

147-053

(P)

AD A121849

**DOCUMENTATION OF
DECISION-AIDING SOFTWARE:
INFER SYSTEM SPECIFICATION**
DECISIONS AND DESIGNS INC.

Dorothy M. Amey
Phillip H. Feuerwerger
Roy M. Gulick

June 1979

N00014-79-C-0069

DTIC
NOV 29 1982
H

ADVANCED 
DECISION TECHNOLOGY
PROGRAM

CYBERNETICS TECHNOLOGY OFFICE
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY
Office of Naval Research • Engineering Psychology Programs

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

82 11 26 198

UNC FILE COPY

DOCUMENTATION OF DECISION-AIDING SOFTWARE: INFER SYSTEM SPECIFICATION

by

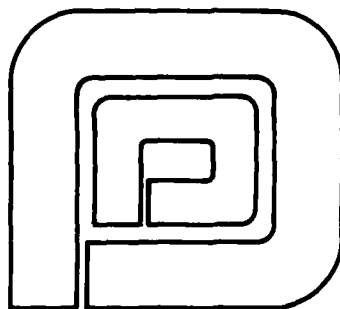
Dorothy M. Amey, Phillip H. Feuerwerger, and Roy M. Gulick

Sponsored by

Defense Advanced Research Projects Agency
ARPA Order 3469

June 1979

DTIC
NOV 29 1982
H



DECISIONS and DESIGNS, INC.

Suite 600, 8400 Westpark Drive
P.O. Box 907
McLean, Virginia 22101
(703) 821-2828

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

CONTENTS

	<u>Page</u>
FIGURES	iii
1.0 INTRODUCTION	1
1.1 Purpose of the System Specification	1
1.2 References	1
1.3 Terms	2
1.3.1 INFER	2
1.3.2 Terms	2
2.0 DESIGN DETAILS	3
2.1 Background	3
2.2 General Operating Procedures	3
2.3 System Logical Flow	3
2.4 HIPO Documentation	6

Accession For	
NEW CPRI	<input checked="" type="checkbox"/>
INDEXED	<input type="checkbox"/>
UNCLASSIFIED	<input type="checkbox"/>
<i>Added on file</i>	
Distribution	
Availability	
Dist	<i>A</i>



FIGURES

<u>Figure</u>		<u>Page</u>
2-1	LEGEND OF HIPO SYMBOLS	5
2-2	OVERALL DIAGRAM AND VISUAL TABLE OF CONTENTS	7

INFER SYSTEM SPECIFICATION

1.0 INTRODUCTION

1.1 Purpose of the System Specification

The INFER System Specification is a technical document written for software development personnel. Together with the INFER Functional Description, the System Specification guides the software development effort by identifying the functional requirements and by providing structured logic diagrams that depict the flow, control, and processing of information within the system.

The System Specification is generic documentation intended to guide and facilitate the preparation of the language-specific and computer hardware-specific program documentation and coding that are necessary to implement and operate INFER at an installation.

1.2 References

- 1.2.1 IBM, HIPO--A Design Aid and Documentation Technique. Technical Publication GC20-1851-0. White Plains, New York: IBM, October 1974.
- 1.2.2 Amey, Dorothy M.; Feuerwerger, Phillip H.; Gulick, Roy M. Documentation of Decision-Aiding Software: INFER Functional Description. McLean, Virginia: Decisions and Designs, Inc., June 1979.
- 1.2.3 Amey, Dorothy M.; Feuerwerger, Phillip H.; Gulick, Roy M. Documentation of Decision-Aiding

Software: INFER Users Manual. McLean, Virginia:
Decisions and Designs, Inc., June 1979.

1.3 Terms

1.3.1 INFER - The name of the system, INFER, is an abbreviation for inference, reflecting the logical process supported by the software.

1.3.2 HIPO - The specification uses the standard Hierarchy plus Input-Process-Output (HIPO) diagramming technique to depict the structural design and logical flow of the system. A legend explaining the HIPO diagramming symbols is included. Reference 1.2.1 provides a complete description of the HIPO documentation technique.

2.0 DESIGN DETAILS

2.1 Background

Systems development personnel should refer to the INFER Functional Description, reference 1.2.2, in conjunction with the documentation contained in this specification. The Functional Description details the inference models implemented by INFER and discusses the specific functions that the software must perform. In addition, systems development personnel may wish to refer to the INFER User's Manual, Reference 1.2.3.

2.2 General Operating Procedures

INFER is a menu-driven system. That is, the system is designed to interact with the user by presenting a sequential hierarchy of menus and asking the user to respond by selecting one option from the current menu. If the user does not select one of the menu options, the system displays the previous menu. In this manner, the user moves up and down the hierarchy, as desired. Whenever data entry is required as a result of option selection, the system specifically requests the data and specifies the format.

The system is also designed to be generally forgiving of procedural errors by the user.

2.3 System Logical Flow

INFER is a hierarchically structured, modular software system. The system structure and logical flow lends itself to presentation in the form of HIPO diagrams, which are contained in this document.

The main purpose of the HIPO diagrams is to provide, in a pictorial manner, the complete set of modular elements necessary to the operation of INFER including all input, output, and internal functional processing. This is done by displaying input items to the process step which uses them, defining the process, and showing the resulting output of the process step.

The HIPO documentation diagrams are designed and drawn in a hierarchical fashion from the main calling routines to the detail-level operation/calculation routines. Extended written descriptions are given below a HIPO diagram whenever it is deemed necessary.

A complete explanation of the symbolic notation used in the HIPO diagrams appears in reference 1.2.1. An abbreviated legend for the symbols used in this specification is presented in Figure 2-1. Note that:

- a. External system subroutines are depicted partly in the Process block and partly out. Internal subroutines are shown within the Process block.
- b. Overview diagrams show general inputs and outputs only, whereas detail/subroutine-level diagrams show specific input/output tables and/or displays.
- c. Rectangular boxes inside the Input/Output block areas are generally used to denote single data items. Two or more boxes are grouped to show that several data items are input/output.
- d. Rectangular boxes inside the Process block indicate repetitive subprocesses.

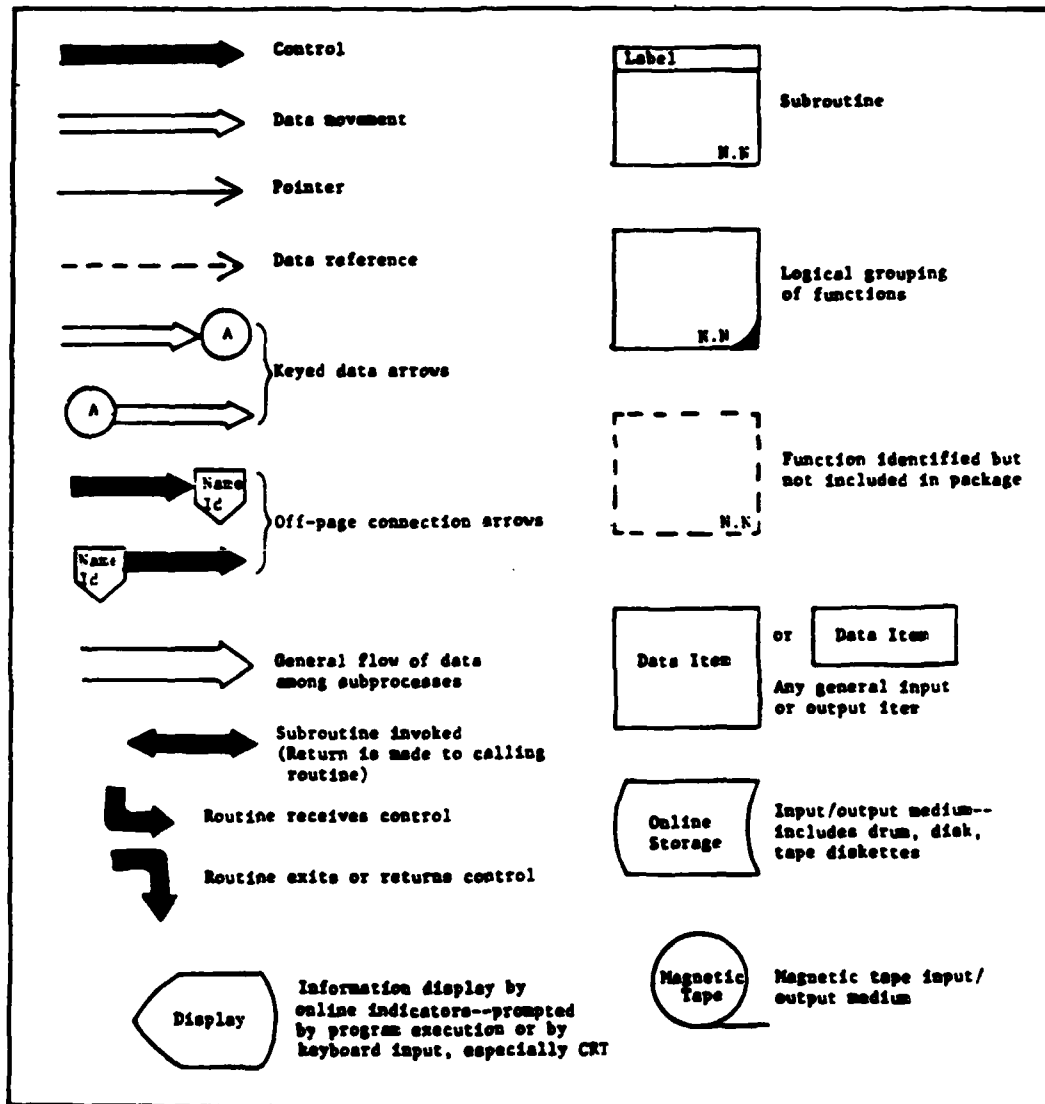


Figure 2-1
 LEGEND OF HIPO SYMBOLS

The HIPO diagrams appear in the next section, which completes the System Specification.

2.4 HIPO Documentation

The HIPO diagram identification numbers and figure numbers used in this section stand alone; i.e., they start with 1.0, increase hierarchically, and are independent of the numbering scheme used to this point in this document.

Figure 2-2 is a system structure chart and represents the overall program logic flow in a visual table of contents. The Visual Table of Contents diagram shows the hierarchical structure, the functional description labels, and the diagram (chart) identifiers of functions of INFER.

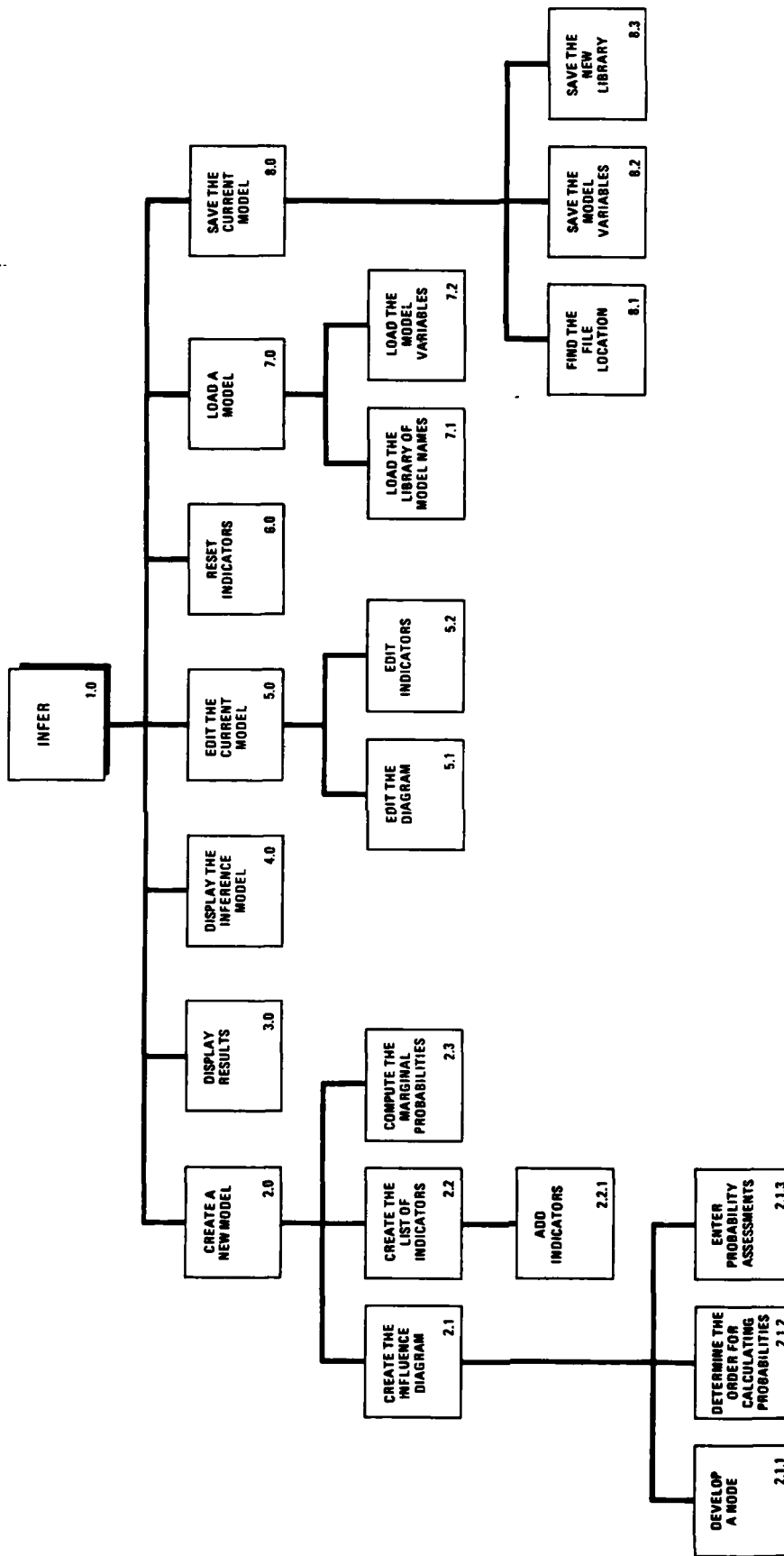
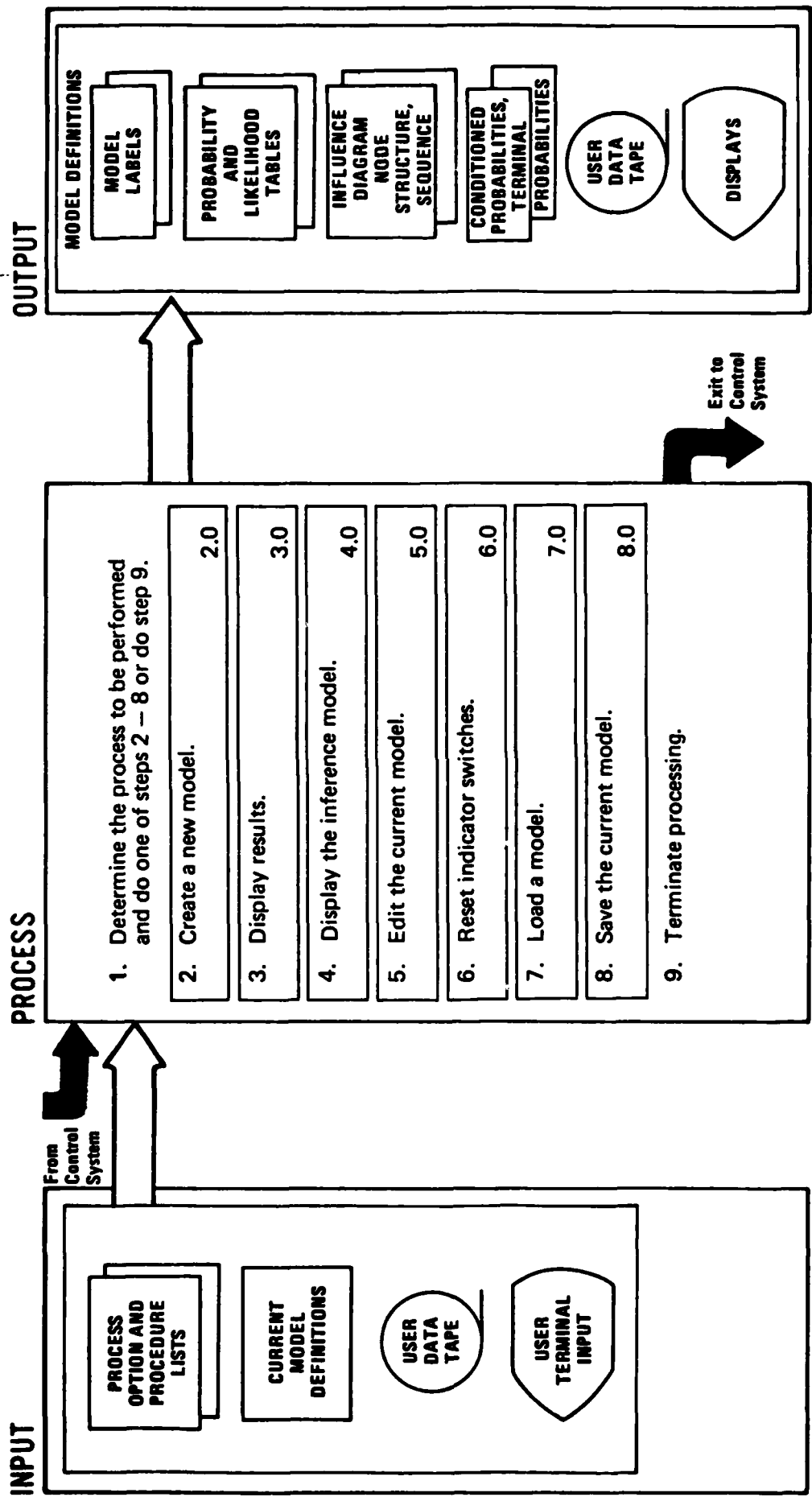


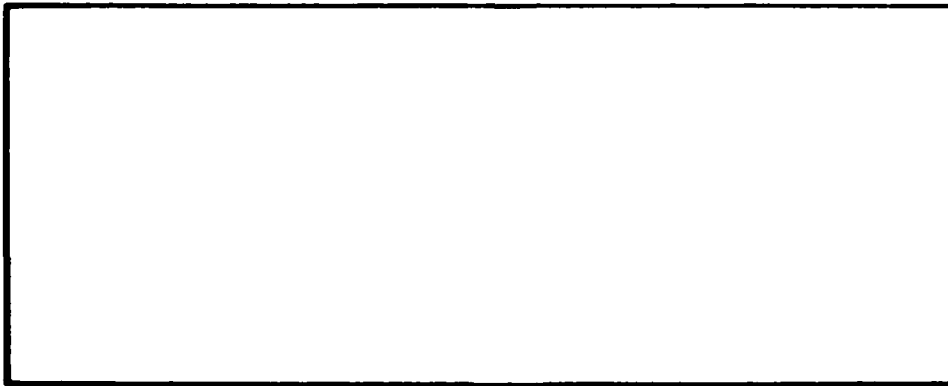
Figure 2.2
 INFER OVERVIEW DIAGRAM
 AND VISUAL TABLE OF CONTENTS



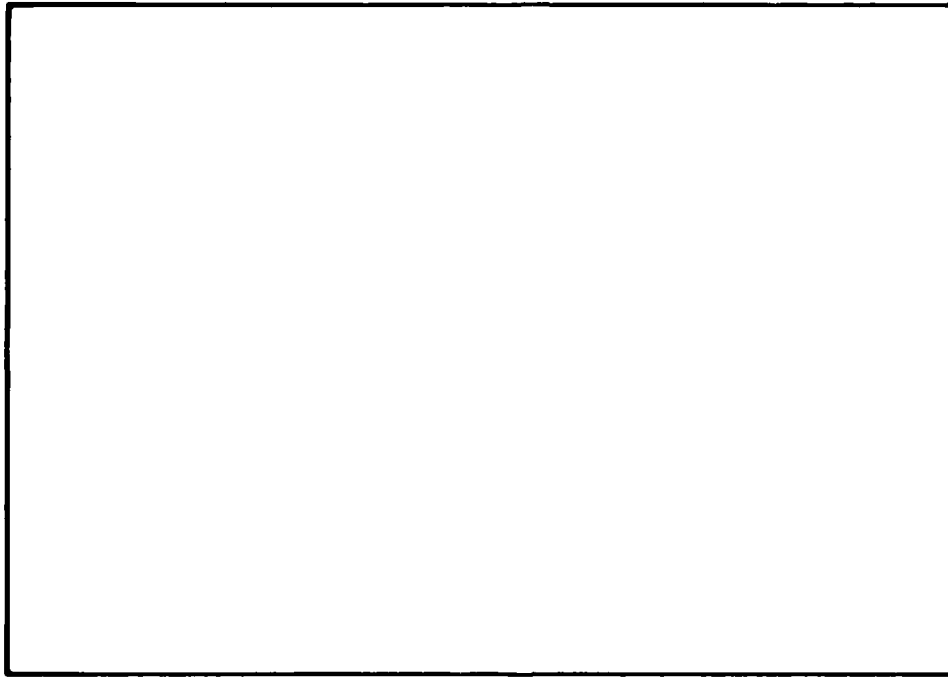
Extended Description

- The available processes are displayed via a descriptive list (MENU) and the user is prompted for the desired process. If no selection is made by the user, then processing stops at step 9 and control is returned to the system (or the calling program).
- A new model is created with user interaction. A model is completely defined by the new event and indicator labels, node structure table, node calculation sequence vector, probabilities and indicator likelihood vectors.
- The key (terminal) event node probabilities are displayed.
- Any specified event node and its set of influencing nodes are displayed along with the associated conditional and conditioned probabilities.
- Labels or values of the model are being modified. Indicators may be added to the model.
- The "On/Off" status of indicators may be changed.
- An existing model can be loaded from storage (tape).

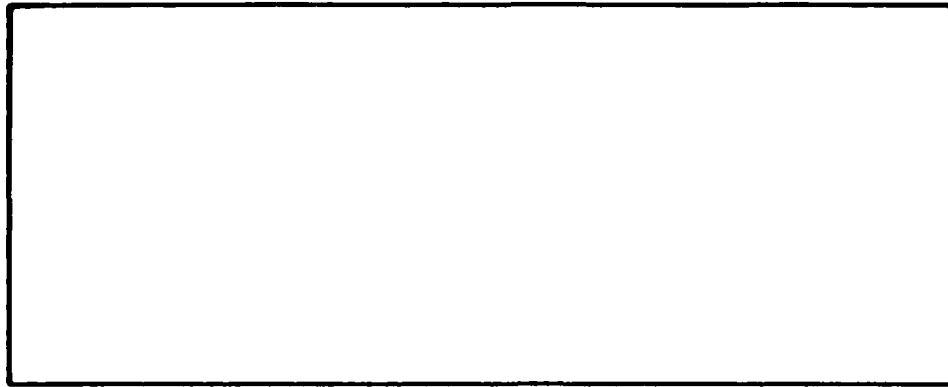
INPUT



PROCESS



OUTPUT

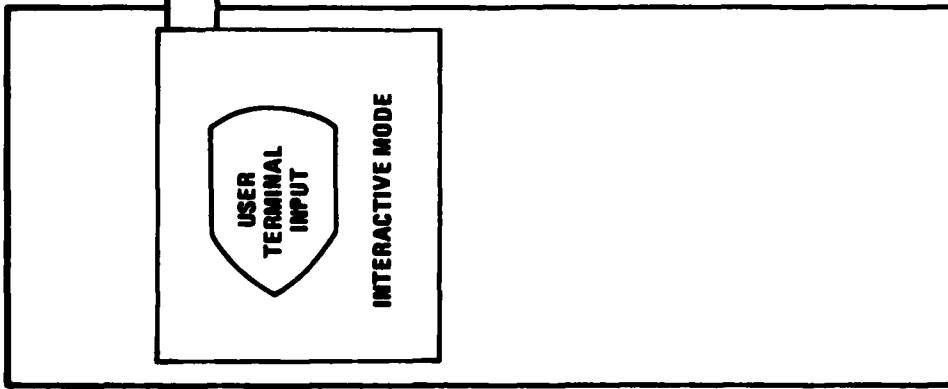


Extended Description (continued)

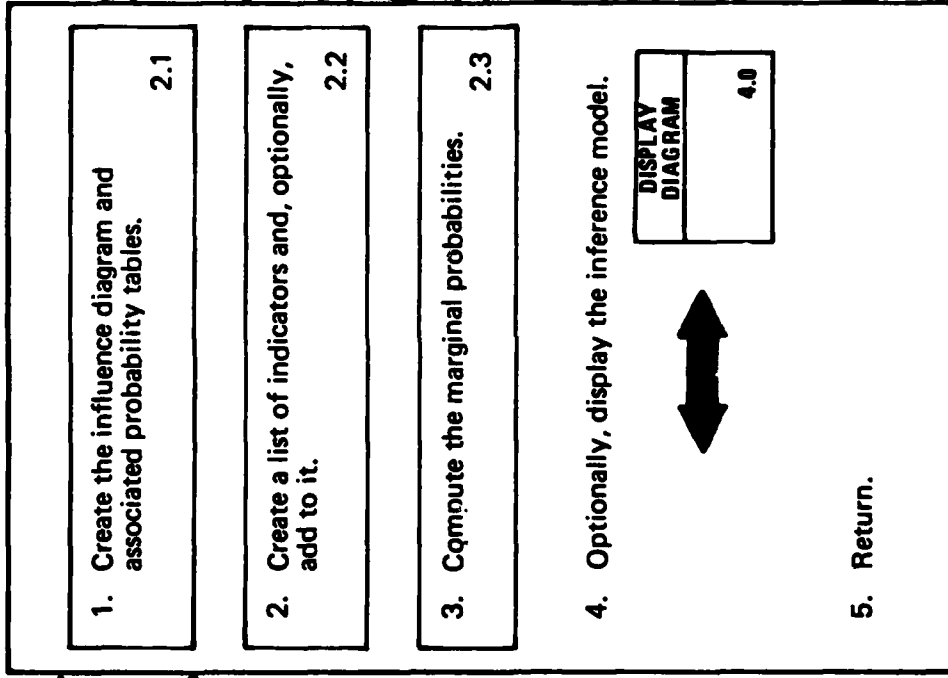
8. The current model variables are stored.

Diagrams 2.0 - 8.0 further depict and describe the processes invoked by steps 2 - 8 of the program. Steps 3 - 6 and 8 require that either step 2 or step 7 has been performed.

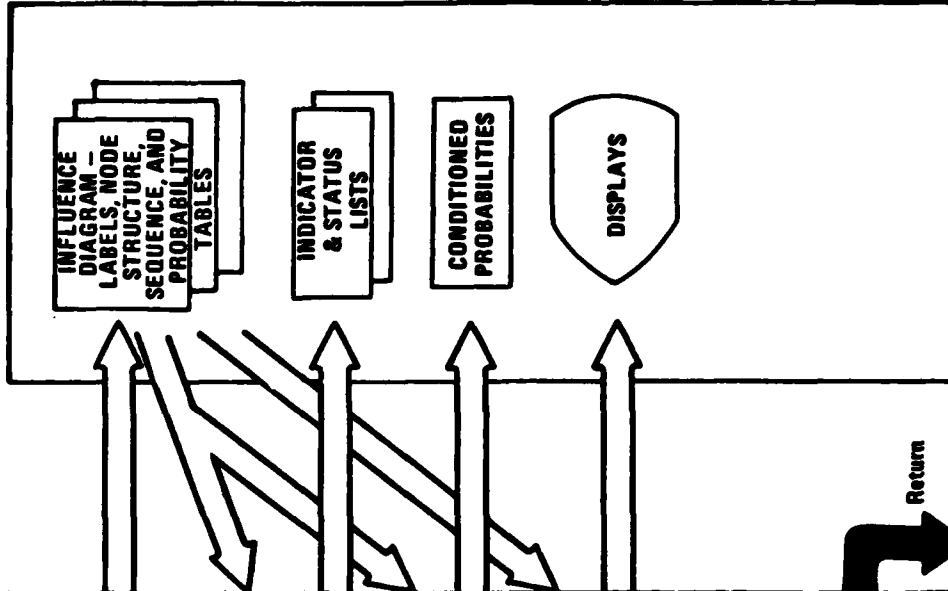
INPUT



PROCESS



OUTPUT



Extended Description

To create a new model, steps 1 through 3 must be performed before returning to the calling routine.

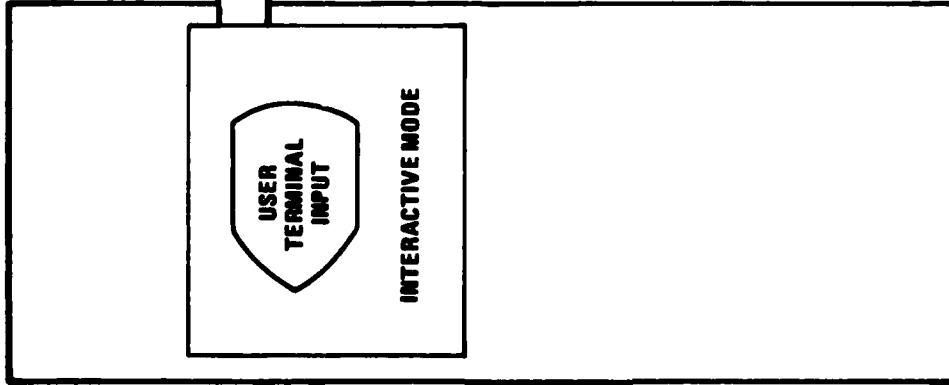
1. The influence diagram is a sequenced set of event-node structure labels that are depicted at subordinate levels of influence. The associated probabilities at each level of influence are elicited after the node sequence is determined.

2. Elicitation of indicator names and the event node which they condition is performed.

3. The final (terminal) set of conditioned probabilities is computed. Intermediate node level marginal probabilities are also derived during this computation and these values are stored in the conditioned probabilities table.

4. The user may optionally review the diagram just created at this point.

INPUT

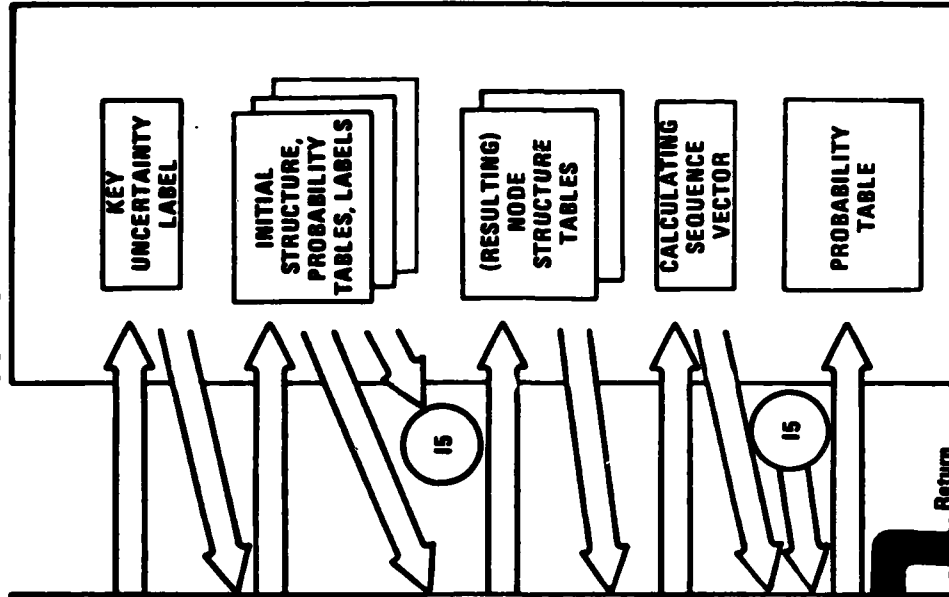


Extended Description

PROCESS

1. Elicit the name for the key uncertainty (terminal event).
2. Create tables for the node structure, probabilities, and event and outcome labels.
3. Develop a node into the diagram and repeat until no new nodes are specified.
 - 2.1.1
4. Determine the order for eliciting and calculating probabilities.
 - 2.1.2
5. Enter the outcome probabilities.
 - 2.1.3

OUTPUT



1. The key uncertainty is the terminal event node for which a final set of conditioned probabilities will be computed.

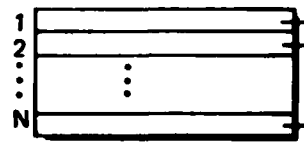
2. A node structure table is required to enable the cross-referencing of influencing nodes. It contains a row for each node defined in the model. Associated probability and label arrays are also needed. The probability matrices are constructed with the same numbers of rows as the maximum number of influencing events for any given event node. These matrices have as many columns as the maximum number of outcomes allowed for any event (see figure 2.1).

3. When a node is developed into the influence diagram, the user is requested to input label names for event nodes and labels for the outcomes of these nodes.

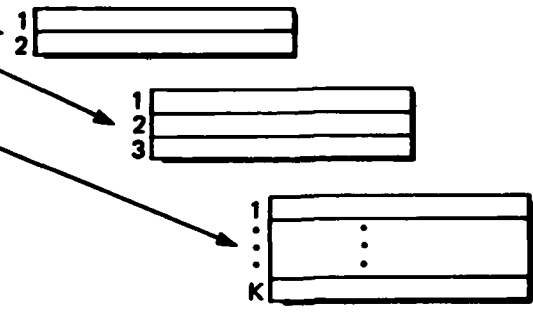
The node structure table has at least five values per node defined. These values are the node numbers of directly influencing nodes (usually 2), the number of influencing nodes specified, the number of associated outcome probabilities, and the number of direct outcomes of the particular event.

LABELS

EVENT NAMES



OUTCOME NAMES for events (=N arrays)

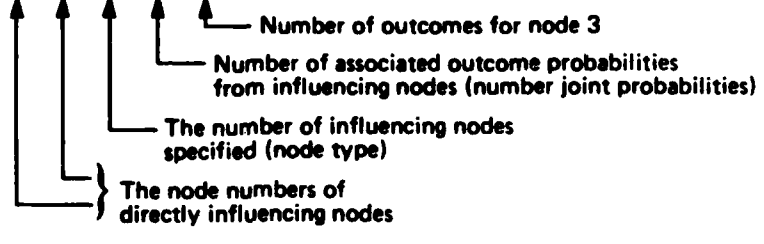


NODE STRUCTURE TABLE

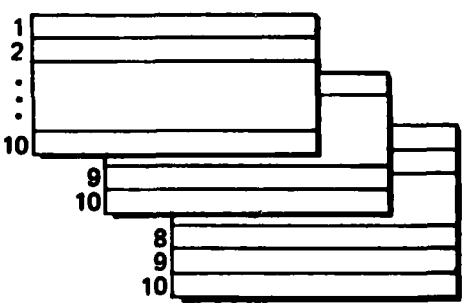
1					
2					
3					
...					
N	0	0	0	0	3

ROW 3

4	5	2	10	5
---	---	---	----	---



PROBABILITY TABLES



N Probability arrays

CONDITIONED

1				
2				
3				
...				
10				

Conditional probabilities

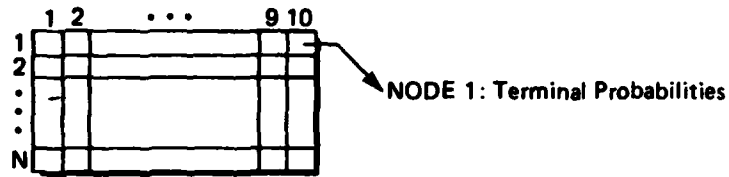
UNCONDITIONED

1	a	b	c	...	P1
2	0	0	0
...					
10	0	0	0

Directly assessed value

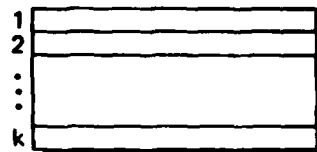
**HIPO Figure 2.1
INFER MODEL DEFINITION TABLES**

CONDITIONED PROBABILITIES

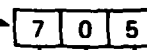
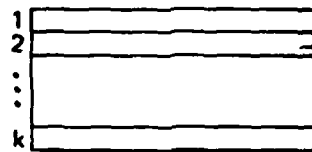


INDICATOR TABLES

LABELS

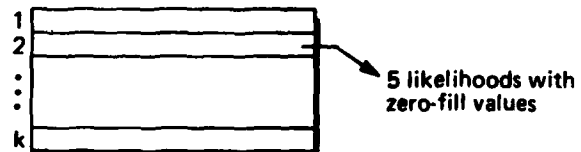


STATUS

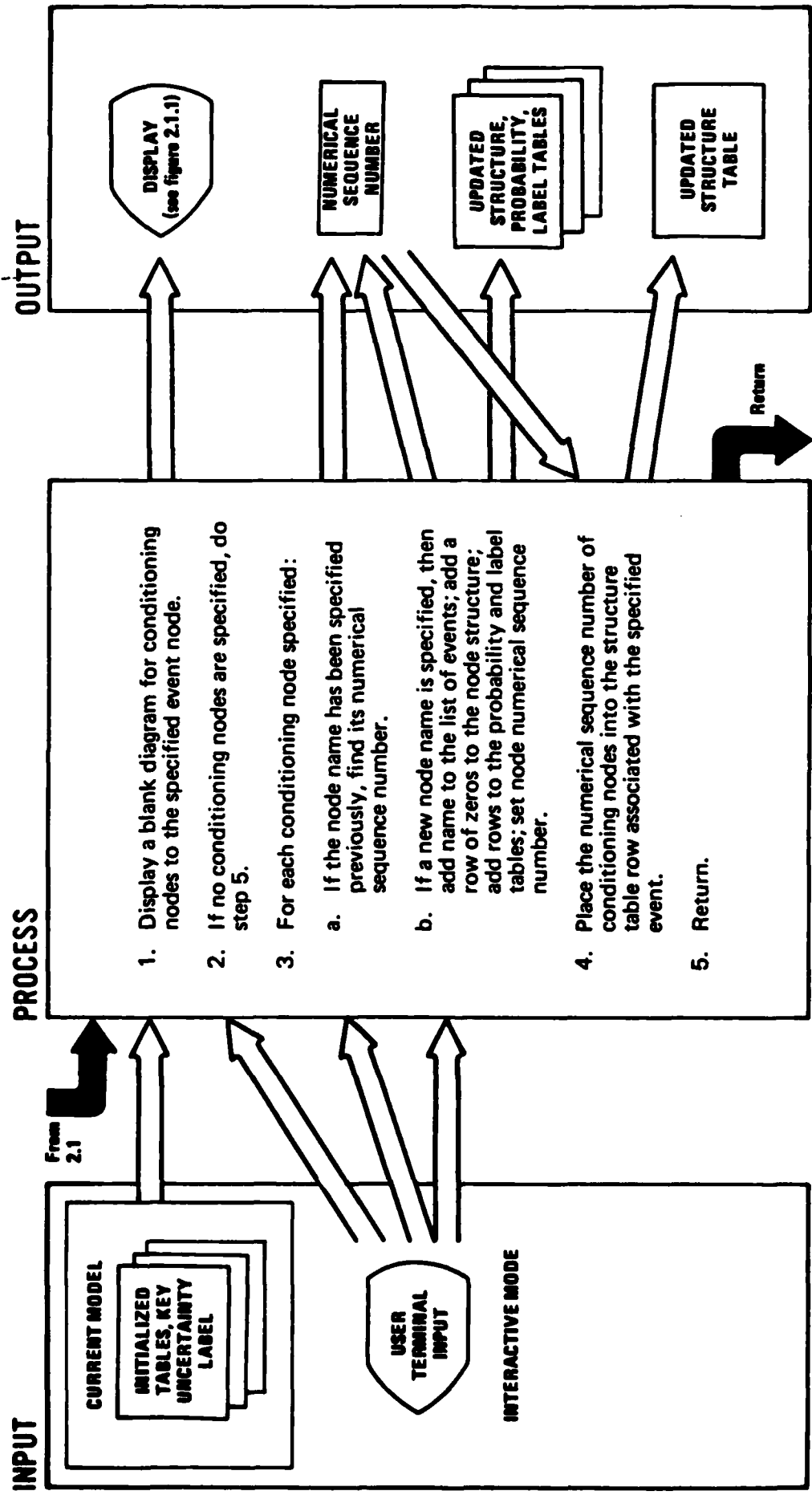


→ No. likelihoods for node 7
 → On/Off status indicating Off
 → Influenced event node number

LIKELIHOODS



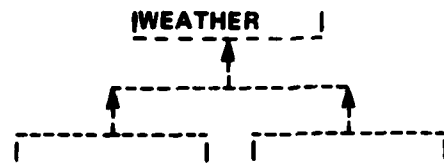
HIPO Figure 2.1
 INFER MODEL DEFINITION TABLES (continued)



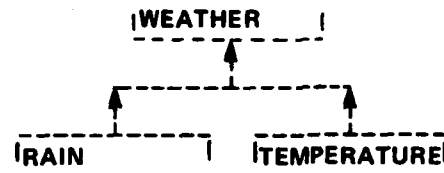
Extended Description

1. The subject diagram is really a sub-diagram of the total hierarchical structure: it is displayed only to depict the point at which influencing nodes are being added. An example of this output is shown in Figure 2.1.1.

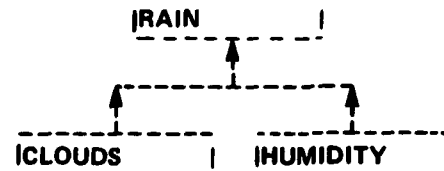
3. This step provides the linkage structure between influencing and conditioned events in the overall diagram. The key uncertainty node is the top node in the overall structure and has node numerical sequence number = 1. Subsequent nodes are numbered according to the order of their input label names.



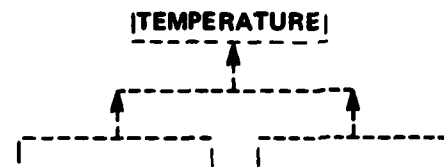
Display 1



Display 2



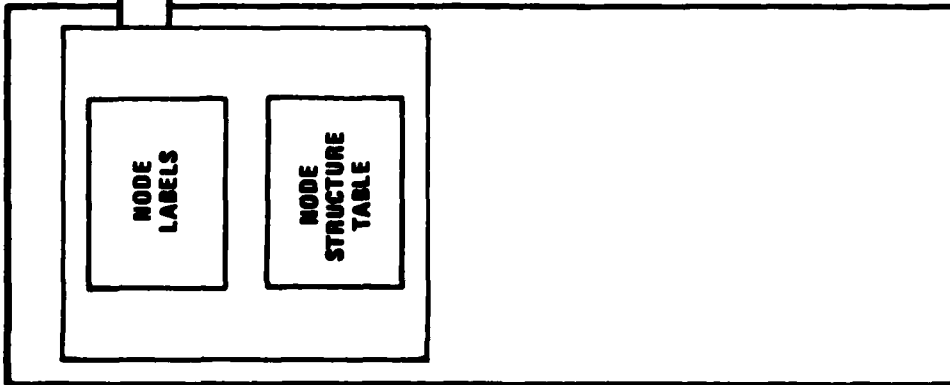
Display 3



Display 4

HIPO Figure 2.1.1
EXAMPLE OF NODE STRUCTURE ELICITATION

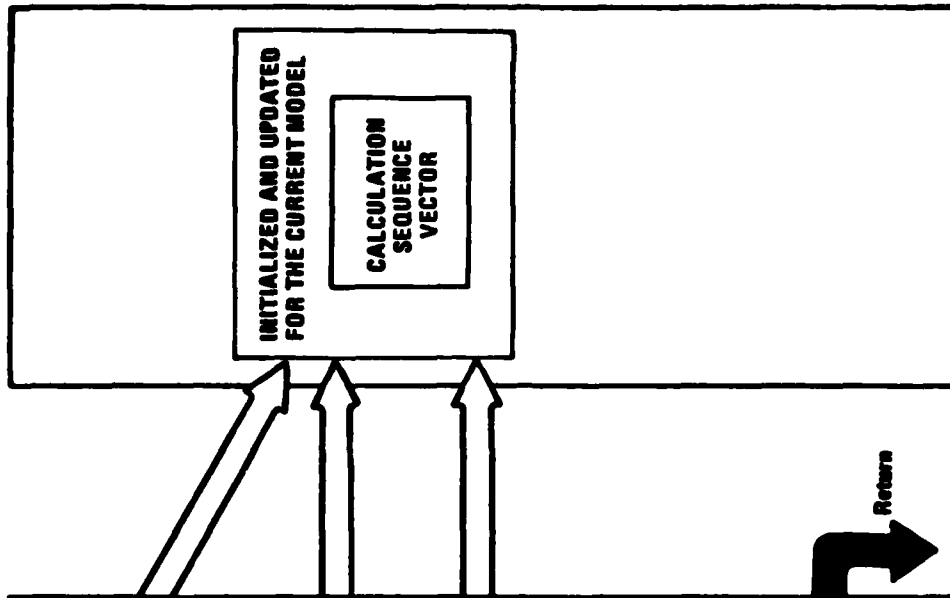
INPUT



PROCESS

1. Initialize an order sequence vector.
2. Determine the non-influenced nodes and place these node numbers in the sequence vector.
3. Add to the sequence vector node numbers for directly influenced events at the next higher level.
4. Repeat from step 3 until node No. 1 has been added.
5. Return.

OUTPUT



Extended Description

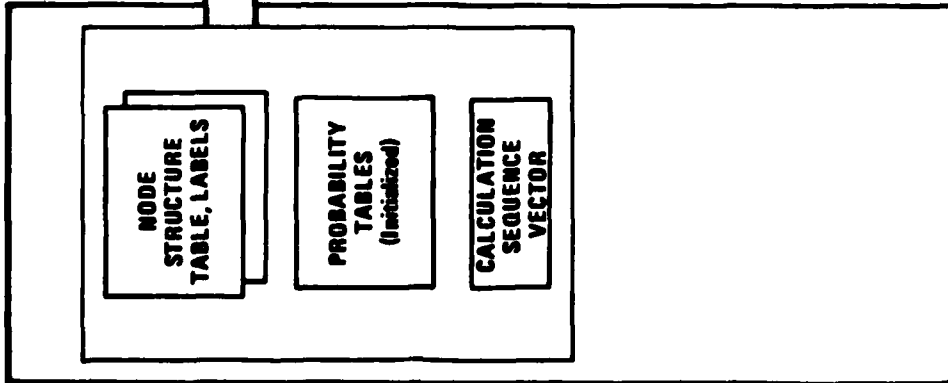
The order for calculating (or eliciting) probabilities is needed in order to compute the conditioned or marginal probabilities after model creation and after influence diagram editing.

2. The node structure table is referenced in order to determine those nodes which have zero entries: the zero entries imply that these nodes are not influenced by other nodes in the model. Node numerical sequence numbers for these nodes are placed first in the sequence vector.

3. For each node number already in the node sequence vector (from step 2 above), add the node number of the directly influenced event to the sequence vector. In order to preserve the order of influencing events, the following algorithm is used:

- a. Create a duplicate 2 column array of the first 2 columns of the node structure table.
- b. If both elements of a row in the "duplicate" are zero, place the row index value (identical to the node number) in the sequence vector.
- c. Replace the node numbers found in part b. with zeros in the "duplicate" array. Replace the index value rows with a number greater than the total number of nodes.
- d. Repeat from b. until node number 1 has been added to the sequence vector.

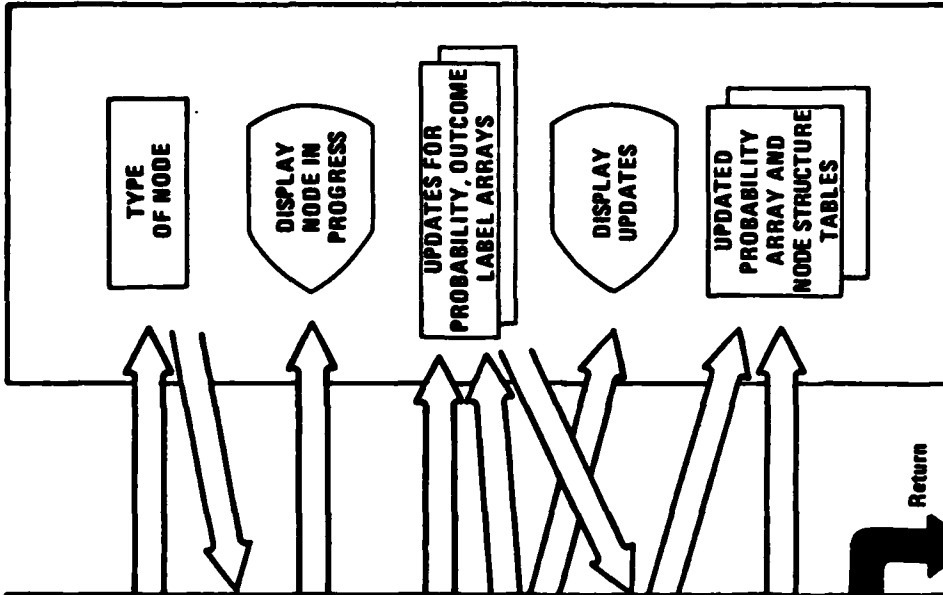
INPUT



PROCESS

- For each event node, k, in the calculating sequence vector.
1. Determine the node type.
 2. Enter probabilities according to the type of node.
 - a. Display node name.
 - b. Elicit the number of outcomes and labels for these outcomes.
 - c. Elicit probabilities.
 - d. Normalize probabilities and display.
 - e. If corrections, repeat from c.
 - f. Enter probabilities in the k-th matrix of the probability table.
 - g. Enter the number of associated probabilities and the number of outcomes in the k-th row of the node structure.
 3. Return.

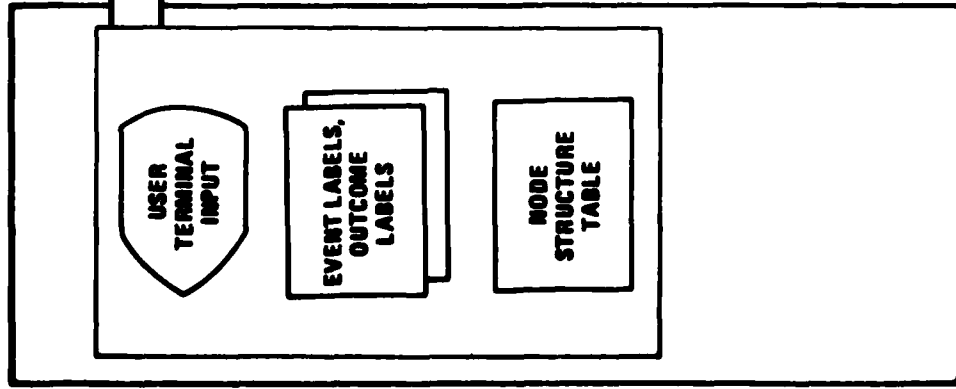
OUTPUT



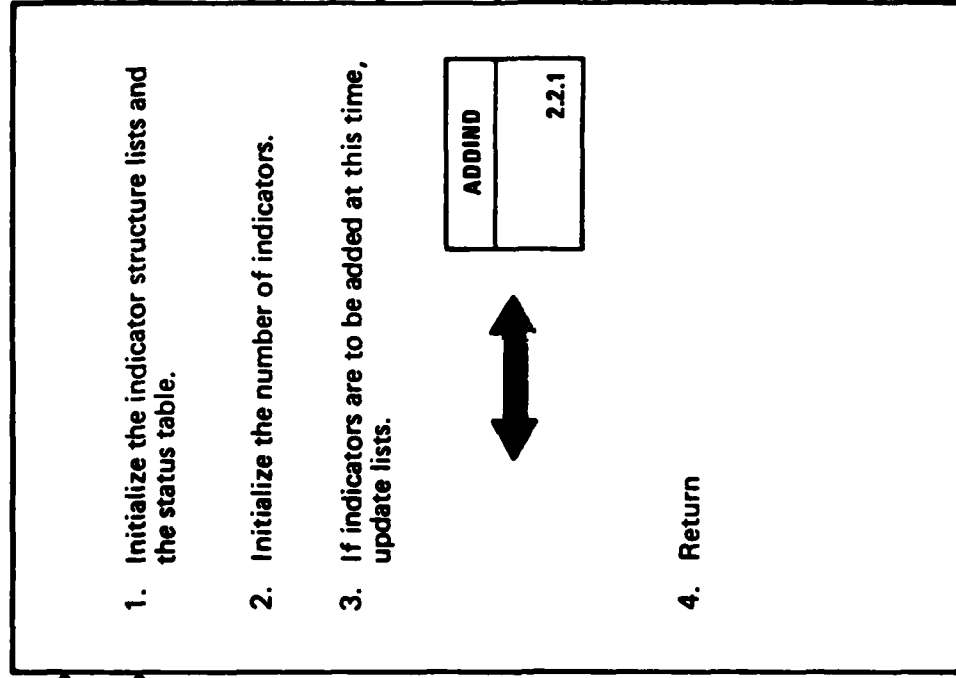
Extended Description

1. The node 'type' is characterized by the number of influencing nodes.
2. c. If the node type is zero (no influencing nodes), the probabilities of outcomes are requested for direct input into the probability array. If the node is type one, then the probabilities are extracted in dependent mode; e.g., "What is the probability of outcome k given the following outcome of event k₁ (or k₂):". A probability is needed for each outcome of influencing events. Multiple dependencies are prompted for type two or greater.
- 2.f. For each event k, a probability matrix contains a row for each possible associated outcome. For a type two event, the number of rows required is greater than or equal to the product of the number of outcomes for the two directly influencing event nodes.

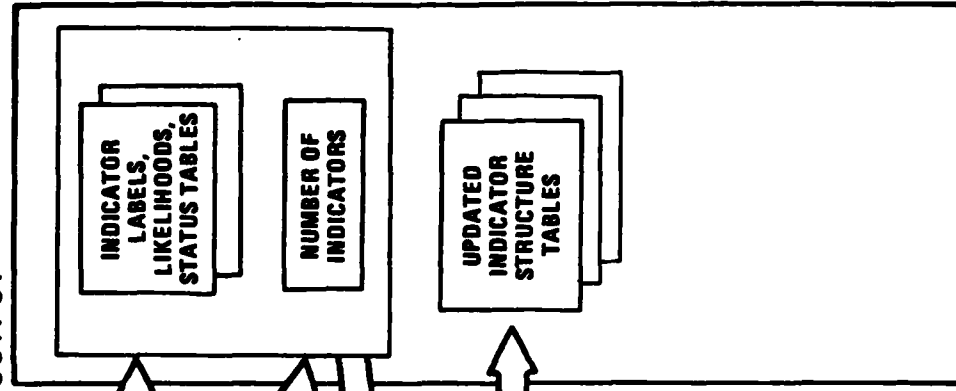
INPUT



PROCESS



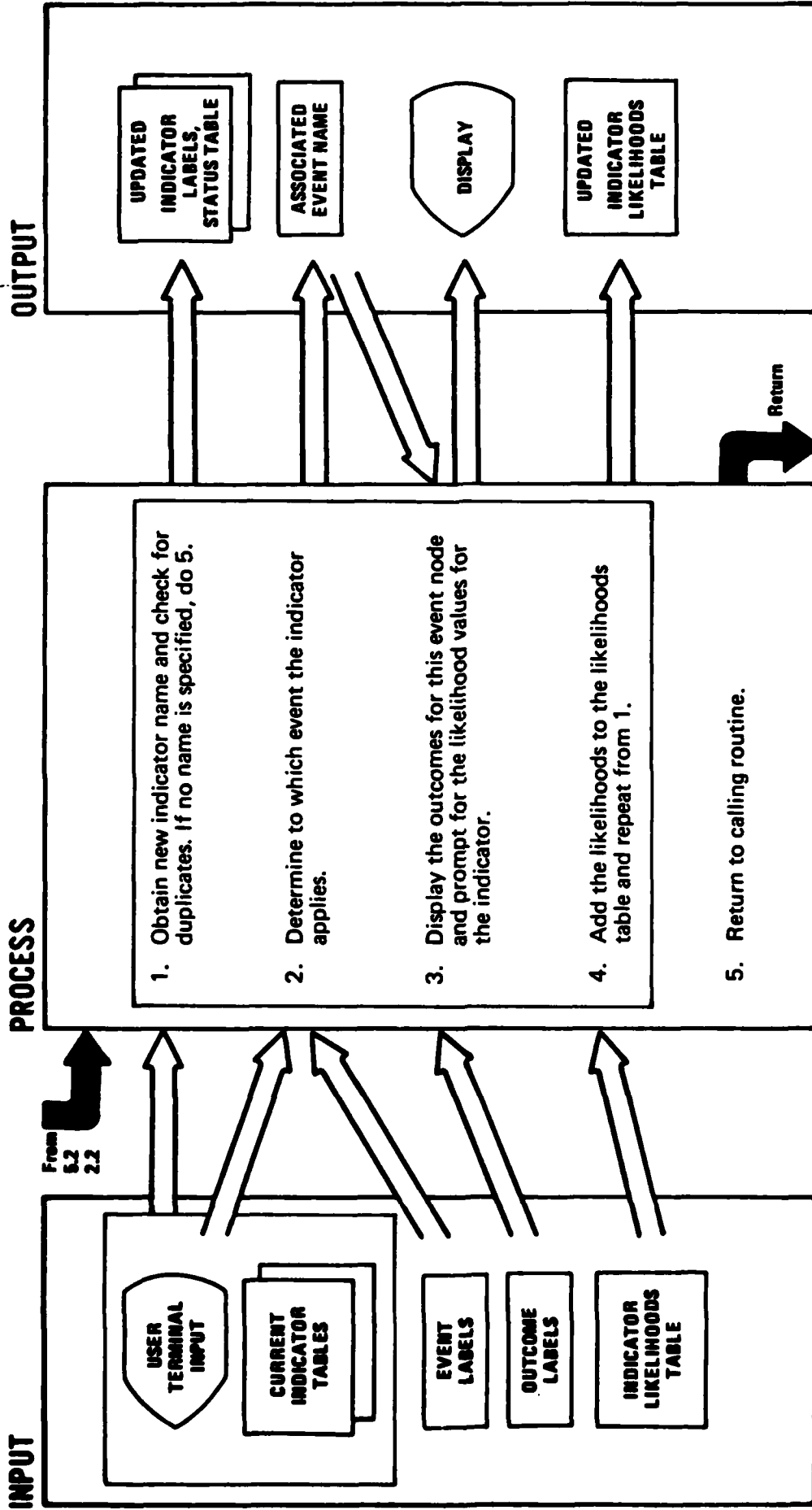
OUTPUT



Extended Description

1. For each indicator defined, the following information should be maintained:
 - the indicator label,
 - the likelihood ratios for each outcome of the event to which the indicator applies,
 - an "On/Off" status setting which shows whether or not the influence of the particular indicator is to be used in the calculation of conditioned probabilities,
 - the node numerical sequence number of the influenced event node.

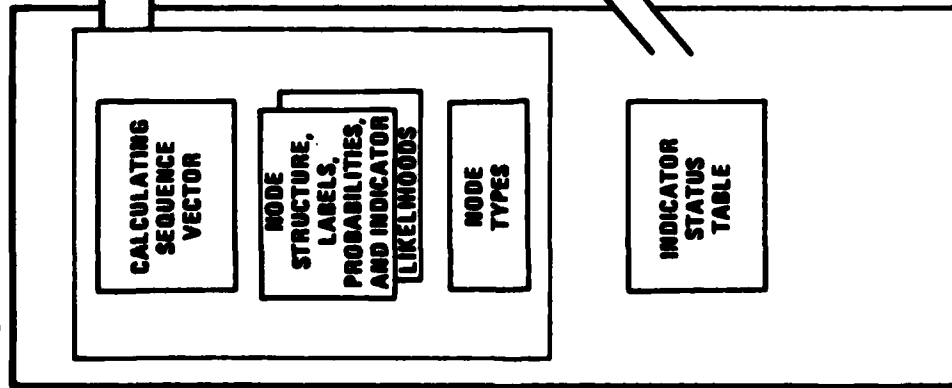
The indicator probability table contains the likelihoods for the specified indicators in the same row sides as the indicator label in the labels array.



Extended Description

- 1-2. The user is prompted for both the indicator label and the event name which the indicator should influence. The node numerical sequence number of the influenced event node is placed in the "On/Off" status table.
3. The user inputs the likelihood ratios or probability values along each event outcome.

INPUT

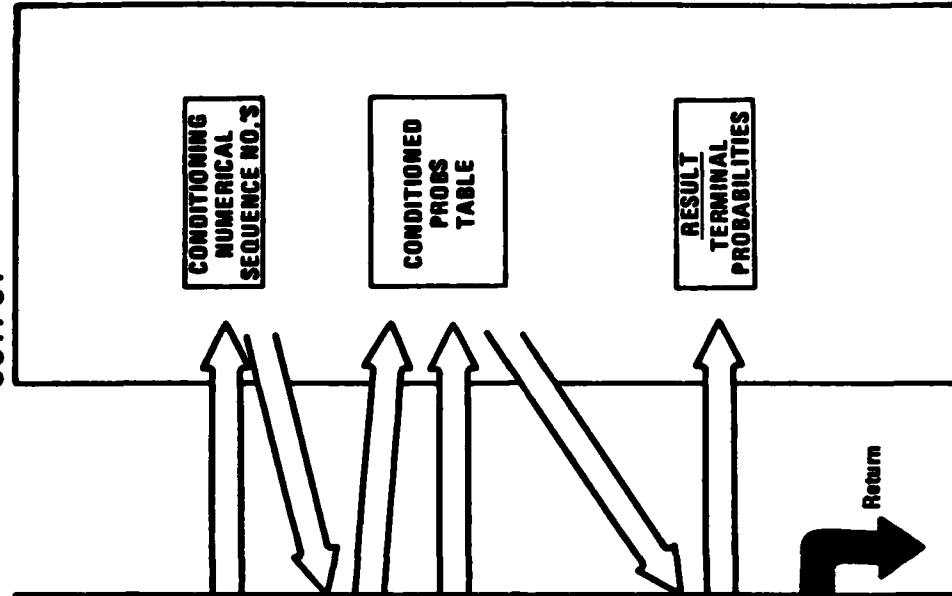


From 2.0

PROCESS

1. For each event node, k, in the calculating sequence vector.
 - a) Obtain the node numbers of influencing nodes.
 - b) Calculate the conditioned probabilities according to type.
 - c) Multiply probabilities by Bayesian update method for any associated indicators that have "On" status.
 - d) Normalize resulting vector.
2. Place final set or the terminal node probabilities in the result vector.
3. Return.

OUTPUT



Extended Description

1. Step 1 is processed in the order of node number occurrence in the calculating sequence vector. The computations required in steps 1.b) - 1.d) are described below for each type of node.

node k is Type Zero:

Obtain the directly specified outcome probabilities for the node from the probability table. Store this vector in the k-th row of the Conditioned Probs table; then perform steps A and B.

node k is Type One:

The conditioned probabilities of the one directly influencing node have already been computed (and stored in Conditioned Probs table). Multiply these probabilities by the conditional probabilities for each outcome of the event node k. Then sum over the conditioned probabilities per outcome of event node k. The result contains a value for each outcome of event node k. Store the resulting vector in the Conditioned Probs table; then perform steps A and B.

System/Program: INFER

Name: _____

Description: Compute the Marginal Probabilities

Diagram ID: 2.3

Page: 2 of 2

INPUT

PROCESS

OUTPUT

Extended Description (continued)

node k is Type Two:

The conditioned probabilities of the two influencing node have already been computed and stored the Conditioned Probs table. Create a vector of joint probabilities by multiplying each conditioned probability value of one influencing event times each single conditioned probability value of the other influencing event.

Multiply (via matrix algebra) the resulting vector times the conditional probabilities of each outcome of event node k and then sum over conditional probs for each outcome. The result contains a value for each outcome of event node k. Store this result in Conditioned Probs table. Then do steps A and B.

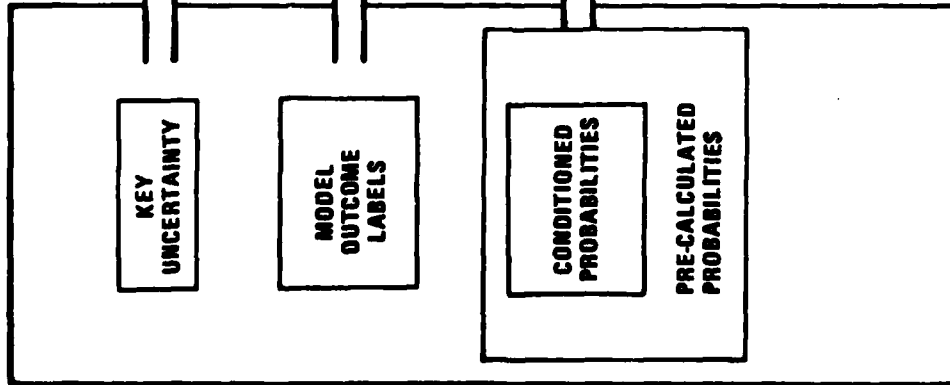
Step A.

Multiply the conditioned probabilities for node k by each set of indicator likelihoods which apply to the node and have "On" status. Store the result in the k-th row of the Conditioned Probs table.

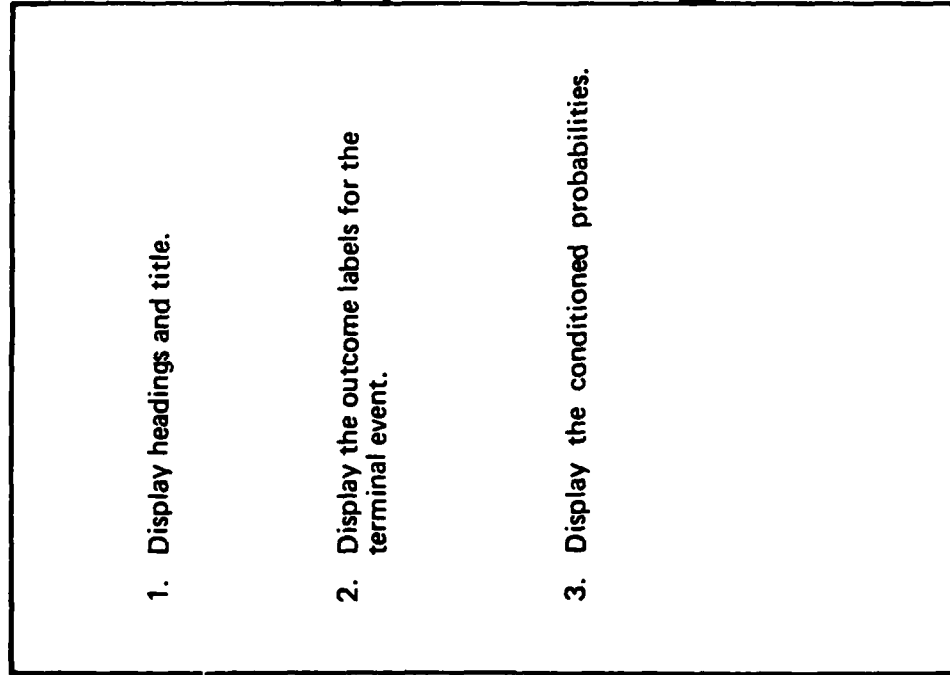
Step B.

Normalize the conditioned probs for node k so that the sum of all outcome probabilities is 1 (or 100%). Store the results back into the Conditioned Probs table.

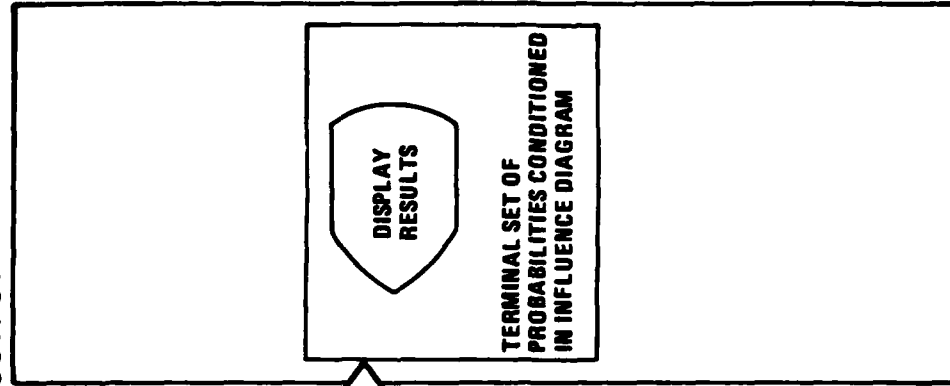
INPUT



PROCESS

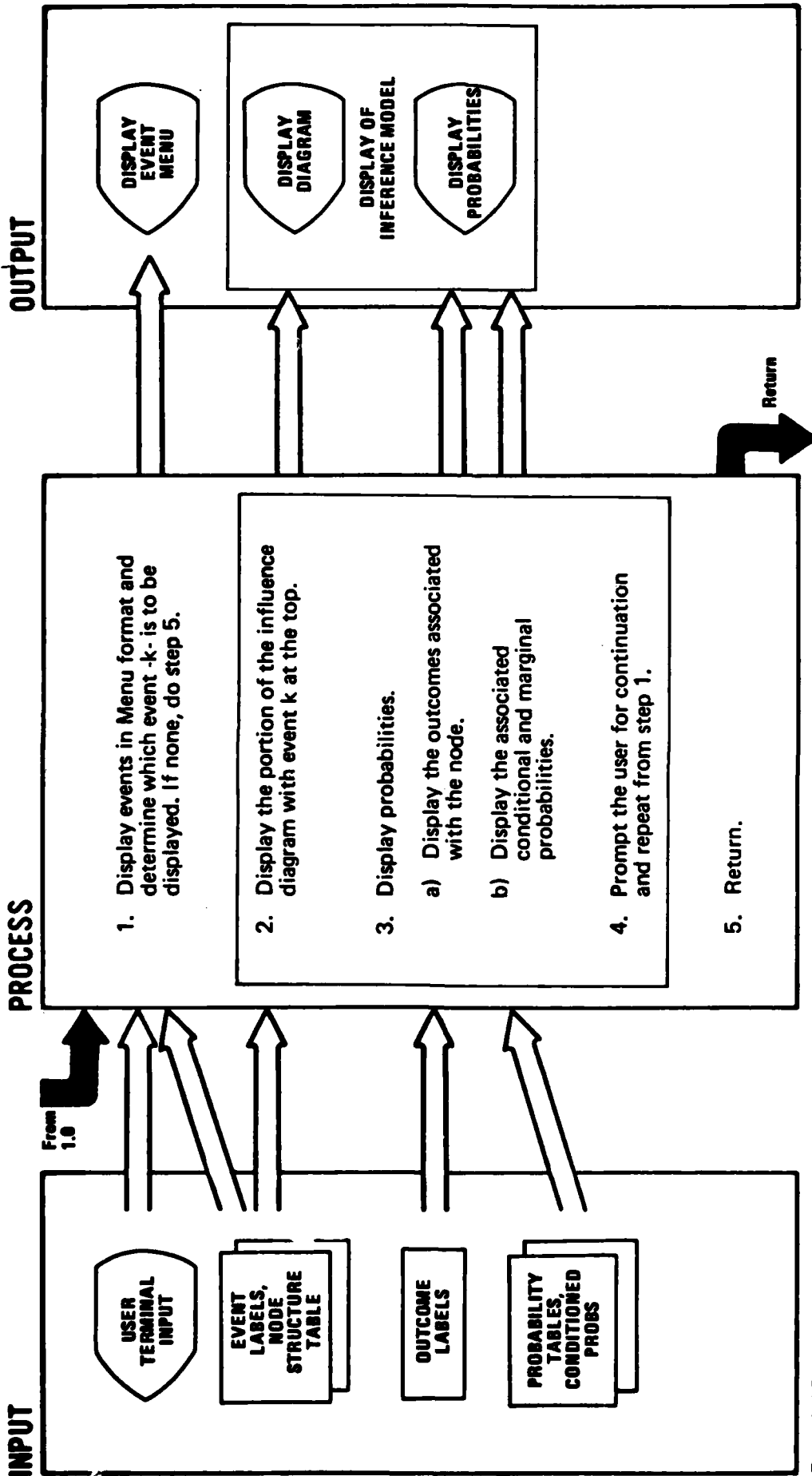


OUTPUT



Extended Description

1. The title contains the model name. The display option label "Display Results" is used as a heading.
2. The outcome labels of the key uncertain event are displayed horizontally.
3. The calculated probabilities of the key uncertain event outcomes are displayed beneath the respective outcome labels.

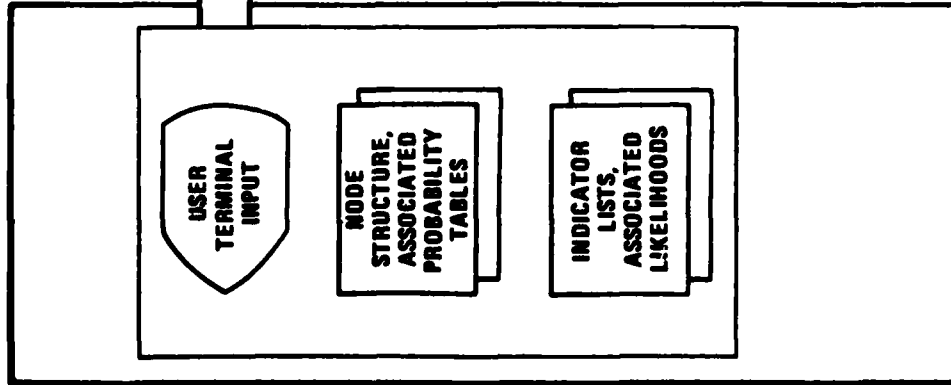


Extended Description

1. A Menu list of the event labels in the model are displayed and the user is prompted to select the node to be displayed. If no event name is specified, then control is returned to the calling routine.
2. The label for event k is depicted in hierarchical manner above labels of event nodes which influence it (see figure 2.1.1).
3. The outcomes labels of event k are displayed horizontally and the outcome labels of directly influencing events are listed vertically; the associated conditional and joint probabilities are displayed beside the appropriate outcome levels.

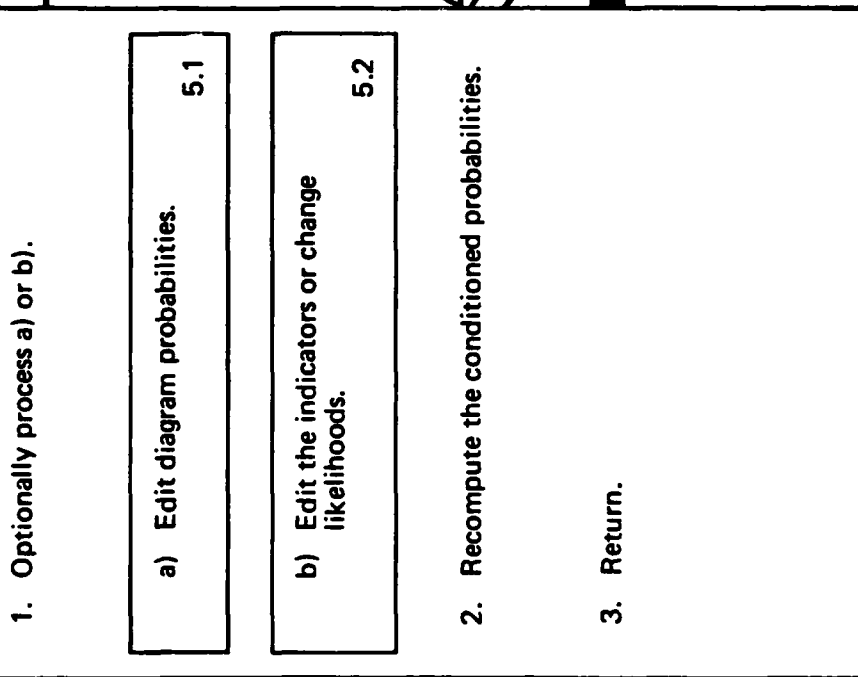
The marginals, i.e., conditioned, probabilities are displayed at the end.

INPUT

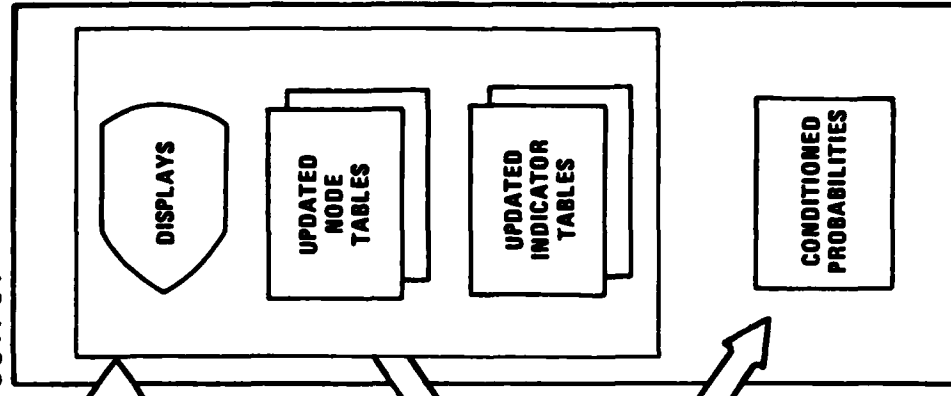


From 1.0

PROCESS



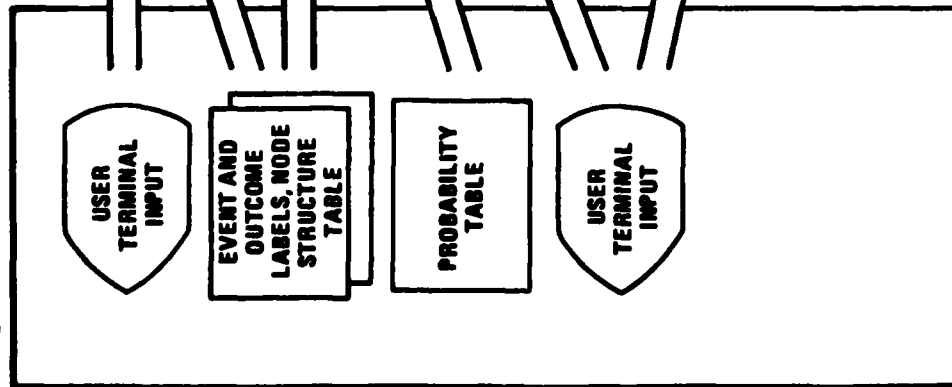
OUTPUT



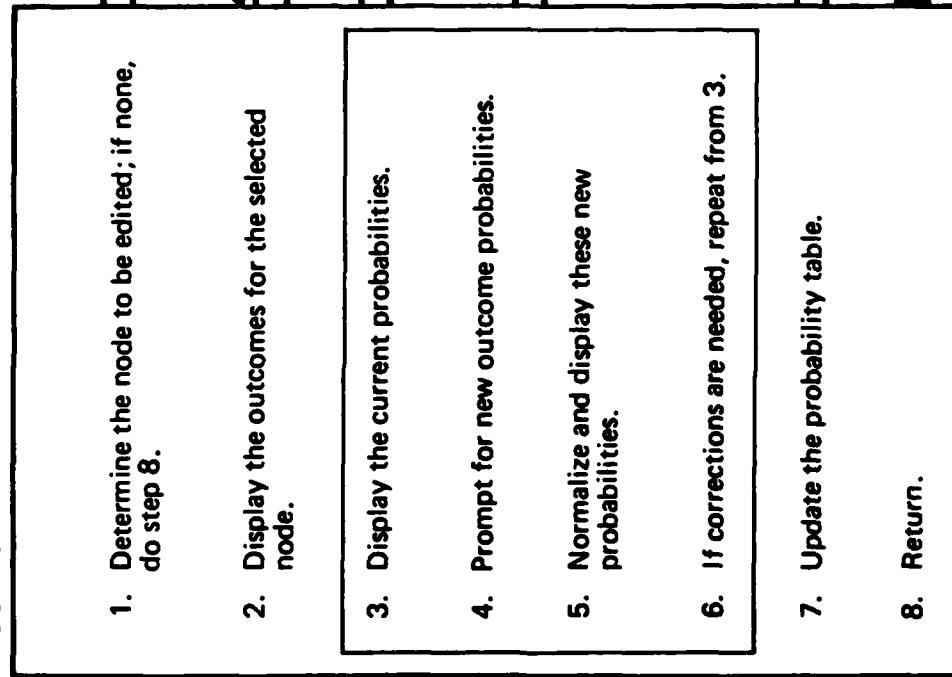
Extended Description

1. The user is given the option of modifying either the influence diagram probabilities or the set of indicators and the associated likelihood ratios. The functions involved for these processes are repetitive procedures.
2. Changes made to the probabilities or the number of indicators require the recomputation of marginal probabilities.

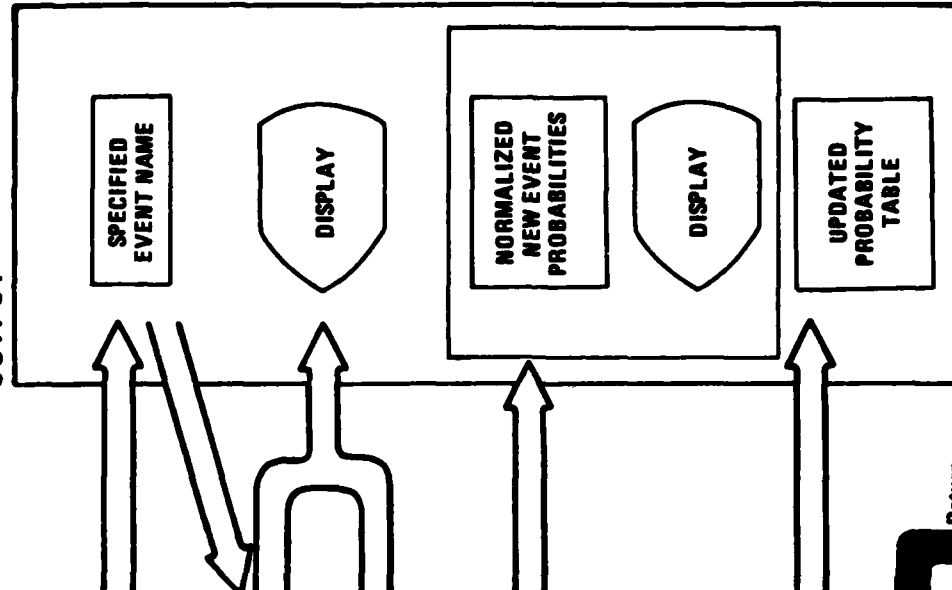
INPUT



PROCESS



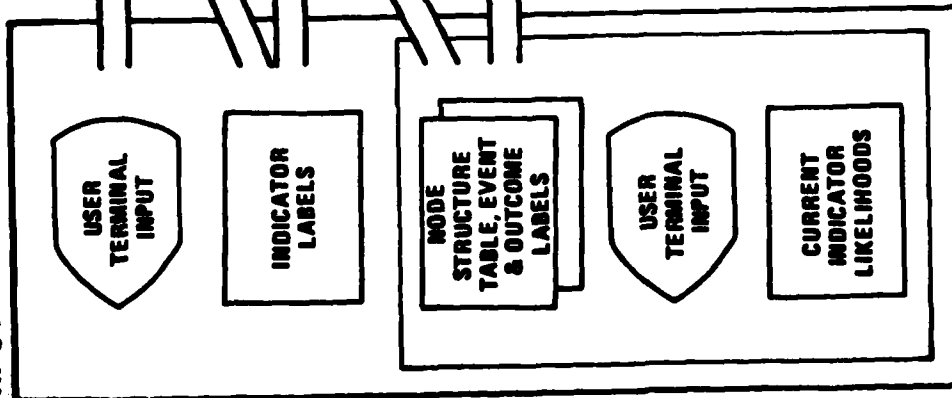
OUTPUT



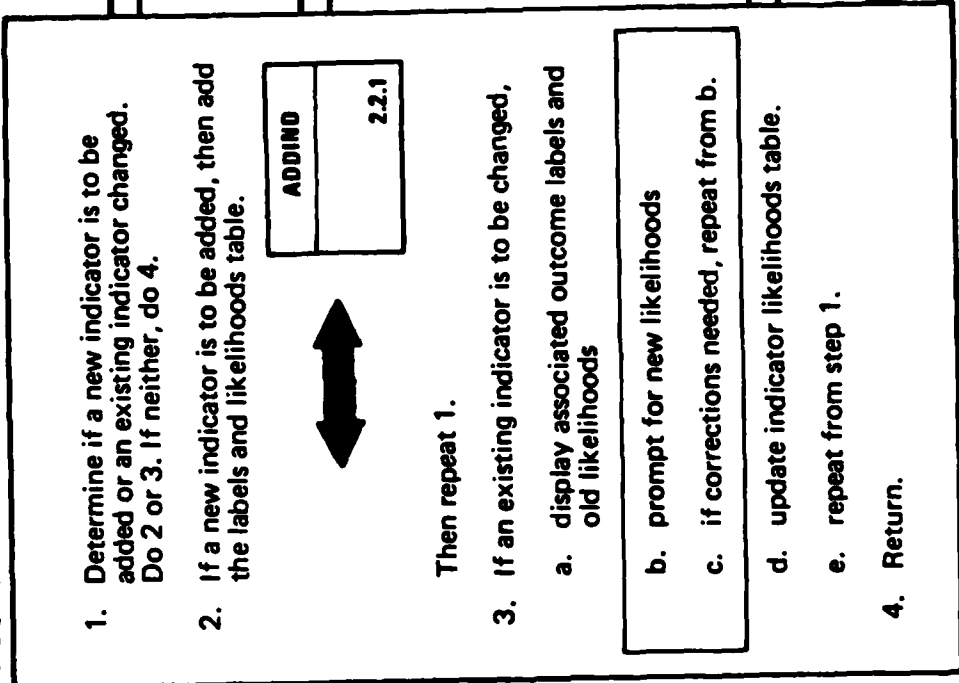
Extended Description

1. Prompt the user for the name of the event node that is to be edited. If the user inputs a blank field, return to the calling routine.
3. The associated outcome probabilities for the specified node are displayed. These values will be the directly assessed values for nonconditioned nodes and the user-specified conditional probabilities for the conditioned event nodes.
The joint and marginal probabilities from the most recent calculations also displayed for conditioned nodes of type 2.
4. Any set of probabilities that were originally provided as input by the user can now be changed.

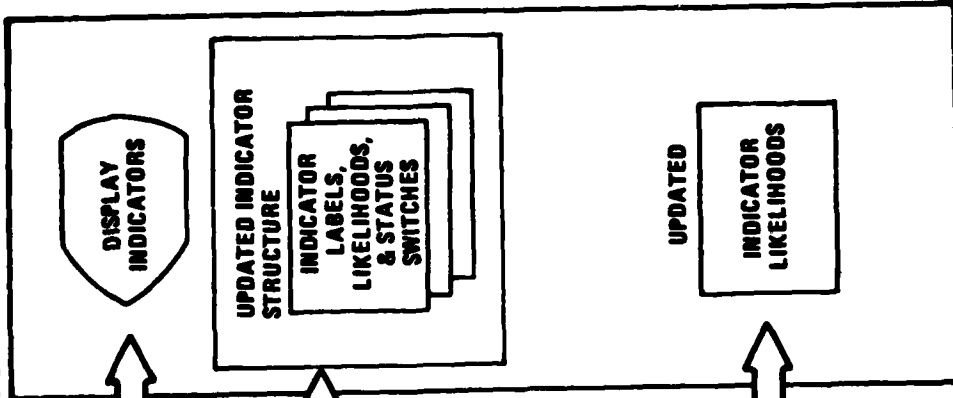
INPUT



PROCESS



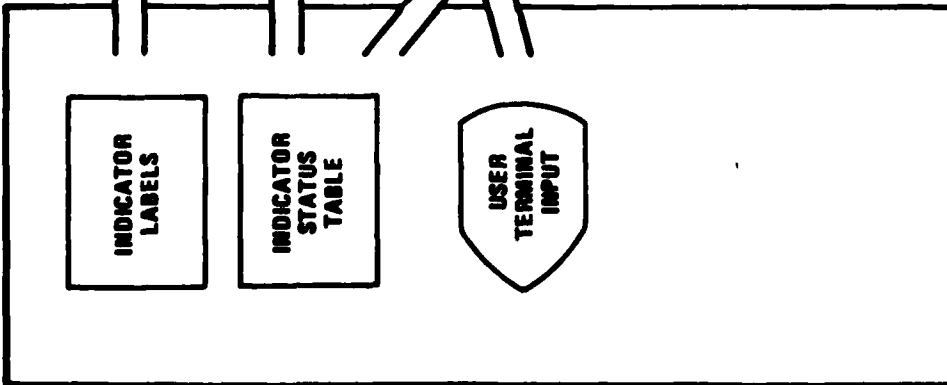
OUTPUT



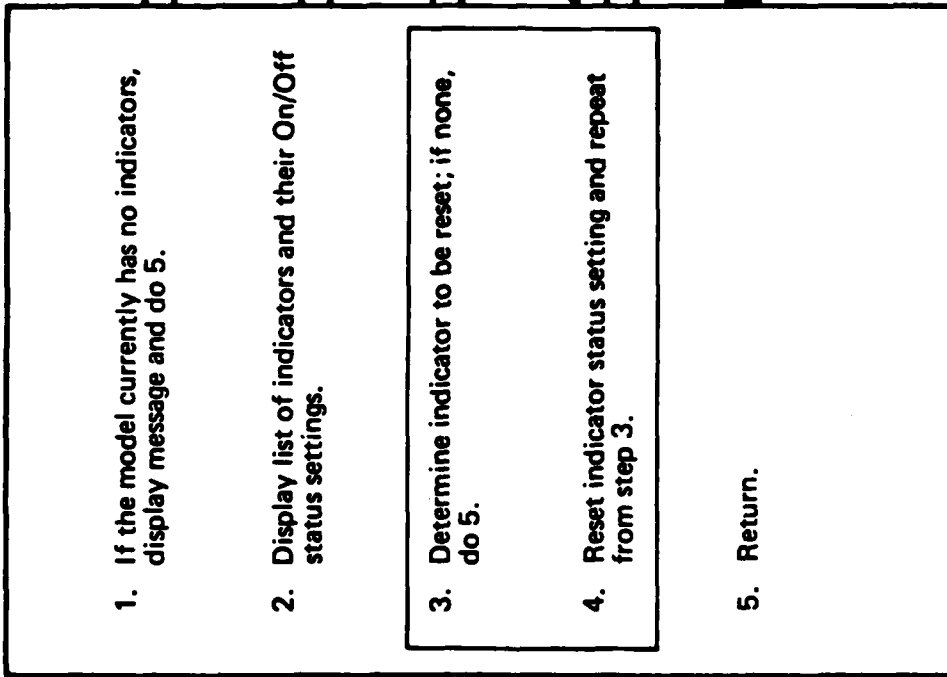
Extended Description

3. The current model structure, whether previously modified or not, is input to this step.

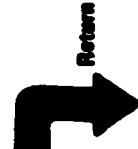
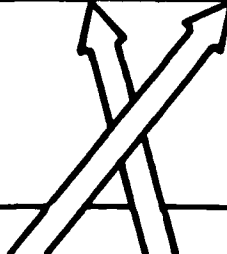
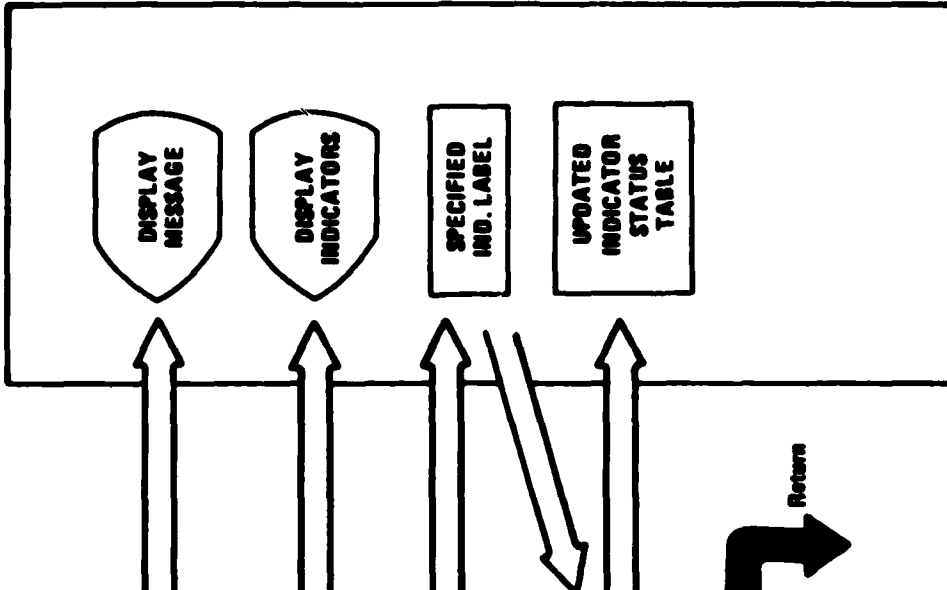
INPUT



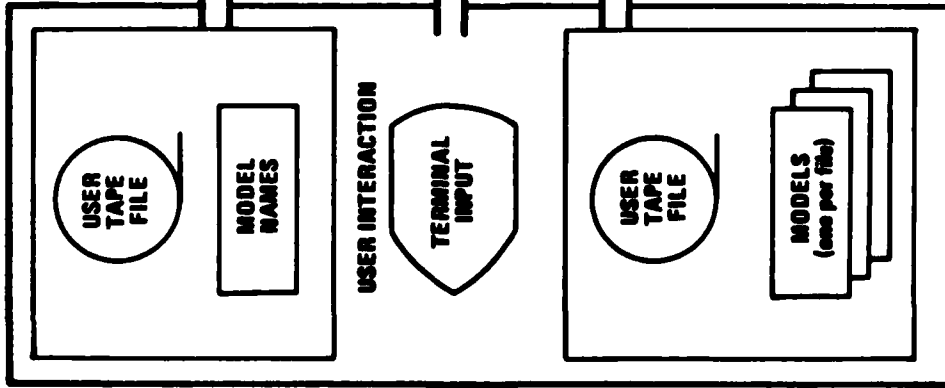
PROCESS



OUTPUT

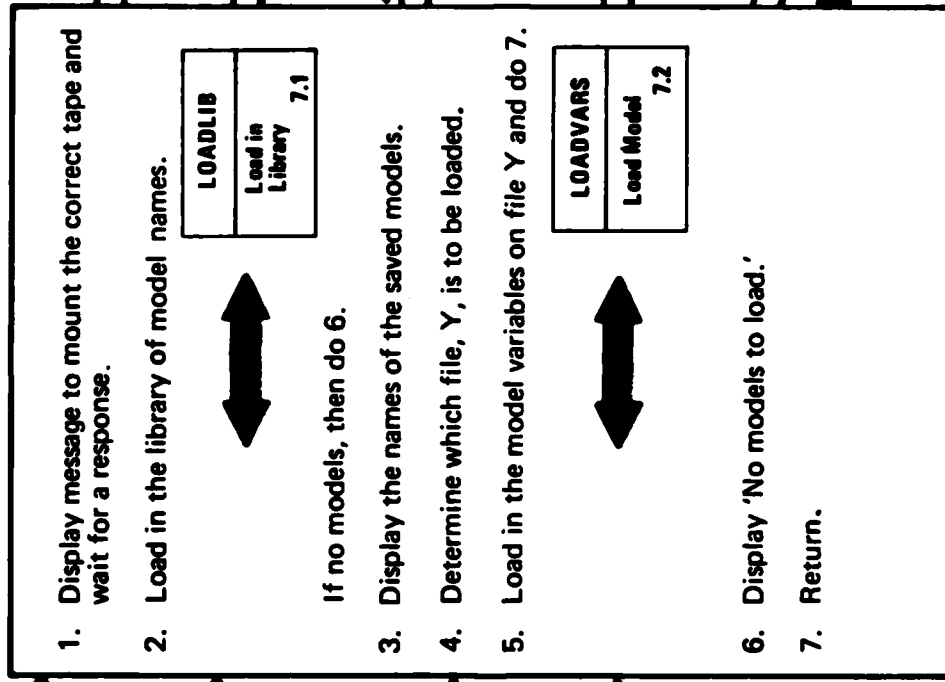


INPUT

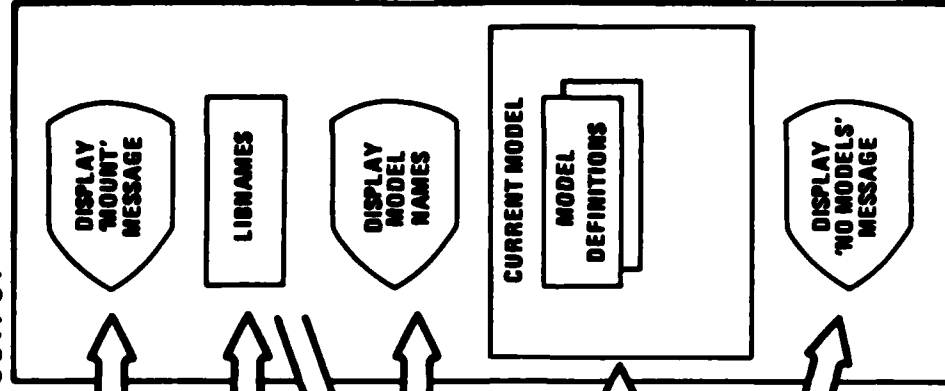


Extended Description

PROCESS



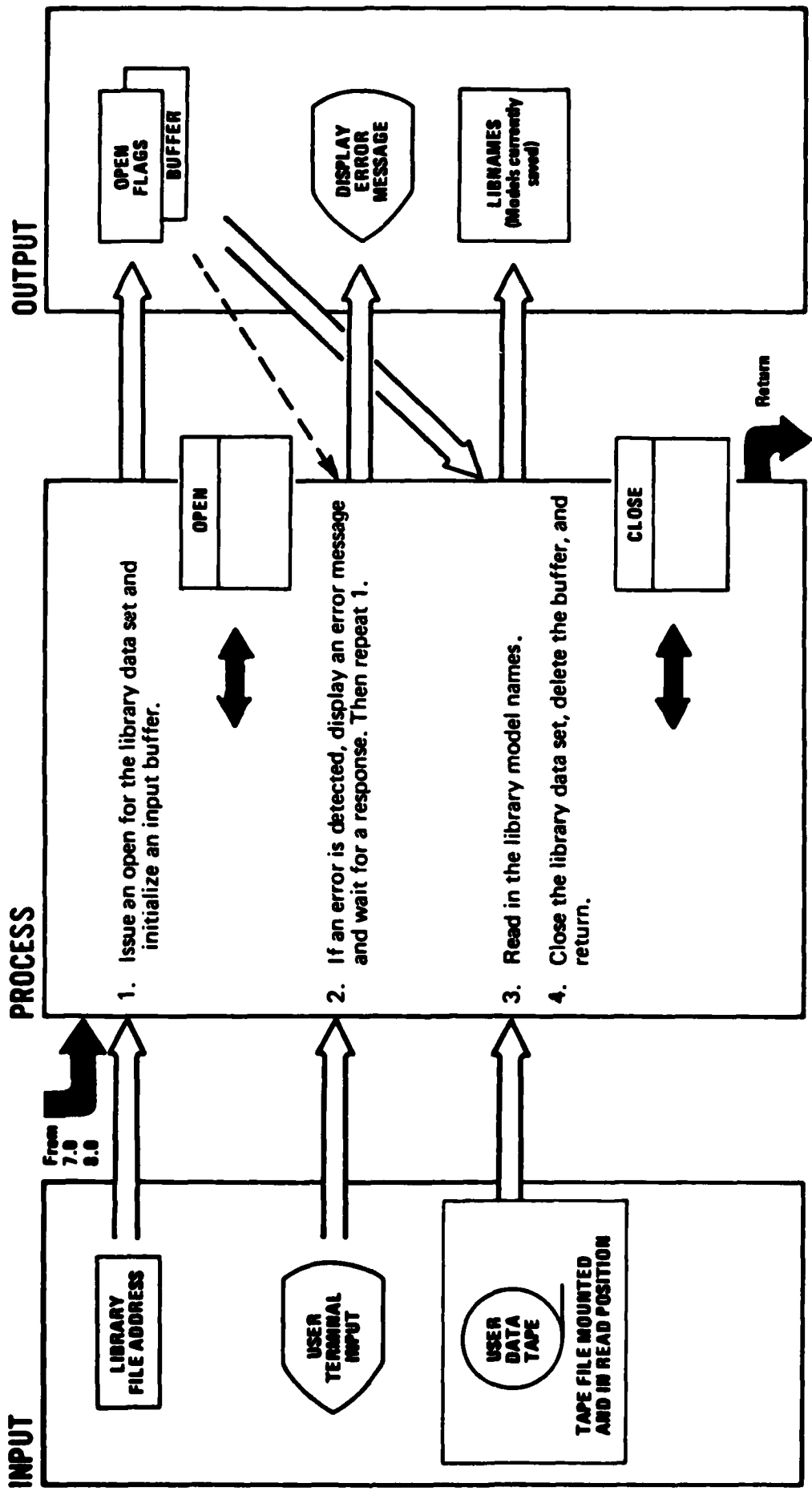
OUTPUT



1. The user may have many tape files on which INFER formatted models are stored. In this step, the user is prompted for a response indicating the desired tape is mounted and online.

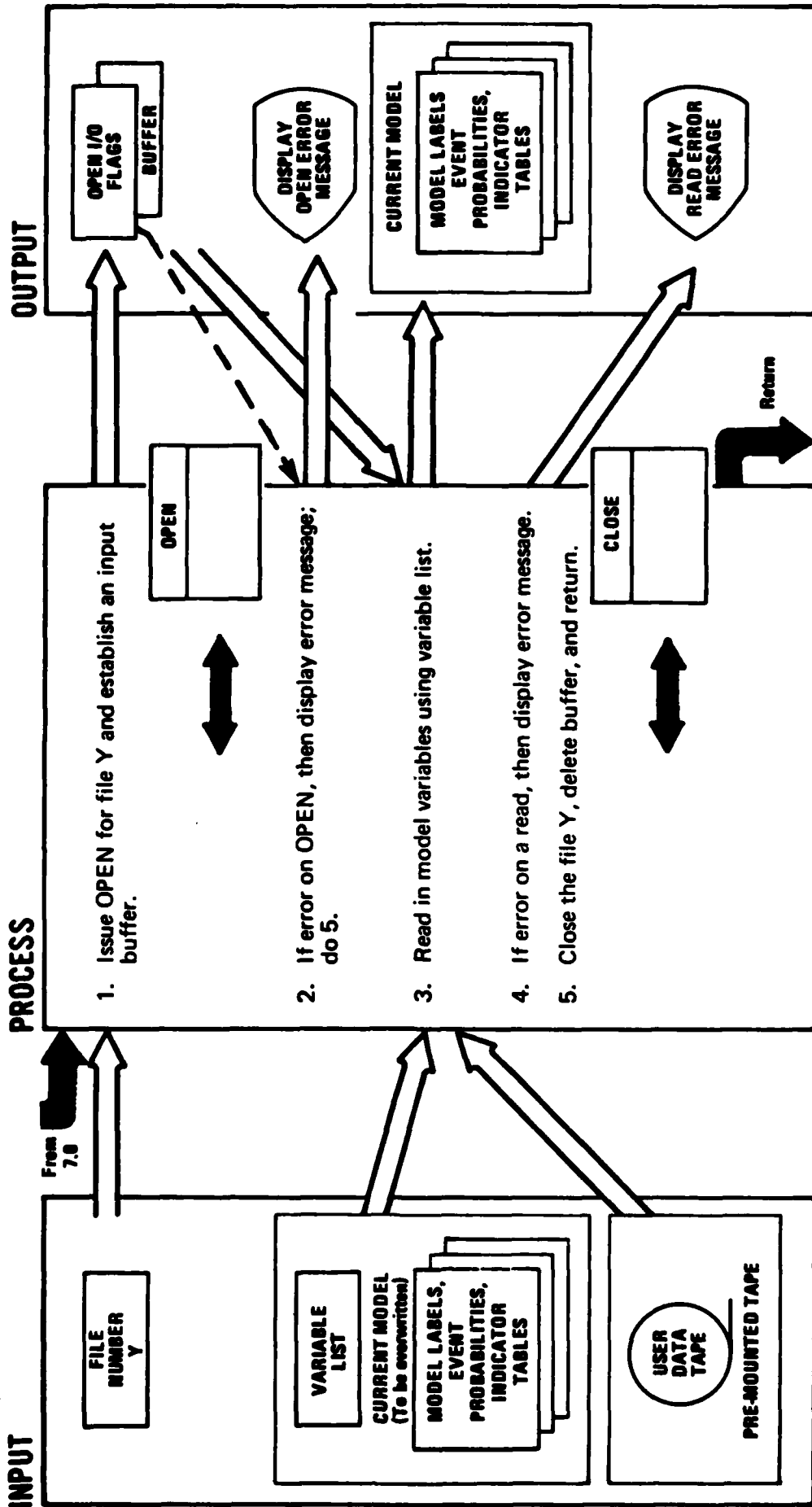
3. The names of the models existing on the mounted tape are displayed in list or Menu format so that the user may select a model for loading.

4. The user is prompted for a model selection: the response may be the list item number or the model name. The requested model is stored in the same tape file as its position relative to the other model names in the displayed list.



Extended Description

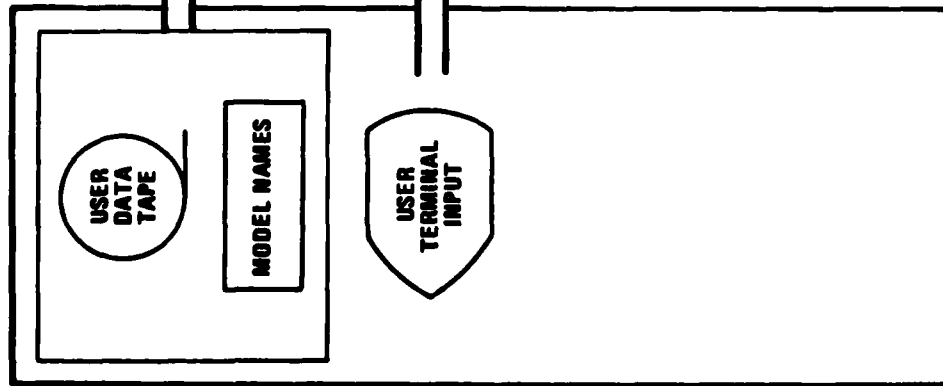
- The library file of model names is available on each INFER formatted data tape. The file is usually stored and retrieved as a character array and resides on the same device with model data. A system OPEN command is needed to ensure that the data file is online and accessible for reading. An input buffer is needed and provides the link between stored information and program addressable information.
- The library model names are retrieved from storage. The character array used for holding these model names, LIBNAMES, is of a form which facilitates display; thus, the names may all be of equal character lengths.
- A system CLOSE command is issued to free the data file for later use.



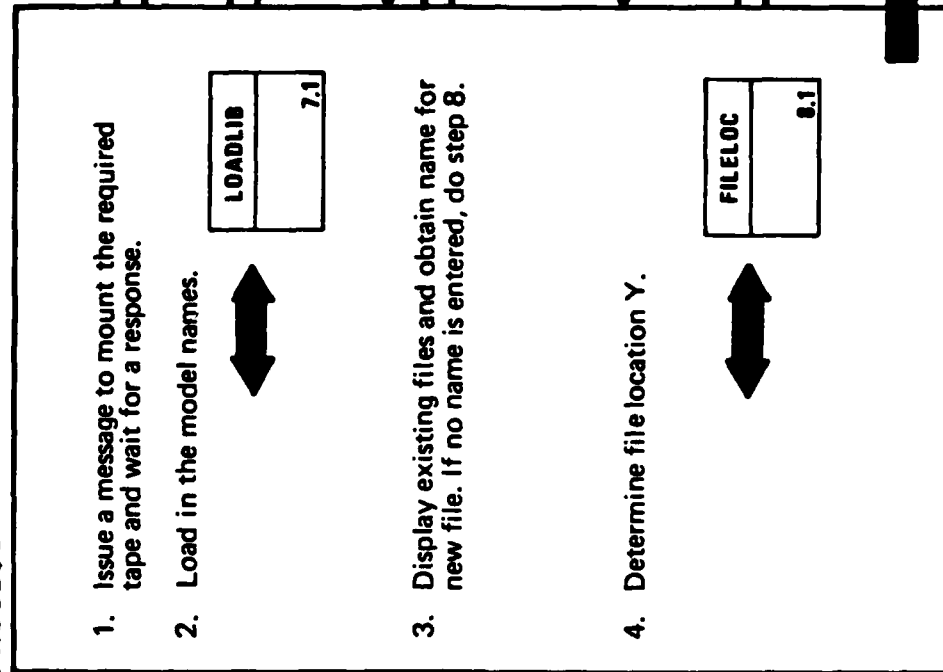
Extended Description

- 3. A list of variable Names or identifiers is kept so that load and store routines will always process the variables in the same sequence order.
- 4. The Model variables retrieved from storage are used in all other program functions (see Diagram 1.0). The variables which must be loaded are the following:
 - EVENT AND ASSOCIATED OUTCOME LABELS
 - NODE STRUCTURE TABLE
 - OUTCOME PROBABILITY ARRAYS
 - INDICATOR LABEL STATUS, AND LIKELIHOOD ARRAYS
 - REDUCED PROBABILITY RESULTS

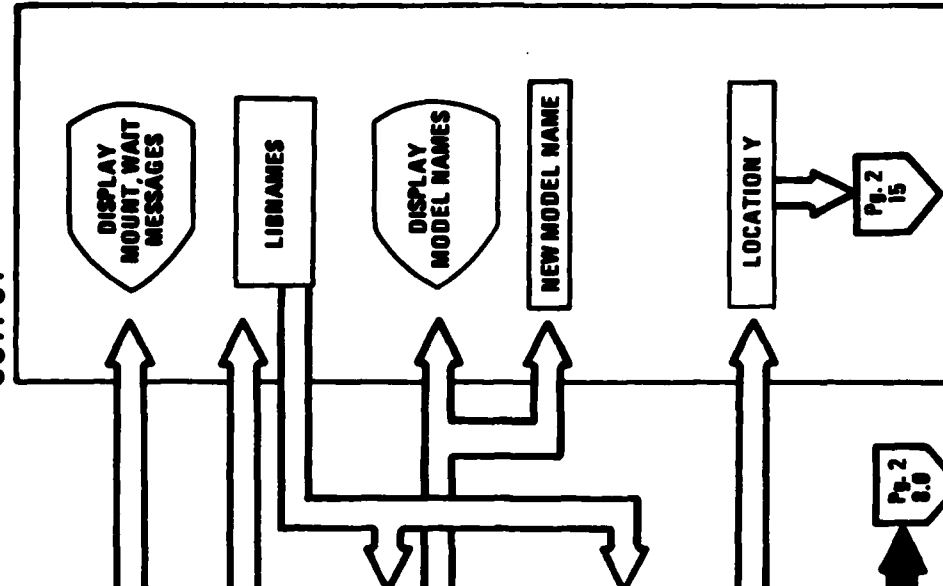
INPUT



PROCESS



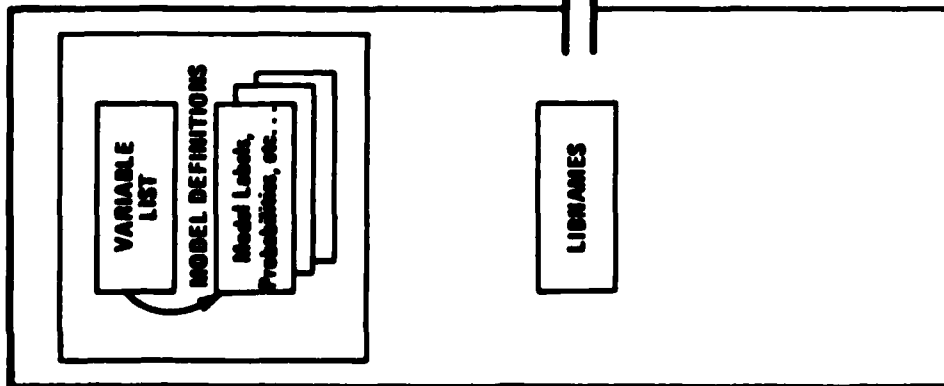
OUTPUT



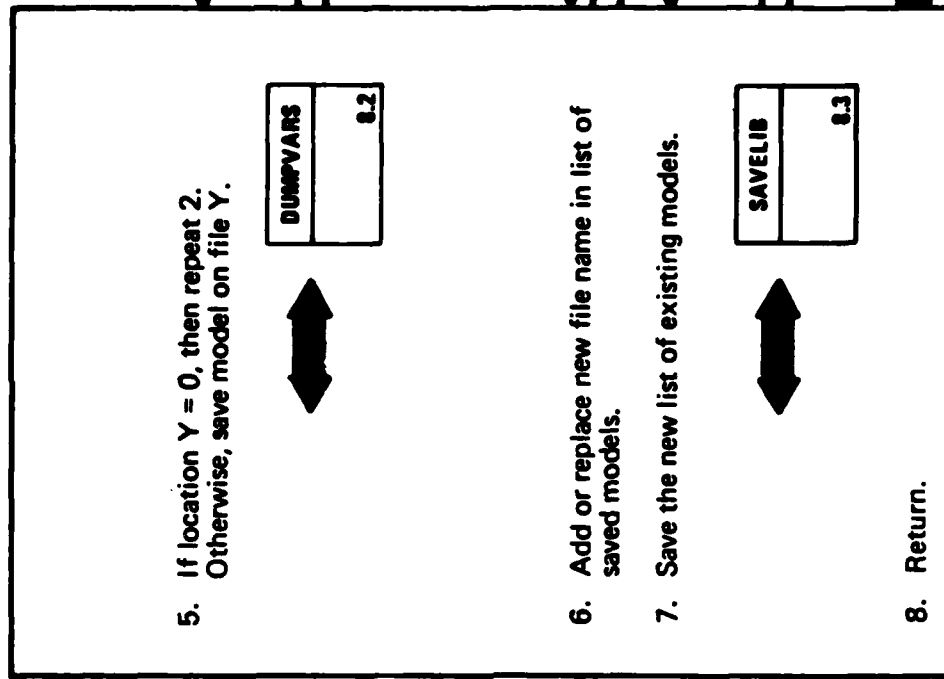
Extended Description

1. The computer program prompts for an indication that the desired storage file/device has been selected and placed online. Any response from the keyboard causes processing to resume.
4. The existing file structure and the amount of available space on the data tape are checked along with the user specification to determine where the model variables are to be stored.

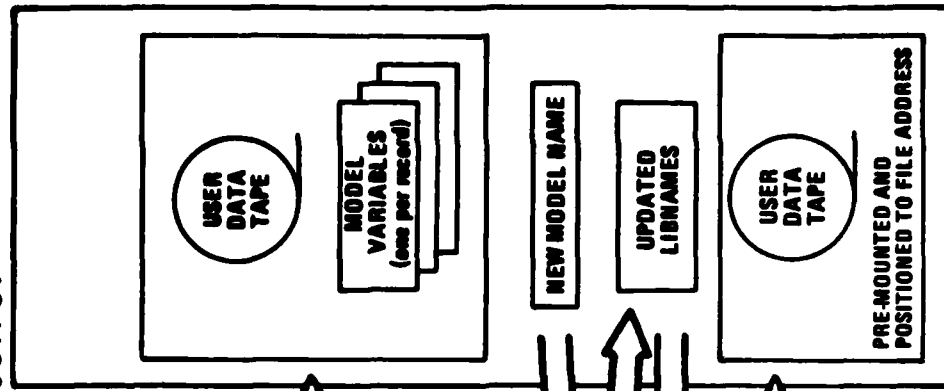
INPUT



PROCESS

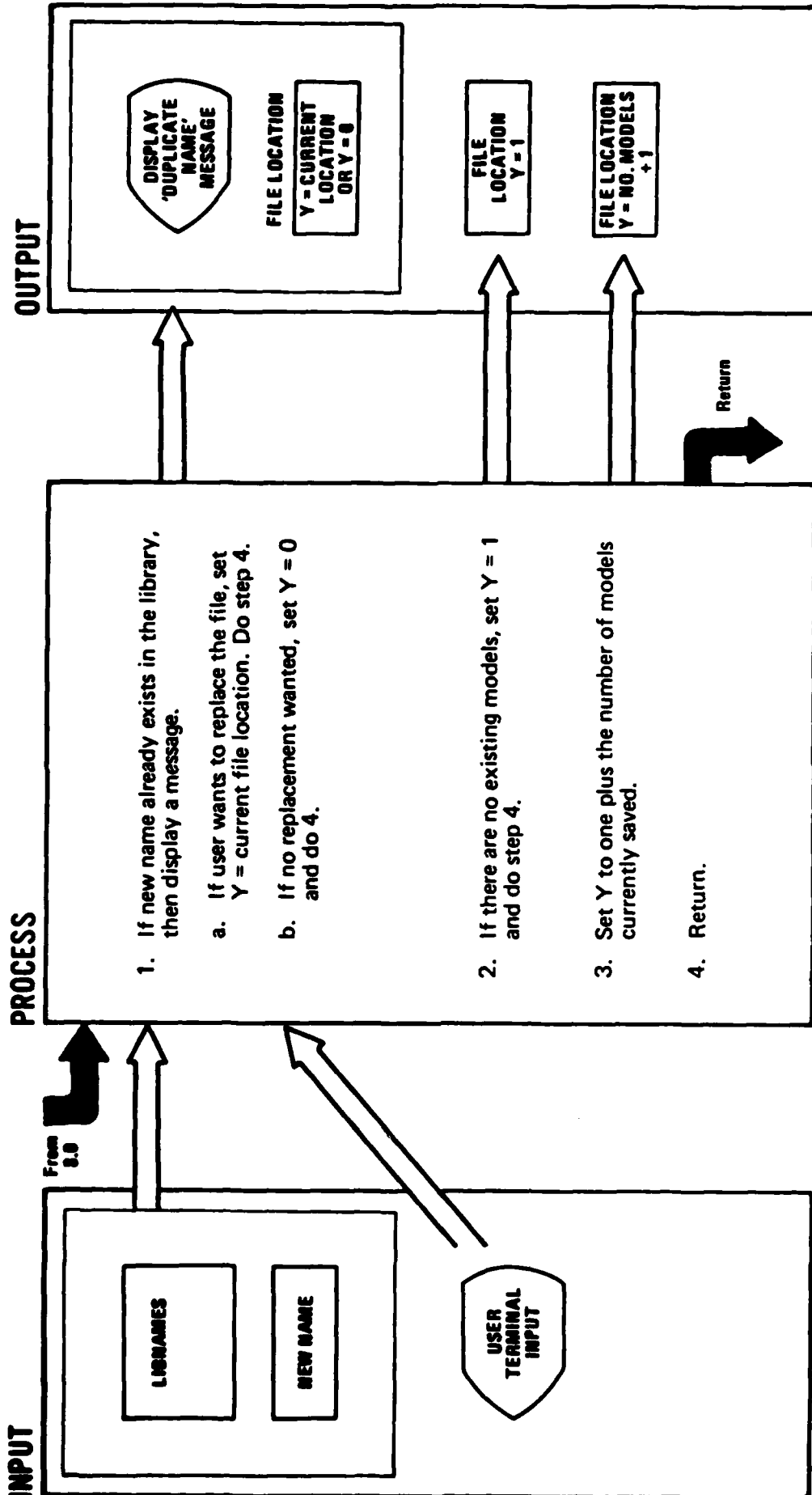


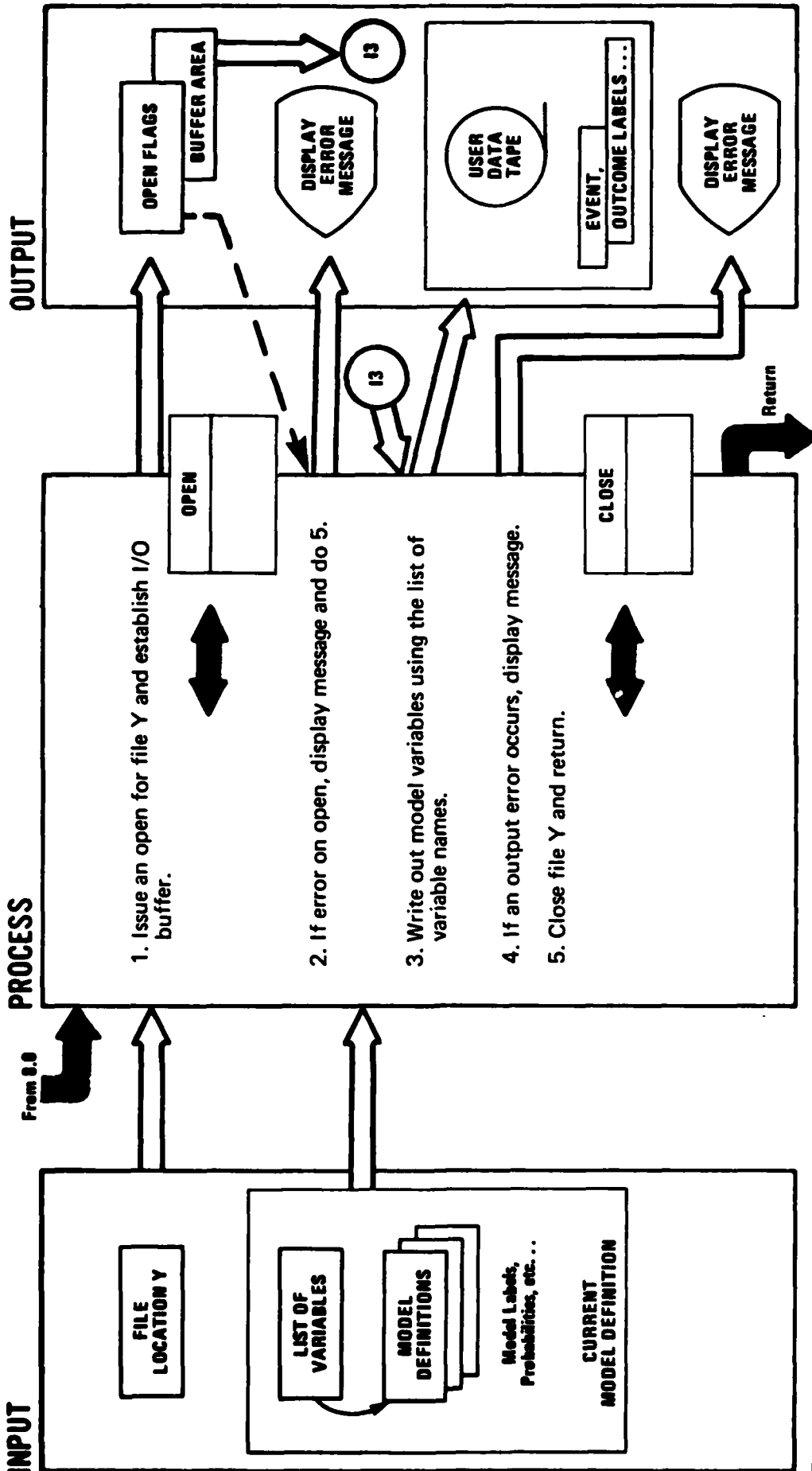
OUTPUT



Extended Description

6. The library name list is updated to reflect the new file structure. The new model name's position in the LIBNAMES array must be the same relative position to other models stored on the device.





Extended Description

1. The file location Y is used to determine an exact storage position on the selected device.
3. The list of variable names is identical to the list of names used to Load a Model. (see Diagram 7.2)

