(Code 0331)
Attention: Dr. Heber G. Moore
Washington, DC 20360

Head, Human Factors Division
Naval Electronics Laboratory Center
Attention: Mr. Richard Coburn
San Diego, CA 92152

Dean of Research Administration
Naval Postgraduate School
Attention: Patrick C. Parker
Monterey, CA 93940

Superintendent
Naval Postgraduate School
Attention: R. J. Roland, (Code 52R1)
C^3 Curriculum
Monterey, CA 93940

Naval Personnel Research and Development
Center (Code 305)
Attention: LCDR O'Bar
San Diego, CA 92152

Navy Personnel Research and Development
Center
Manned Systems Design (Code 311)
Attention: Dr. Fred Muckler
San Diego, CA 92152

Naval Training Equipment Center
Human Factors Department (Code N215)
Orlando, FL 32813

Naval Training Equipment Center
Training Analysis and Evaluation Group
(Code N-00T)
Attention: Dr. Alfred F. Smode
Orlando, FL 32813

Director, Center for Advanced Research
Naval War College
Attention: Professor C. Lewis
Newport, RI 02840

Naval Research Laboratory
Communications Sciences Division (Code 5403)
Attention: Dr. John Shore
Washington, DC 20375

Dean of the Academic Departments
U.S. Naval Academy
Annapolis, MD 21402

Chief, Intelligence Division
Marine Corps Development Center
Quantico, VA 22134

Department of the Army

Alan H. Curry
Operations and Management Science Division
U.S. Army Institute for Research in Management Information and Computer Science
730 Peachtree St., N.E. (Suite 900)
Atlanta, Georgia 30308

Deputy Under Secretary of the Army
(Operations Research)
The Pentagon, Room 2E621
Washington, DC 20310

Director, Army Library
Army Studies (ASDIRS)
The Pentagon, Room 1A534
Washington, DC 20310

U.S. Army Research Institute
Organizations and Systems Research Laboratory
Attention: Dr. Edgar M. Johnson
5001 Eisenhower Avenue
Alexandria, VA 22333

Director, Organizations and Systems Research Laboratory
U.S. Army Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333

Technical Director, U.S. Army Concepts Analysis Agency
8120 Woodmont Avenue
Bethesda, MD 20014

Director, Strategic Studies Institute
U.S. Army Combat Developments Command
Carlisle Barracks, PA 17013

Commandant, Army Logistics Management Center
Attention: DRXMC-LS-SCAD (ORSA)
Ft. Lee, VA 23801

Department of Engineering
United States Military Academy
Attention: COL A. F. Grum
West Point, NY 10996

Marine Corps Representative
U.S. Army War College
Carlisle Barracks, PA 17013

Chief, Studies and Analysis Office
Headquarters, Army Training and Doctrine Command
Ft. Monroe, VA 23351

Commander, U.S. Army Research Office (Durham)
Box CM, Duke Station
Durham, NC 27706

Department of the Air Force

Assistant for Requirements Development and Acquisition Programs
Office of the Deputy Chief of Staff for Research and Development
The Pentagon, Room 4C331
Washington, DC 20330

Air Force Office of Scientific Research
Life Sciences Directorate
Building 410, Bolling AFB
Washington, DC 20332

Commandant, Air University
Maxwell AFB, AL 36112

Chief, Systems Effectiveness Branch
Human Engineering Division
Attention: Dr. Donald A. Topmiller
Wright-Patterson AFB, OH 45433

Deputy Chief of Staff, Plans, and Operations
Directorate of Concepts (AR/XOCCC)
Attention: Major R. Linhard
The Pentagon, Room 4D 1047
Washington, DC 20330

Director, Advanced Systems Division (AFHRL/AS)
Attention: Dr. Gordon Eckstrand
Wright-Patterson AFB, OH 45433

Commander, Rome Air Development Center
Attention: Mr. John Atkinson
Griffis AFB
Rome, NY 13440

IRD, Rome Air Development Center
Attention: Mr. Frederic A. Dion
Griffis AFB
Rome, NY 13440

HQS Tactical Air Command
Attention: LTCOL David Dianich
Langley AFB, VA 23665

Other Government Agencies

Chief, Strategic Evaluation Center
Central Intelligence Agency
Headquarters, Room 2G24
Washington, DC 20505

Director, Center for the Study of Intelligence
Central Intelligence Agency
Attention: Mr. Dean Moor
Washington, DC 20505

Mr. Richard Heuer
Methods & Forecasting Division
Office of Regional and Political Analysis
Central Intelligence Agency
Washington, DC 20505

Office of Life Sciences
Headquarters, National Aeronautics and Space Administration
Attention: Dr. Stanley Deutsch
600 Independence Avenue
Washington, DC 20546

Other Institutions

Department of Psychology
The Johns Hopkins University
Attention: Dr. Alphonse Chapanis
Charles and 34th Streets
Baltimore, MD 21218

Institute for Defense Analyses
Attention: Dr. Jesse Orlansky
400 Army Navy Drive
Arlington, VA 22202

Director, Social Science Research Institute
University of Southern California
Attention: Dr. Ward Edwards
Los Angeles, CA 90007

Perceptronics, Incorporated
Attention: Dr. Amos Freedy
6271 Variel Avenue
Woodland Hills, CA 91364

Stanford University
Attention: Dr. R. A. Howard
Stanford, CA 94305

Director, Applied Psychology Unit
Medical Research Council
Attention: Dr. A. D. Baddeley
15 Chaucer Road
Cambridge, CB 2EF
England

Department of Psychology
Brunel University
Attention: Dr. Lawrence D. Phillips
Uxbridge, Middlesex UB8 3PH
England

Decision Analysis Group
Stanford Research Institute
Attention: Dr. Miley W. Merkhofer
Menlo Park, CA 94025

Decision Research
1201 Oak Street
Eugene, OR 97401

Department of Psychology
University of Washington
Attention: Dr. Lee Roy Beach
Seattle, WA 98195

Department of Electrical and Computer Engineering
University of Michigan
Attention: Professor Kan Chen
Ann Arbor, MI 94135

Department of Government and Politics
University of Maryland
Attention: Dr. Davis B. Bobrow
College Park, MD 20747

Department of Psychology
Hebrew University
Attention: Dr. Amos Tversky
Jerusalem, Israel

Dr. Andrew P. Sage
School of Engineering and Applied Science
University of Virginia
Charlottesville, VA 22901

Professor Raymond Tanter
Political Science Department
The University of Michigan
Ann Arbor, MI 48109

Professor Howard Raiffa
Morgan 302
Harvard Business School
Harvard University
Cambridge, MA 02163

Department of Psychology
University of Oklahoma
Attention: Dr. Charles Gettys
455 West Lindsey
Dale Hall Tower
Norman, OK 73069

Institute of Behavioral Science #3
University of Colorado
Attention: Dr. Kenneth Hammond
Room 201
Boulder, Colorado 80309

LME