

ESD TDR 64-234  
Vol IV  
ESTI FILE COPY

ESD-TDR-64-234  
Volume IV

ESD RECORD COPY

RETURN TO  
SCIENTIFIC & TECHNICAL INFORMATION DIVISION  
(ESTI), BUILDING 1211

COPY NR. \_\_\_\_\_ OF \_\_\_\_\_ COPIES

DESIGN AND USE OF INFORMATION SYSTEMS  
FOR  
AUTOMATED ON-THE-JOB TRAINING  
  
VOLUME IV

Graphical Symbology and Logic Diagrams for Use as Training Aids

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-234

JANUARY 1965

Benjamin C. Duggar  
Ronald C. Rosenberg  
Thomas B. Sheridan  
Sylvia R. Mayer

DECISION SCIENCES LABORATORY  
ELECTRONIC SYSTEMS DIVISION  
AIR FORCE SYSTEMS COMMAND  
UNITED STATES AIR FORCE  
L. G. Hanscom Field, Bedford, Massachusetts

ESTI PROCESSED

DDC TAB     PROJ OFFICER  
 ACCESSION MASTER FILE

\_\_\_\_\_ **AL 46216**

DATE \_\_\_\_\_ **AL 4621**

ESTI CONTROL NR. \_\_\_\_\_

CY NR \_\_\_\_\_ OF \_\_\_\_\_ CYB

*ESRM*



Project 7682, Task 768204

*AD616551*

(Prepared under Contract No. AF 19 (628)-455 by Bio-Dynamics Incorporated,  
Cambridge, Massachusetts.)

### DDC AVAILABILITY NOTICE

Copies have been deposited with the Defense Documentation Center. (DDC)

### DISSEMINATION NOTICE

DDC release to the Clearing House for Federal Scientific and Technical Information is authorized. (CFSTI) (Formerly OTS)

### LEGAL NOTICE

When US Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

### OTHER NOTICES

Do not return this copy. Retain or destroy.

ESD-TDR-64-234  
Volume IV

DESIGN AND USE OF INFORMATION SYSTEMS  
FOR  
AUTOMATED ON-THE-JOB TRAINING

VOLUME IV

Graphical Symbolology and Logic Diagrams for Use as Training Aids

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-234

JANUARY 1965

Benjamin C. Duggar  
Ronald C. Rosenberg  
Thomas B. Sheridan  
Sylvia R. Mayer

DECISION SCIENCES LABORATORY  
ELECTRONIC SYSTEMS DIVISION  
AIR FORCE SYSTEMS COMMAND  
UNITED STATES AIR FORCE  
L. G. Hanscom Field, Bedford, Massachusetts



Project 7682, Task 768204

(Prepared under Contract No. AF 19 (628)-455 by Bio-Dynamics Incorporated,  
Cambridge, Massachusetts.)

FOREWORD

One of the research goals of the Decision Sciences Laboratory is the development of design principles for automated training subsystems which could be built-into future Information Systems. Such subsystems would provide Information Systems with the capability of training automatically their own operators. To be able to design such a capability requires first the solution of many conceptual and experimental problems. This study concerns development of a logic diagraming technique for use as a training aid in those subsystems.

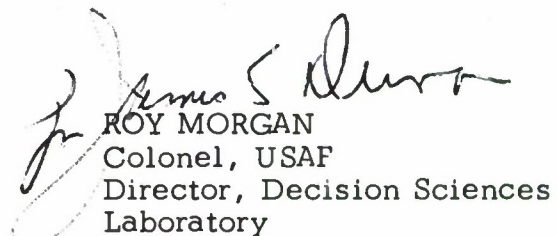
This report is one in a series supporting Task 768204, Automated Training for Information Systems, under Project 7682, Man-Computer Information Processing. The research was conducted from 1962 to 1964. The Principal Investigator was Dr. Thomas B. Sheridan and the Contract Monitor was Dr. Sylvia R. Mayer

PUBLICATION REVIEW AND APPROVAL

This Technical Documentary Report has been reviewed and is approved.

FOR THE COMMANDER

  
DONALD W. CONNOLLY  
Chief, Display Division  
Decision Sciences Laboratory

  
ROY MORGAN  
Colonel, USAF  
Director, Decision Sciences  
Laboratory

GRAPHICAL SYMBOLOGY AND LOGIC DIAGRAMS  
FOR USE AS TRAINING AIDS

Abstract

This report describes the results of a study to develop a graphical symbology and logic diagramming technique for use as a training aid. This work is addressed to the need for a language which describes the logical relationships among task components and the interactions between man and machine in advanced computer-based information systems.

Symbols and a logic diagramming technique were developed and refined by utilization with several different types of tasks. This "language" has been found to be useful for the following purposes: a) to supplement written instruction manuals, b) as an instructional tool without text, and c) as a performance aid when displayed directly on an operational console. A step-by-step methodology for constructing logic flow diagrams is presented, and applications are discussed.

## TABLE OF CONTENTS

	Page
I. Introduction	1
A. General Motivations for Using a Graphical Notation	1
B. Requirements of a Graphical Notation Describing Decision Tasks	3
II. Development of Symbols	7
A. Criteria	7
B. Graphical Information Carriers	8
C. Symbols for Man-Machine Interactions	10
III. A Logic Flow Diagramming Technique	16
A. Task Analyses	16
B. Hierarchical Levels and Layout Constraints	19
C. Examples of Task Diagrams	21
IV. Use of Logic Flow Diagrams as Teaching Aids	32
A. Pre-Instruction Requirements	32
B. Types of Tasks which May be Diagrammed	33
C. Other Applications	34
V. Conclusions and Recommendations	36
VI. References	37

## LIST OF FIGURES

		Page
Figure 1	Overall Task Diagram for Placing a Telephone Call	23
Figure 2	Diagram for Performing Tone Acquisition Routine	23
Figure 3	Diagram for Determining if the Call Went Through	24
Figure 4	Overall Task Diagram for Numbers Game Using Bookkeeping Aids	25
Figure 5	Level 2 Diagram	25
Figure 6	Level 2 Diagrams	27
Figure 7	Level 2 Diagram	28
Figure 8	Level 3 Diagrams	28
Figure 9	Logic Flow Diagram for Projection Onto Console	29
Figure 10	Procedures for Developing a Task Logic Flow Diagram	31

## I. Introduction

### A. General Motivations for Using a Graphical Notation

The general category of graphs or diagrams that is the subject of this report is represented by the linear graph of topology--a collection of dots (nodes) and lines (branches). Such a graph form is a convenient notation to use when trying to describe certain types of relationships among elements. Some typical examples are road maps, organization charts, computer flow charts, and electric circuit diagrams.

Some of the types of properties which graphs possess, in contrast to tables, lists, and other forms of organization, are well illustrated by the example of a road map. If we wished to know the approximate time to go from A to B, a point-to-point table of distances would allow us to make a rapid estimate. But this would not tell us how to get there within the estimated time. Use of the graph would tend to bog us down in a mire of detail and decision as to route, distance, traffic, probable speed, etc.. On the other hand, if we wished to chart a route, the graph (map) would be the relevant form. Again, to locate any given ~~street~~ or town, the best method is to use a street or town index and map grid cross-referencing. Maps, graphs, or charts are useful for showing the relationships among items. Single value information is best displayed in a tabular format.

The organization of sophisticated levels of information may be made possible by adopting some metrical properties in a graph. For example, a family tree which is related to a time scale in some direc-

tion can display not only genealogical relationships but also their time relations.

Although the phenomenon is not well understood, graphical forms can serve as highly efficient clues to complex behavior patterns. Many electronic technicians know little of calculus or, perhaps, algebra, yet they are able (after some experience) to relate complex circuit performance to circuit schematics. Perhaps part of the reason is to be found in the fact that people generally have a fair amount of practical contact with graphical forms (e. g. , road maps, maze games, and board games, such as chinese checkers), which may be called upon in specific context. For the electronics technician, the circuit schematic is a "language" whose meaning has become evident through repeated association. He does not have to have a specific knowledge of the "grammatical rules" to understand what it is that the schematic describes.

Another graph property of interest is the fact that, in the organizing of information, graph notation permits a certain amount of modification rather easily, as well as allowing abstractions of the information to be made in the same type of notation. There is also the possibility of attaining considerable clarity in the presentation of certain complex relationships by using a well-defined notation (e. g. , consideration of the typical task "cycle" as action-question-result-decision).

When a graph form is used as a teaching aid, it often supplements cumbersome verbal descriptions, references the items under discussion, and indicates logical relationships.

Finally, there is the possibility that such a descriptive form, when applied to certain varieties of problem-solving behavior, as indicated by the Newell-Shaw-Simon approach (1), may lead to additional insights concerning the processes involved.

More generally, the advantages of using a graph to convey a particular set of information depend upon two relations: 1) the similarity between the dimensions of the information to be represented and a two-dimensional spatial graph (the dimensions available in a two coordinate system), and 2) the similarity between the dimensional graph and the recipient's preferred modes of cognitively manipulating relationships.

B. Requirements of a Graphical Notation for Describing Decision Tasks

Arnell (7) noted that, "to an engineer, the compact symbols of graphics represent the language of communication". Because of the advantages of such specialized languages the possibility of developing an efficient symbology for man-machine tasks should be explored. The properties which are desirable in a graphical notation which is used to describe console-oriented decision-making tasks may be considered in relationship to three areas of application. These are:

1. Some properties of a good notation for abstract representation.
2. Additional properties required for use on console-oriented tasks.
3. Additional properties which enhance value as an instructional aid.

In discussing some general properties of a good man-machine notation, several ideas presented by Kajori (2), in reference to mathematical notation, are appropriate. If the symbols are to be broadly

useful, they must be defined with rigor. One such test is that the symbols be reducible to a set of machine operations, or to another statement of definition in mathematical terms. The symbols themselves should be independent of context, but should permit the appending of context as desired.

It may prove very useful to allow for an individualized version in conjunction with standard notation, where the individual part may be merged into standard form. This property may encourage evolution of the standard form, as well.

The above properties are evident in computer flow-charting notation (3). The symbols typically used can be converted (even automatically) into machine instruction sets. Clearly, the symbols themselves do not depend on the meanings or the values of any quantities actually being used. In addition, it is relatively easy to mix private notation with standard forms, and then merge the result.

When the tasks to be described are console-related, there are several features of a graphical notation which could be exploited. One of these features is that the notation can often be given a spatial relation to the actual console layout. In fact, console design may benefit from a description of a proposed task in graphical terms. The logic flow diagram may suggest console layout criteria which are not apparent in any other form of task description.

If the graphical notation reflects a reasonably sound model of the "actual" processes involved in man-machine interactions (at least

certain types), then the graph may help in identifying difficult and easy parts of the proposed task. For example, if the task description shows a sequence of many branch (decision) points, each dependent on memorization of preceding branches, operator training will be difficult.

A most promising aspect of using graphical symbology for console-oriented tasks would seem to be in instructional applications. For example, a graph which is very closely (if not isomorphically) related spatially to the console layout could be projected on the console, controlled in its colors, intensities, or other qualities, and used automatically by a teaching machine. This would be a very powerful presentation device (see Section IV of this report).

Another feature or property of graphs is the ease of adopting a convention to permit the imbedding of hierarchical detail (see Section III). For example, the first graph may show the entire task (logic and/or context) in very concise form, and any of several sections may be referenced by the end points of a branch in this graph. Any subsection form will be similar to that of the superior graph, and the structure may be repeated in several levels (see Section III). The potential training value of such an arrangement (particularly for machine teaching) is enormous and may also be valuable as a means of organizing reference material. With this arrangement, an operator can find just the level of detail he needs for any desired section of the task.

It is possible to standardize prose terminology, but the possibility

of being able to standardize a symbolic graphical notation, so as to permit a widely-transferable training in typical console-task patterns, seems even more promising. In this respect, the use of symbols for which people have a fairly common experience may be both helpful and distracting. They may be helpful because they seem "natural", but if the person's experience is contrary to our intentions, they may introduce "noise". For example, some people have difficulty in perceiving a convention which allows a variety of different node inputs, but also allows many identical outputs from the same node. However, a two-step process, using two separate symbols for input and output nodes, is usually acceptable.

The graphical task description is of use in guiding the organization of an instruction manual for the task and console. Because of its efficiency in showing the logical relationships among task components, the graphical task description may also serve as an index for the complex task instruction manual. For these applications, the notation must clearly distinguish the decisions which result in diverging branches and show where the sequences reconverge.

## II. Development of Symbols

### A. Criteria

A review of the literature related to analytic symbology did not disclose any set of generally accepted symbols for describing man-machine interactions, but did suggest a number of criteria for the development of such a symbology. The criteria which governed the selections used in this study were slight modifications of those described in an earlier report (4) and included the following:





1. The symbols must be easily learned and clearly distinguish between each of the basic elements of task cycles (action, question, and result).
2. The symbols must lend themselves to a presentation format in which the time or sequential continuum is intuitively apparent.
3. The symbology must be capable of specifying unambiguously and without redundancy the essential logical relationships of alternative actions in man-machine tasks characteristic of computer-based systems.
4. The symbols must permit the appending of context in a prominent location, and in such a way as not to distract from the graphical representation of logical relationships.

The multitude of special purpose symbologies described in computer programing or data-processing manuals, task analysis, or methods measurement texts, signal flow and circuit diagraming manuals, ASA and MIL standards for graphic symbols, etc., provides a rich background, and has suggested a number of the features of the symbology developed in this study. These special purpose symbologies have been compiled into a recently published comprehensive dictionary of standard symbols (7). The majority of basic symbols and variations of basic symbols described in reference 7 do not meet the criteria previously cited, probably because they evolved as a short-hand notation rather than as an analytic language.

B. Graphical Information Carriers

1. Lines

The primary properties of lines which are readily perceived and which can be varied to increase information content are:

- a. Continuity 
- b. Combinations
  - 1) Parallel 
  - 2) Series 
- c. Color (single or combination)
- d. Special symbols 

The following secondary properties may be varied for each of the above, but are less easily perceived:

- a. Size (thickness, length)
- b. Spacing

Graphical flow charts used as teaching aids should employ a minimum number of variable line properties which require interpretation by the learner. Continuous lines are the easiest to produce and have favorable association value. Broken lines or combination lines have high attention value if used infrequently, but will require a key if several varieties are used in the same diagram. Colors may be used on lines which are to be coded from one diagram to another, or to enhance discriminability of certain paths through the diagram. Such use does not require a color key, since the colors have no meaning and merely

enhance the discriminability of existing relationships. Considerations in selection of color codes for flow diagrams have been discussed in a previous report (5). Meanings of special symbols must be learned, or, if infrequently used, may be described in a key. Secondary properties, because of perceptual ambiguity, should not be used to increase the information content of graphs which are to be used as teaching aids.

## 2. Junctions

The angles formed at a line juncture, and the orientation of the junction, can be varied to carry information. Thus, a right angle turn could have a different meaning than would a  $45^{\circ}$  angle. A right angle turn directed down could have one meaning and an upright right angle turn another. The variations are many, limited only by the discriminability of angular differences and by the number of incident lines.

Use of a small restricted set of juncture meanings will facilitate rapid recognition and use as a teaching aid. However, maximum flexibility will be achieved only if the juncture meaning is independent of the number of converging or diverging paths.

## 3. Loops (Shapes and Figures)

The variety of loops or figures which can be devised to convey information is virtually endless. However, most of those which are easily constructed and discriminated are either three-sided, four-sided, or primarily curved arcs. In addition to the number of sides, the shape may be varied. A trapezoid may have meaning different than a square, an obtuse isosceles triangle may have meaning different than

a right triangle, and a semicircle may convey a different meaning than a circle. The orientation of the figure may also be varied to change the meaning. For example, a triangle pointing to the right could be used to carry different information than a triangle pointing to the left. Figures should be easily constructed, should be readily discriminable, even when crudely drawn, and should provide for the appending of context.

### C. Symbols for Man-Machine Interactions

The following paragraphs describe the graphical information carriers selected to show the logical relations among man-machine task elements. Other symbols could have served, but those used in this study appeared to represent the best compromise among competing criteria. Since the symbology is intended for application in teaching aids, simplicity and a minimum variety were the major considerations.

The basic task cycle usually begins with an action, or series of actions, followed by a question. The symbol for an action is a box, for a question, a double box:



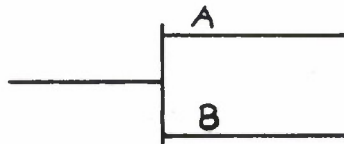
action



question

The question identifies the data requirements for evaluation of the results (in a decision context). The answer (or result) then specifies the next action. Each question symbol is followed by a decision symbol which designates the categories of results, or answers, which

specify different successive actions. The decision symbol is represented by the interaction of a horizontal and vertical line, followed by the alternative answer paths:

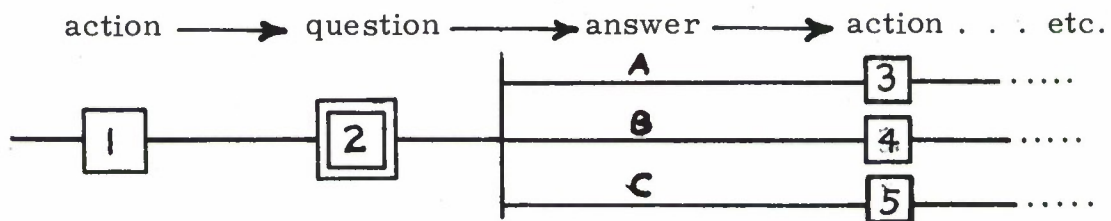


decision as to which path to follow determined by answer A or B

The set of alternative answers and successive actions must be exhaustive.

Note that although the action and decision symbols are similar to those appearing in computer flow charts, the question symbol is different (3). Computer flow charts use the diamond symbol for a question. Branches emerge directly from the diamond when there are only 2 or 3 alternative answers. The double box was used in the present study because of its efficiency for enclosing contextual information and clarity in appending answer source clues.

The complete task cycle runs from left to right:

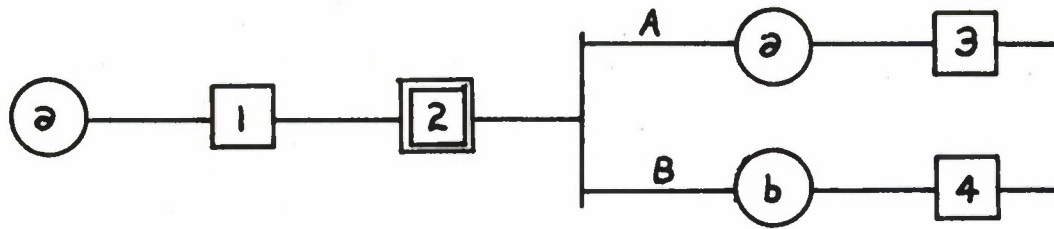


The above sequence can be described narratively as: action 1 is performed, followed by question 2. Depending upon the results of past actions, or the situation, the answer to question 2 will be A, B, or C.

The task operator decides which answer is correct and proceeds along path A, B, or C taking action 3, 4, or 5.

For single man tasks, there is no need for a symbol which designates the man. However, for multi-man systems, there may be a need to show the interactions among task components performed by different individuals. The symbol for a human operator is a circle

and is usually, but not necessarily, used at the start of a typical task cycle. For example, the following:



shows man "a" performs action 1 leading to question 2. Depending on whether the answer is A or B, man "a" continues with action 3 or man "b" continues with action 4.

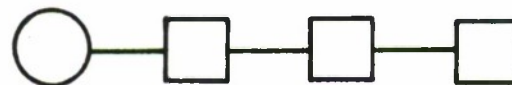
The lines connecting the symbols show the direction of flow. As long as the left to right convention is followed, there is no need for arrows. In some circumstances, this convention may not be practical (see Section IV) and arrows may be necessary.

Descriptions of many tasks can be written so as to have few or no deviations from the basic action-question-result cycles. However, in some cases, the action element may consist of a series of steps for which it is desirable to use symbols that discriminate between the ordered and the unordered "and" relationship. Similarly, the "or" branches are not always contingent upon results, or answers, to past questions and may represent a choice based on a whim. These logical relationships are illustrated as follows:

1. The "and" relation

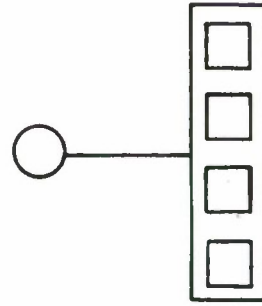
a. Sequential (ordered)

do each in turn



b. Unordered

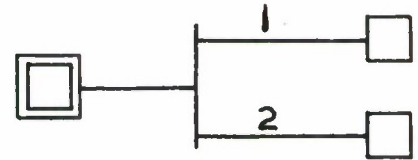
do all in any order



2. The "or" relation

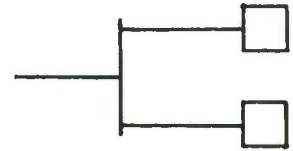
a. Contingent

answer to question  
specifies correct branch



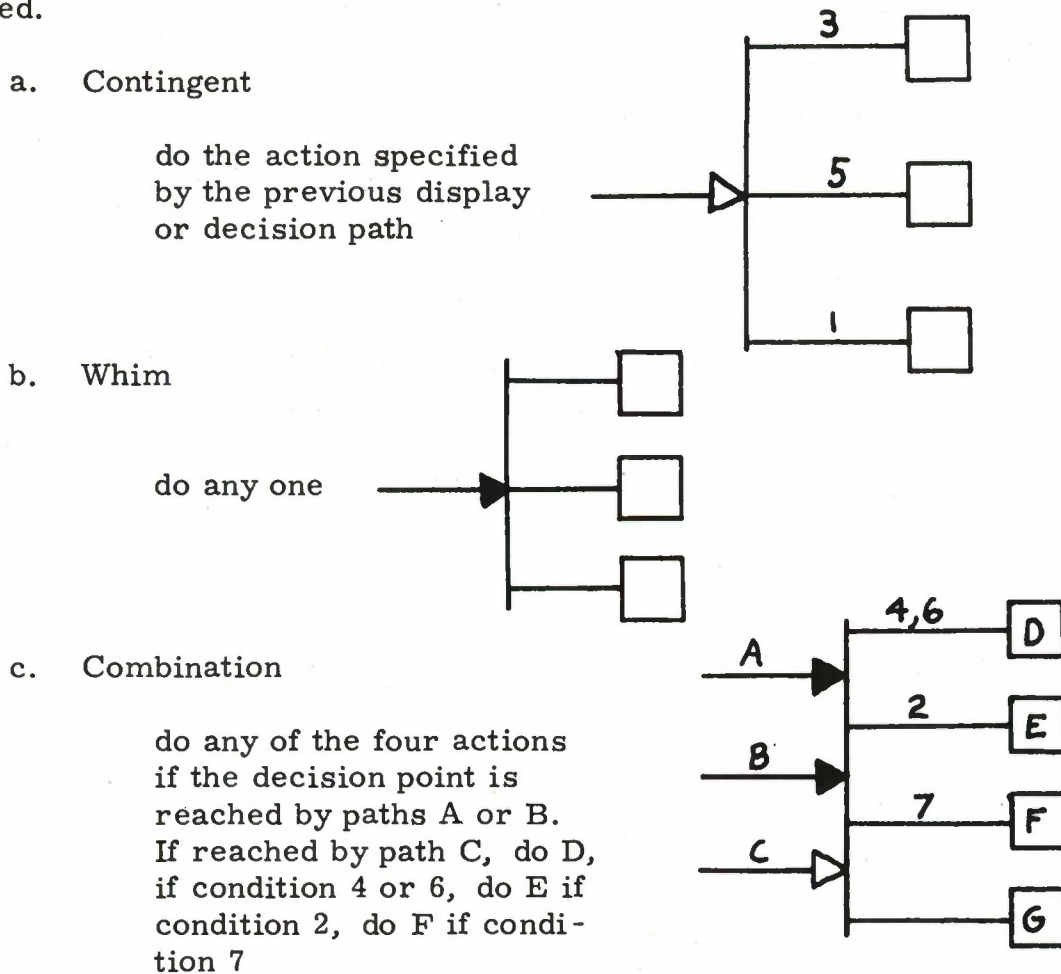
b. Whim

do any one, choice point  
does not follow a question  
and all branches are equally  
acceptable.



The preceding set of symbols has been found to be sufficient to describe a variety of tasks (see Section IV). However, in cases where the task description is to be used directly on the operational equipment, additional conventions are necessary. For example, in a previous study, logic flow diagrams were drawn directly on an abstract model of the SAGE console (4). These diagrams consisted solely of action and decision symbols keyed to a display "catalog". Sequences of actions were specified by the display status, and, although each such sequence could be described using the symbology previously discussed, simultaneous display of all such sequences would have been confusing.

Since many of the sequences of actions specified by different displays had common sub-sequences, it was possible to reduce the number of paths by utilizing decision symbols having both multiple input and output branches. To differentiate between contingent and whim "or" branches, the use of open or closed arrowheads, respectively, was adopted.

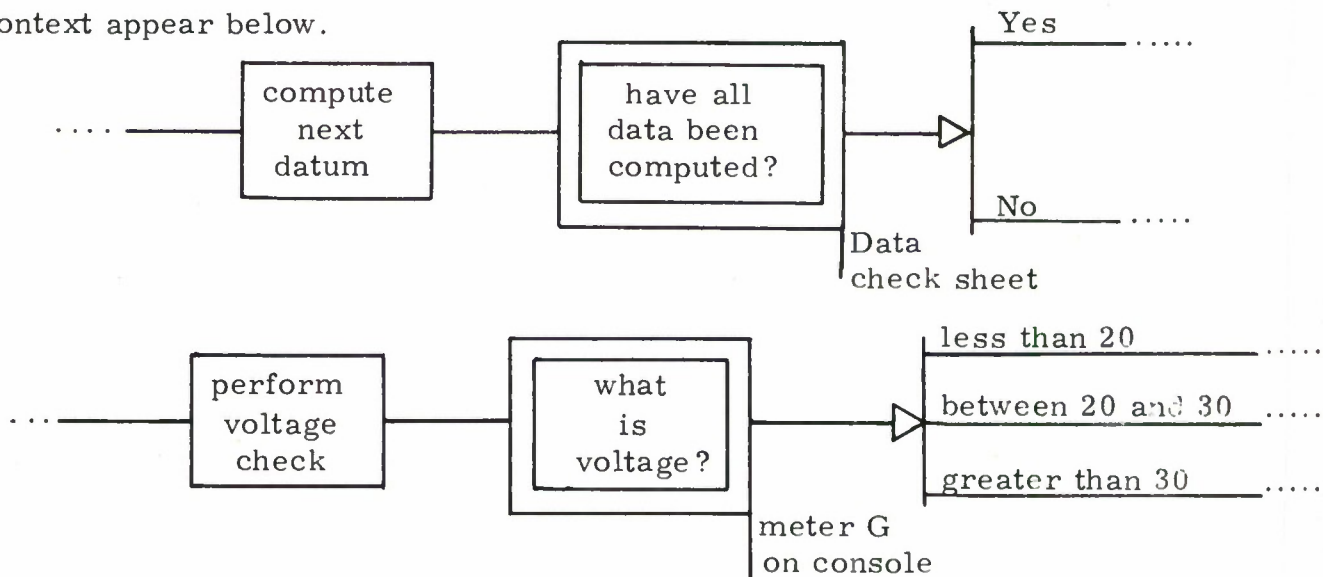


With complex tasks, the alternatives at a contingent decision point may be so numerous as to require a lengthy search. Then, performance will benefit by use of color codes keyed to the display state. Multiple color codes may follow the same direction of flow symbols

if space permits, or codes may be used only at decision branches.

Note that these logic symbols differ markedly from those specified in ASA Y32.14 or MIL-STD 806B for logic diagrams. However, the differences in symbols reflect the differences in purposes for which they were developed (analytic circuit block diagrams versus teaching aids).

Context should be included within each action and question box. Context may also be appended to the question box, placed along the direction of flow symbols after decision points, and at the beginning and end of each diagram. The information placed within the action symbol tells what action or actions to take. Depending upon the detail level of the diagram, the action may be simple (for example, "press activate action button") or complex (for example, "reinitiate action"). Within the question symbol, insert the question statement. Again, the level of complexity may vary with the detail level of the diagram (for example, "is indicator 4 on?", or "is mission proceeding satisfactorily?"). For instructional purposes, the answer source should be appended below the lower right hand corner of the question symbol. Context appearing along the direction of flow symbols at decision points refers to the previous question or display state. Examples of appended context appear below.



### III. A Logic Flow Diagraming Technique

The following paragraphs describe the step-by-step procedures for constructing logic flow diagrams. The first step consists of a detailed task analysis. The task statement is next interpreted into a series of task cycle symbols which are then combined into diagrammatic form. The diagram is then refined to meet certain format requirements, depending upon the ultimate uses anticipated. The formatting procedures can be specified as a step-by-step iterative process. Examples of task diagrams are presented below, including diagrams which describe the procedures for drawing task diagrams.

#### A. Task Analysis

A detailed task analysis represents an essential first step in the development of any teaching (program (5)). Methods of task analysis vary with the end requirements. Construction of graphic logic flow diagrams requires the identification of task goals, sequences of actions, the order in which they are performed, and the logical contingencies governing the selection of branching paths. Task analysis must therefore tabulate all actions, decisions, and goals. The logic flow diagram may then be used to show the relationships of actions and decisions to goals.

The first step in task analysis is to identify the over-all objective or goal. Next, list the actions which are normally required under unusual circumstances and identify actions which may have to be repeated.

Specify the decision which leads to use of these alternative sequences or repetitions. Insert these decisions along the sequential continuum and then add the alternative action branches. Carry each branch through until the desired goal, or best alternative, is reached. For each action, specify the qualitative and/or quantitative criteria for successful completion. Is it possible to fail to achieve any of these criteria? If so, what must be done and what is the information feedback? In some cases, success may be impossible. At what points can this be determined and what actions are then taken. The steps involved in task analysis prior to logic diagramming are illustrated in the following example.

Consider the task of telephoning an operations center for an assignment. The caller is presumed to know the correct number. The goal is to exchange information with someone at the operations center.

The sequence of actions normally required is:

1. Pick up receiver
2. Wait for dial tone
3. Dial number
4. Wait for answer
5. Exchange information

Actions which may be required under unusual circumstances include:

1. A procedure to acquire a dial tone
2. Hang up and repeat procedure if no answer (also recheck number)

3. Find another phone
4. Query answerer to ascertain if connection is correct

The decisions and criteria which lead to each of these alternative actions will now be included along the sequential continuum.

1. Pick up receiver.
2. Decide if there is a dial tone.
  - 3a. If no dial tone, perform tone acquisition procedure.
  - 4a. Decide if there is a dial tone.
  - 5a. If no dial tone, find another phone and return to beginning.
  - 3b. If there is a dial tone, dial the number.
4. Wait for answer (up to 12 rings).
5. Decide if there is an answer.
  - 6a. If no answer, hang up. Recheck number and repeat procedure.
  - 6b. If there is an answer, query the answerer.
7. Decide if the right number was reached.
  - 8a. If wrong number, hang up. Recheck number and repeat.
  - 8b. If correct number, exchange information.

The criteria for determining if there is a dial tone and whether the correct number has been reached should also be specified if the task is to be taught to a completely naive individual. Further details regarding the dial tone acquisition procedure and methods of dialing and checking the number should also be included.

The product of the task analysis becomes cumbersome with more complex tasks. Verbal descriptions of alternative courses of actions

become difficult to reference, and the relationships among diverging or converging paths become difficult to perceive. Therefore, use of logic diagrams as reference sources during the task analysis may be advantageous.

#### B. Hierarchical Levels and Layout Constraints

A logic diagram which includes a detailed breakdown of all elements of a complex task may be extremely large. A large and complex diagram is undesirable for use as a teaching aid. The relationships of individual actions and decisions to the goals become lost in detail, and more important sequences become obscured by seldom used paths. For the experienced user, the detailed logic diagram may serve as an aid in tracing out procedures for unusual circumstances, but the mass of unnecessary data may impair rapid referencing. Observations of both naive and experienced subjects' use of detailed logic flow diagrams have confirmed these drawbacks (see Section IV) and led to the development of hierarchical formats for logic flow diagrams. Format requirements will differ between diagrams used as abstract representations of the task and those used directly on the operational equipment.

Logic flow diagrams developed as abstract representations of a task should adhere to the hierarchical format. Series of minor task cycles may be compressed into a single task cycle, multiple separate actions into a single inclusive action, etc., until the over-all task can be represented by a maximum of 7-10 action and question symbols.

The resulting diagram shows only the major decision points and clearly shows the logical relationships among major alternative courses of action. The task overview diagram represents the first level of detail. Each of the action and question symbols in the level 1 diagram may be expanded into a 7-10 symbol level 2 diagram, if necessary. Similarly, the action and question symbols in level 2 diagrams may be expanded in level 3 diagrams. The steps in development of hierarchical logic flow diagrams are as follows:

1. Organize the task analysis data into chains of basic task cycles (action, question, result). Diverging chains originate at the decision point following each question. When parallel chains converge, they should do so just before an action or question symbol.
2. Eliminate duplication of basic sequences through rearrangement of diverging/converging branches or changing the scope of any individual unit (for example, an action unit could include several basic sequences which need not be specified in detail at this time). Major decision points should not be absorbed or obscured during this step; the purpose of this simplification is to eliminate detail which distracts attention from the major decision points.
3. Compress the diagram to a maximum total of 7-10 action and question symbols. The diagram will now provide a superficial overview of the entire task with the most important decision points emphasized.

4. Identify which symbol-context units are not self-explanatory --follow steps 1, 2, and 3 to diagram the sub-sequences of each such unit. Repeat this analysis of the units in the second level diagrams and form third level diagrams if needed. Continue to develop lower order diagrams until all are self-explanatory to a person versed in the symbology.

For displaying the diagram directly on operational equipment, the physical arrangement of the console will govern format, and hierarchical structure will usually not be possible. Multiple action sequences without intervening decisions, and lack of space for diagramming questions and results will be encountered. A diagram developed for direct console application has been described in a previous report (4). When the diagram was divided into successive frames for use in programmed instruction using operational equipment, console layout constraints were much less severe (see Section IV). However, for many existing consoles the exceptions to the use of recommended symbology conventions, even with a multi-frame presentation, would be so frequent as to suggest the need for individually tailored conventions.

#### C. Examples of Task Diagrams

In the course of this study, logic flow diagrams were developed for a variety of tasks. Some of these diagrams were then used as teaching aids, others served merely as examples for trying out and refining the diagramming technique. Several of the diagrams are

reproduced on the following pages.

The task of placing a telephone call, described earlier, is diagrammed in Figures 1, 2, and 3. The over-all task appears as a level 1 diagram in Figure 1. Two of the symbol-context units have been expanded into level 2 diagrams in Figures 2 and 3. To facilitate referencing, each action and question symbol has been numbered in the level 1 diagram and the corresponding number carried over to the level 2 expansion of any given unit. Alphabetic subscripts have been added in the level 2 diagram for further referencing by level 3 diagrams (not shown).

The level 1 diagram for performing the "numbers game", a querying-reasoning exercise, is shown in Figure 4. This game, described in an earlier report (4), was modified by providing the user with bookkeeping aids. The subject is seated at a desk on which are placed the logic diagrams, a stack of 3 x 5 cards, a sheet of paper with a list of 30 four-digit numbers (test numbers), and two blank tables which serve as bookkeeping aids (possible digits register and scores register). His only prior instruction concerns the meaning of symbols and abbreviations used in the diagrams. The abbreviations LTN, PDR, and SR in Figure 4 refer to the list of test numbers, possible digits register, and score register, respectively.

It was not possible to successfully teach the game by use of logic flow diagrams until a standard bookkeeping procedure was introduced.

Figure 1. Overall Task Diagram for Placing a Telephone Call

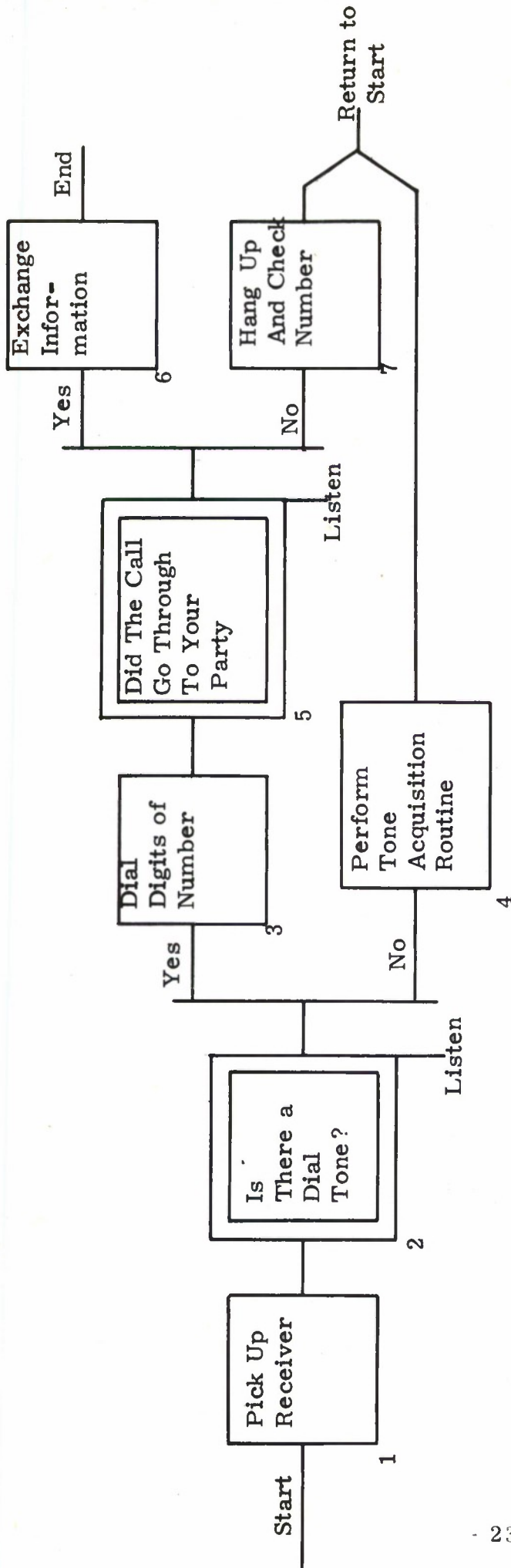


Figure 2. Diagram for Performing Tone Acquisition Routine

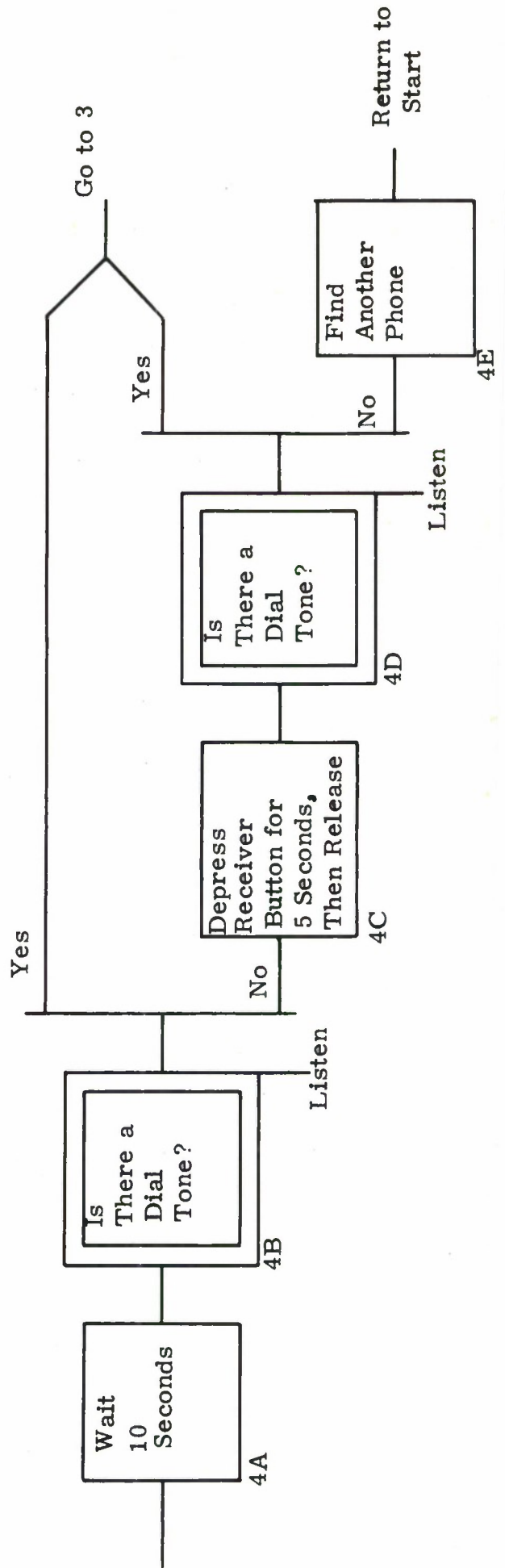


Figure 3. Diagram for Determining if the Call Went Through

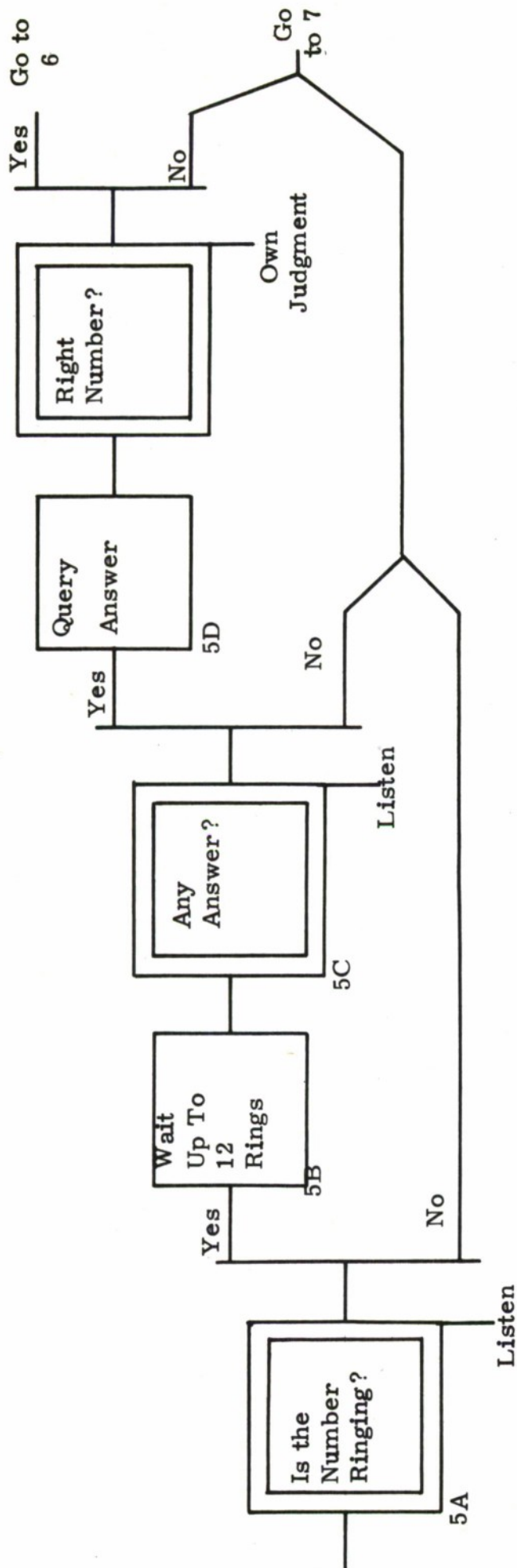


FIGURE 4. OVERALL TASK DIAGRAM FOR NUMBERS GAME USING BOOKKEEPING AIDS

OVERALL TASK DESCRIPTION

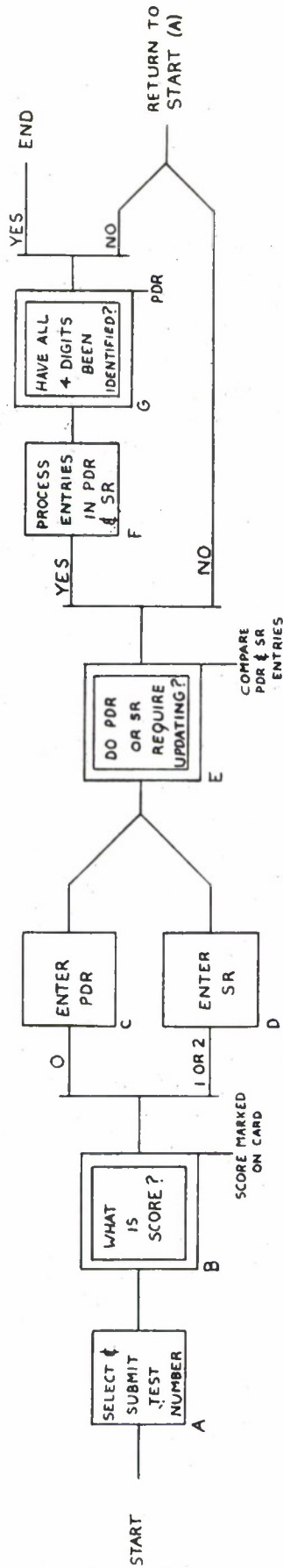
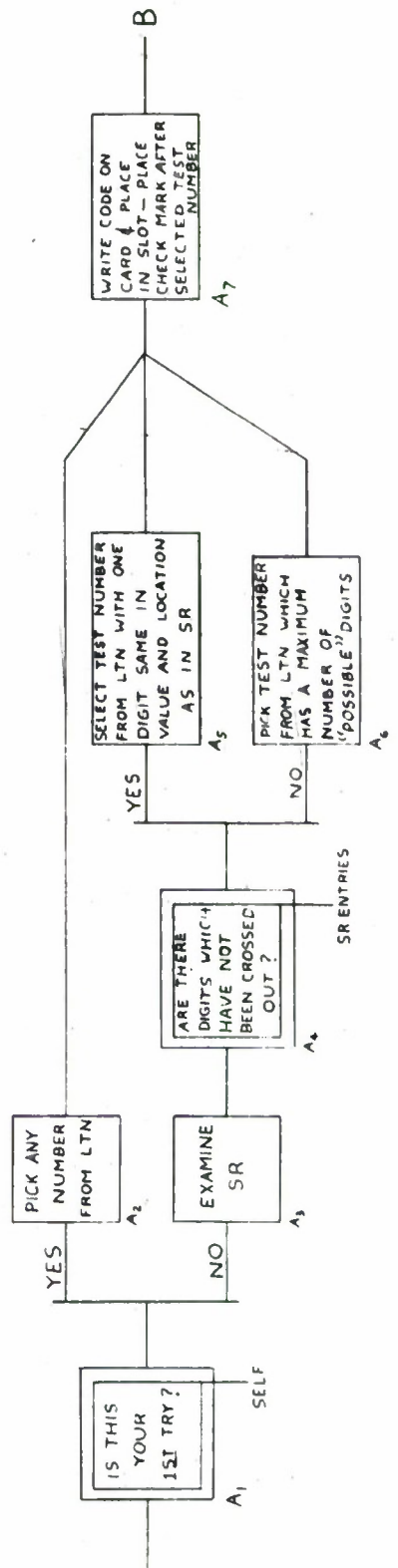


FIGURE 5. LEVEL 2 DIAGRAM

A. SELECT & SUBMIT TEST NUMBER

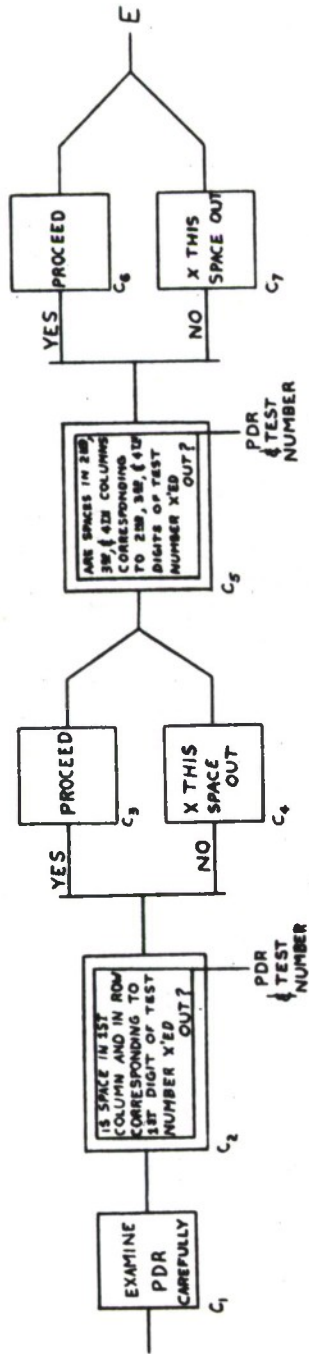


The essence of the game is to systematically glean all possible information from each successive query, and to only use test numbers which potentially will yield the maximum information. Consequently, the logic diagrams describe how to bookkeep and how to select test numbers having a maximum potential information content. The overall task diagram tells the user to start by selecting and submitting a test number. If the user doesn't know how to do this, he looks at the level 2 diagram for block A (Figure 5). After he has submitted the appropriate test number, the question is asked concerning the score he received. He is told to look at the card in order to find the score. Depending upon whether the score is zero or 1 or 2, he takes action C or D. Again, if he does not know how to enter the PDR or the SR, he consults the level 2 diagrams (Figure 6). The user progresses along until, through a process of logical deduction, he is able to identify all four digits of an "unknown" number. An additional level 2 diagram appears in Figure 7, and two level 3 diagrams appear in Figure 8.

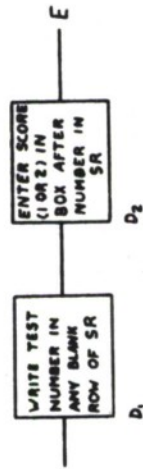
The logic flow diagram in Figure 9 was developed as one frame of a teaching program used in a teaching manual or projected directly onto an abstract SAGE console representation. The task illustrated is that of changing to intercept mode. When projected onto the abstract console, the action boxes contained the appropriate controls. Note that there are no question or contingent "or" symbols. The entire sequence of actions was coded to the display status and contains only

# FIGURE 6 LEVEL 2 DIAGRAMS

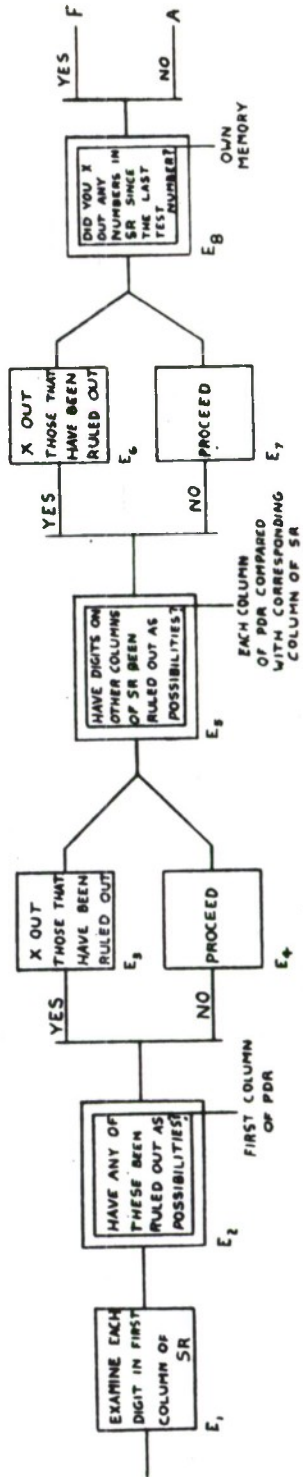
## C. ENTER ZERO SCORE IN PDR



## D. ENTER SCORE OF 1 OR 2 IN SR

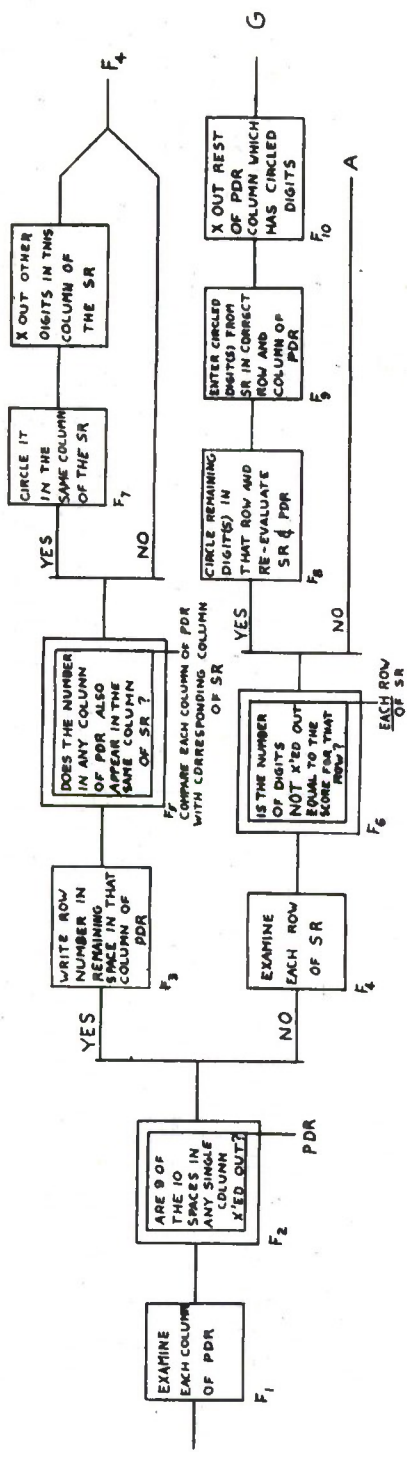


## E. DETERMINE IF PDR OR SR REQUIRE UPDATING



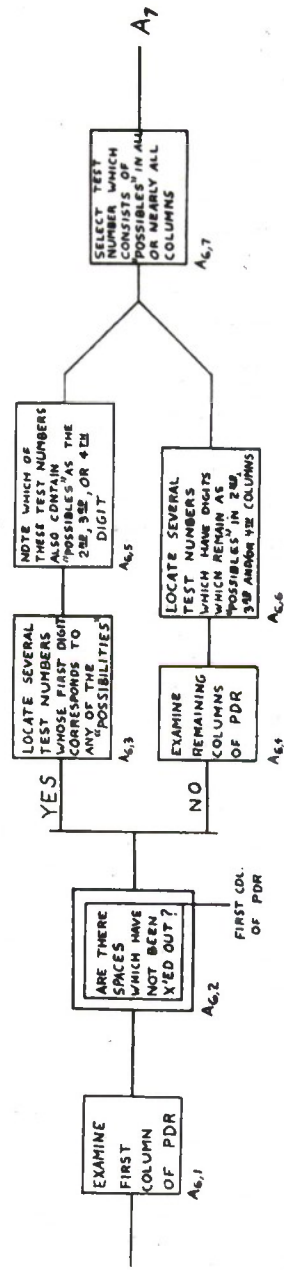
# FIGURE 7. LEVEL 2 DIAGRAM

## F. PROCESS ENTRIES IN PDR & SR

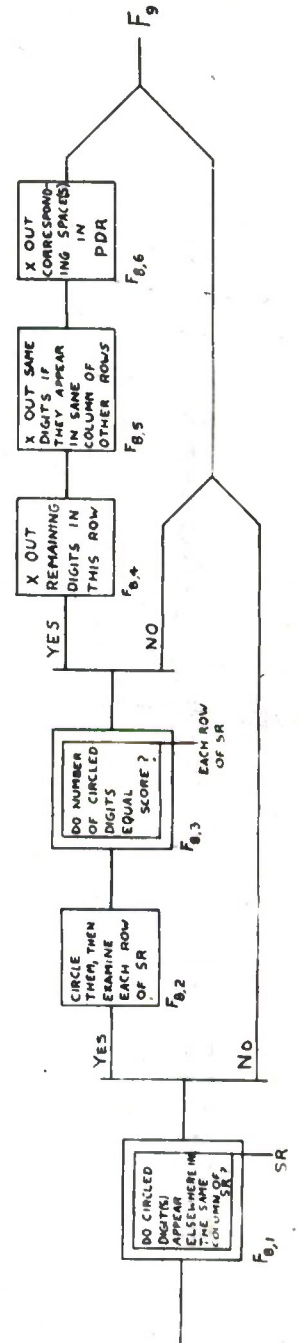


# FIGURE 8. LEVEL 3 DIAGRAMS

## A6 PICK TEST NUMBER WITH A MAXIMUM NUMBER OF "POSSIBLE" DIGITS



## F8 RE-EVALUATE SR & PDR



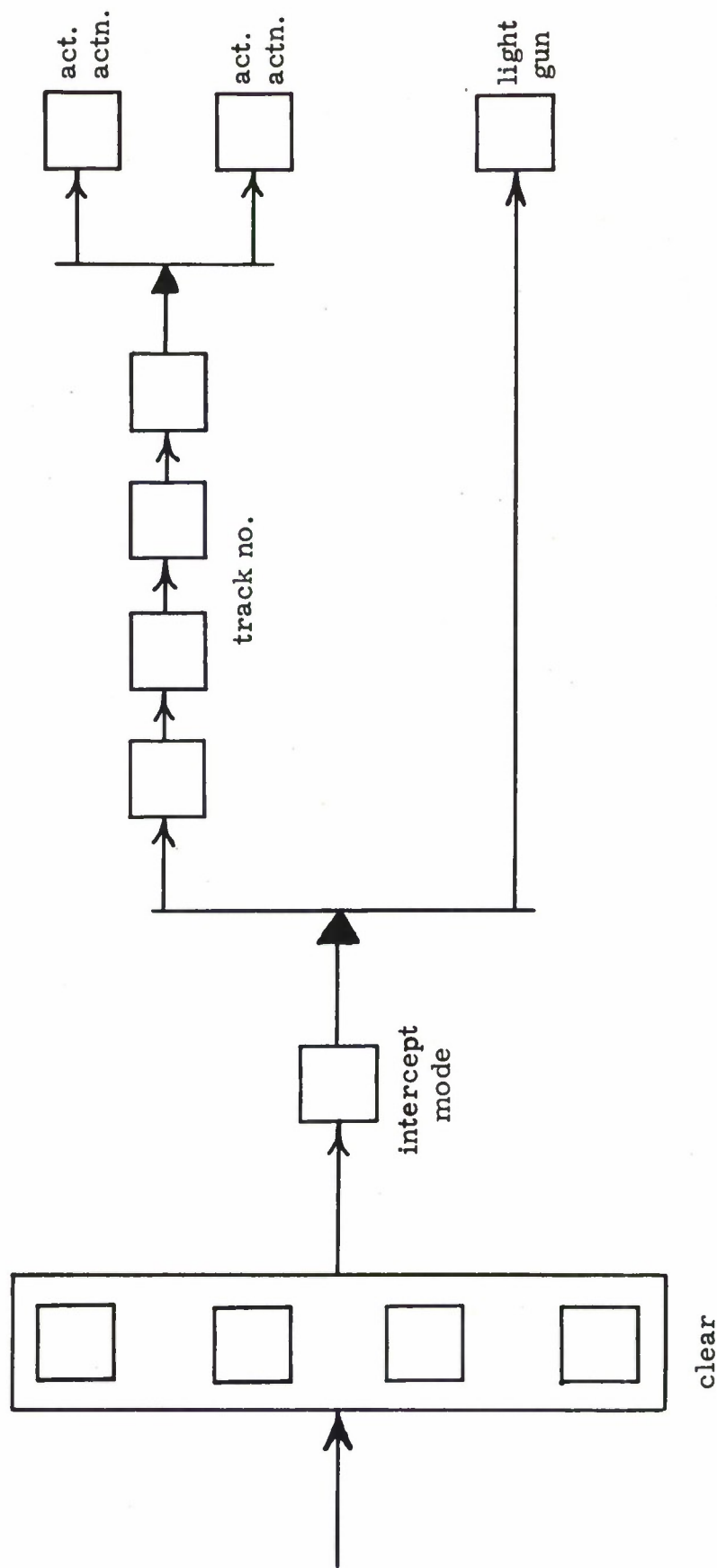


Figure 9. Logic Flow Diagram for Projection Onto Console.

ordered "and", unordered "and", and whim "or" symbols. The sequences of display changes for successful and unsuccessful missions were represented diagrammatically and used by the console operator as a reference source (a more detailed description of the diagrams used in this application appears in reference 6).

The procedures for developing logic flow diagrams are summarized by the diagram in Figure 10. Note the importance of being able to specify the questions and answers which govern the selection of alternative actions. The detailed procedures for developing logic flow diagrams are not in themselves difficult to express in a narrative fashion. Therefore, lower level diagrams are not warranted.

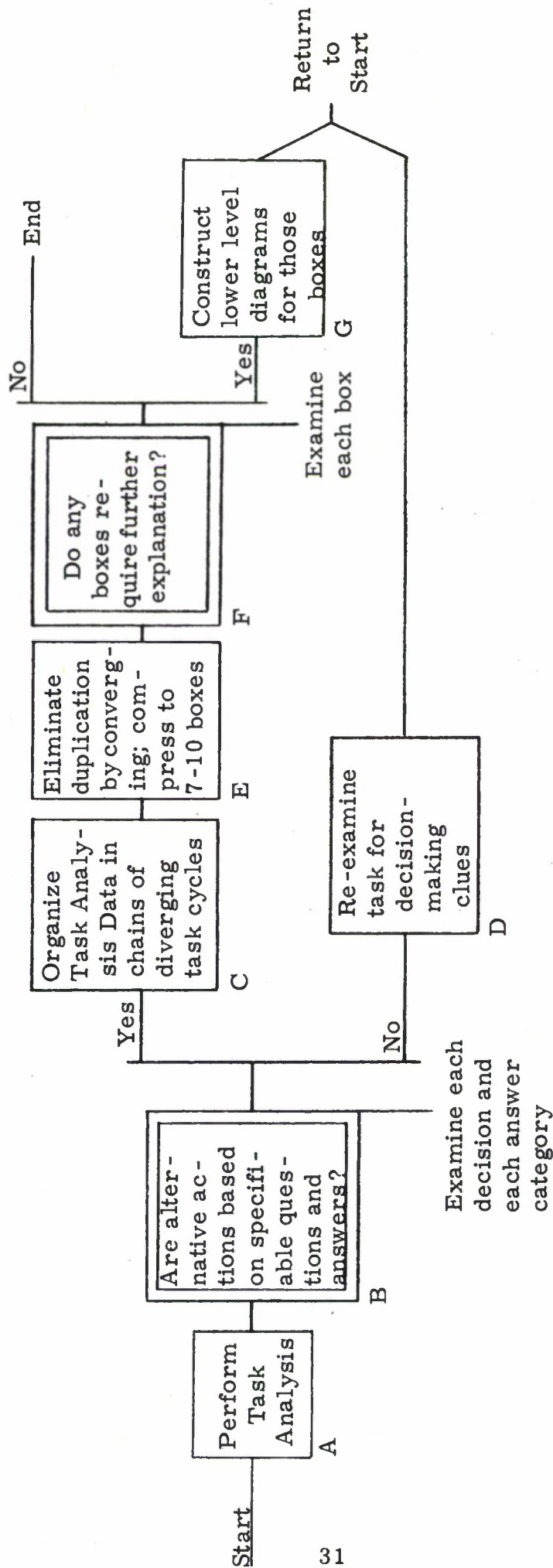


Figure 10. Procedures for developing a task logic flow diagram.

#### IV. Use of Logic Flow Diagrams as Teaching Aids

##### A. Pre-Instruction Requirements

Individuals exposed to the logic flow diagramming symbology for the first time may experience considerable difficulty in guessing the correct meanings. However, after a simple explanation or instructional course concerning the symbols, most persons can follow the diagrams without requiring further assistance. The extent of pre-instruction required has not been systematically studied. The most extensive pre-instruction provided in any of the experiments conducted to date consisted of having the subject read a four-page manual. The manual contained a description of the symbology and an example in which the task of placing a telephone call was diagrammed. Reading time averaged less than 5 minutes, after which all six subjects successfully performed an unfamiliar task (the numbers game) guided by only a set of logic flow diagrams (6).

No follow-up experiments have been conducted to determine whether the effectiveness of logic flow diagrams as a teaching aid increases when the students have had repeated past exposures to the symbology. However, experience with other graphical notation systems suggest that performance measures, such as speed of comprehension, continue to improve over periods of practice much longer than the 5 minutes used in the previously described experiment. Therefore, adoption of a standard symbology for man-machine interactions may tend to increase

its efficiency as a teaching aid beyond that demonstrated to date.

### B. Types of Tasks Which May Be Diagramed

The variety and complexity of tasks which may be described in logic flow diagrams is limited only by the skill of the task analyst. However, the value of constructing logic flow diagrams for use as teaching aids will depend on the nature of the task. Manipulative skills, for example, cannot be learned from logic flow diagrams, and extremely simple tasks will not warrant the development of logic flow diagrams. For example, the procedures for manual shutdown of a nuclear reactor could be beneficially displayed in logic flow diagram form, but a diagram is not needed to instruct the operator how to turn out the overhead lights before leaving.

Some tasks may not appear amenable to diagrammatic representation, but portions of the task, or sub-routines, may be beneficially diagramed. For example, a description of automobile driving would be difficult. However, a logic flow diagram for dealing with a car that refuses to start would be a very useful tool. Specific items should be checked in an ordered sequence, and alternative courses of action followed according to the results.

Logic flow diagrams may have their greatest potential as teaching or user aids when applied to the operation of computer-based information systems. One of the authors of this report has developed a logic flow diagram for instructional and reference use for the LOGIN

procedure on a time-shared teletype console. The relationships among the actions and results are cumbersome to express in a narrative fashion, but are easily perceived in diagrammatic form. Also, many computer-oriented individuals are adept at constructing and reading program flow charts and should have no difficulty in applying this new symbology.

In the teaching of computer programming, difficulty is often encountered with the iterative loop concept. For this reason, the diagrams which were drawn in the course of this study have avoided the use of an iterative loop notation. In cases where a loop was needed, the first step was diagrammed and an action symbol used to specify repetition of the previous sequence of steps. For example, blocks  $C_5$  and  $E_5$  in Figure 6 could have been deleted by a loop notation which guides the user back to  $C_2$  and  $E_1$ , and which increments the digit or column under consideration. It has not been established whether additional symbols are needed to specify initial conditions, indexed variables, terminating conditions, and contingent transfer back to the start of the loop.

### C. Other Applications

In addition to describing the relationships among task components, logic flow diagramming techniques may be a useful aid in the organization of instruction manuals. The instruction manuals for two dissimilar tasks have been written and organized to correspond to logic flow diagrams developed for the tasks (6). In one case, logic flow diagrams

developed for portions of the task were included in the text material. Although other techniques for organizing training manuals might serve equally well, or the same organization might have evolved without the necessity of logic diagramming, this method is suggested because of its intuitive appeal, ease of application, and ease of standardization. For speed of referencing, a table of contents can also be constructed in diagrammatic format.

Sequential action symbols have been used on equipment control panels for many years, but no formal symbology had evolved to show the logical relationships among task elements. Use of logic flow diagrams displayed directly on control panels has been shown to be feasible. Control panel layout is seldom designed to permit left to right or top to bottom flow conventions. However, as manipulative demands for the operation of controls decrease and instructional requirements increase, it may be desirable to organize the physical layout of consoles according to new criteria.

## V. Conclusions and Recommendations

A logic flow diagramming symbology has been developed and has been demonstrated to be useful as a teaching aid for a variety of tasks. A technique for constructing logic flow diagrams for virtually any type of task has been described. However, the most important area of application is for the description of decision-type tasks. In addition to use as a teaching aid, logic flow diagrams may suggest criteria for the layout of control panels and for the organization of instruction manuals.

It is recommended that logic flow diagrams be constructed which describe the operation of an advanced Air Force Information System. These diagrams should then be integrated into the existing or planned operator training program on an experimental basis. The diagrams should also be examined in relation to the console layout, and potential cost/benefit trade-offs of redesign assessed.

## VI. References

1. Newell, H., Shaw, J. C., and Simon, H. H., "Elements of a Theory of Human Problem Solving", Psychol. Rev., 65, 151-487, 1958.
2. Kajori, F., A History of Mathematical Notation, Vol. II, Open Court Publishing Co., Chicago, Illinois, 1929.
3. Flowcharting Techniques, Form C20-8152, International Business Machines Corporation, Technical Publications Department, White Plains, N. Y.,
4. Sheridan, T. B. and Mayer, S. R., Design and Use of Information Systems for Automated On-the-Job Training, I. Conceptual and Experimental Approaches, ESD TDR 64-234, U. S. Air Force, Electronic Systems Division, L. G. Hanscom Field, Bedford, Mass., December 1963.
5. Sheridan, T. B., Duggar, B. C., and Mayer, S. R., Design and Use of Information Systems for Automated On-the-Job Training, II. Design of Self-Instructional Features, ESD TDR 64-234, U. S. Air Force, Electronic Systems Division, L. G. Hanscom Field, Bedford, Mass., January 1964.
6. Design and Use of Information Systems for Automated On-the-Job Training III. Experimental Use of Three Instructional Concepts, ESD TDR 64-234, Decision Sciences Laboratory, U. S. Air Force, Electronic Systems Division, L. G. Hanscom Field, Bedford, Mass., 1965.
7. Arnell, A., Standard Graphical Symbols, McGraw-Hill Book Co., Inc., New York, 1963.
8. Mayer, S. R., Self Instruction in an Information-Processing Task, OAL-TM-59-9, Operational Applications Laboratory, L. G. Hanscom Field, Bedford, Mass., 1959.
9. Mayer, S. R., A Method for Studying Aspects of Job Design and Training Requirements of Operators in Computerized Air Defense Systems, Air Force Human Engineering, Personnel and Training Research Symposium 1960, National Academy of Sciences, National Research Council Publication 783.

Unclassified

Security Classification

**DOCUMENT CONTROL DATA - R&D**

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*

1. ORIGINATING ACTIVITY (Corporate author)  BioDynamics, Inc.		2a. REPORT SECURITY CLASSIFICATION  Unclassified	
		2b. GROUP	
3. REPORT TITLE Design & Use of Information Systems for Automated on-the-Job Training, Vol. IV, Graphical Symbology and Logic Diagrams for Use as Training Aids			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
5. AUTHOR(S) (Last name, first name, initial) Duggar, Benjamin C., Sheridan, Thomas B., Rosenberg, Ronald C., and Mayer, Sylvia R.			
6. REPORT DATE  January 1965		7a. TOTAL NO. OF PAGES  37	7b. NO. OF REFS  9
8a. CONTRACT OR GRANT NO. AF19(628)455		9a. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO. 7682			
c.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.		ESD-TDR-64-234	
10. AVAILABILITY/LIMITATION NOTICES Qualified requesters may obtain copies of this report from DDC. DDC release to CFSTI is authorized.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Decision Sciences Laboratory, Electronic Systems Division, USAF, L. G. Hanscom Field, Bedford, Mass. 01731	
13. ABSTRACT  This report describes the results of a study to develop a graphical symbology and logic diagramming technique for use as a training aid. This work is addressed to the need for a language which describes the logical relationships among task components and the interactions between man and machine in advanced computer-based information systems.  Symbols and a logic diagramming technique were developed and refined by utilization with several different types of tasks. This "language" has been found to be useful for the following purposes: a) to supplement written instruc- tion manuals, b) as an instructional tool without text, and c) as a performance aid when displayed directly on an operational console. A step-by-step methodology for constructing logic flow diagrams is presented, and applications are discussed.			

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Training Aids Information Systems Computers Man-Machine Relations Programming Language Symbols Logic Diagrams						

**INSTRUCTIONS**

**1. ORIGINATING ACTIVITY:** Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (*corporate author*) issuing the report.

**2a. REPORT SECURITY CLASSIFICATION:** Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

**2b. GROUP:** Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

**3. REPORT TITLE:** Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

**4. DESCRIPTIVE NOTES:** If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

**5. AUTHOR(S):** Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

**6. REPORT DATE:** Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

**7a. TOTAL NUMBER OF PAGES:** The total page count should follow normal pagination procedure, i.e., enter the number of pages containing information.

**7b. NUMBER OF REFERENCES:** Enter the total number of references cited in the report.

**8a. CONTRACT OR GRANT NUMBER:** If appropriate, enter the applicable number of the contract or grant under which the report was written.

**8b, 8c, & 8d. PROJECT NUMBER:** Enter the appropriate military department identification, such as project number, subproject number, system number, task number, etc.

**9a. ORIGINATOR'S REPORT NUMBER(S):** Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

**9b. OTHER REPORT NUMBER(S):** If the report has been assigned any other report numbers (*either by the originator or by the sponsor*), also enter this number(s).

**10. AVAILABILITY/LIMITATION NOTICES:** Enter any limitations on further dissemination of the report, other than those

imposed by security classification, using standard statements such as:

- (1) "Qualified requesters may obtain copies of this report from DDC."
- (2) "Foreign announcement and dissemination of this report by DDC is not authorized."
- (3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through \_\_\_\_\_."
- (4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through \_\_\_\_\_."
- (5) "All distribution of this report is controlled. Qualified DDC users shall request through \_\_\_\_\_."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

**11. SUPPLEMENTARY NOTES:** Use for additional explanatory notes.

**12. SPONSORING MILITARY ACTIVITY:** Enter the name of the departmental project office or laboratory sponsoring (*paying for*) the research and development. Include address.

**13. ABSTRACT:** Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as (TS), (S), (C), or (U).

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

**14. KEY WORDS:** Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.