Research Note 84-144

1

# MOPADS FINAL REPORT

Joseph Polito

Pritsker & Associates, Inc.

and

K.R. Laughery

Calspan Corporation

for

Contracting Officer's Representative

Charles C. Jorgensen

ARI Field Unit at Fort Bliss, Texas

Michael H. Strub

SYSTEMS RESEARCH LABORATORY

Franklin L. Moses, Acting Director

ari

U. S. Army

Research Institute for the Behavioral and Social Sciences

November 1984

# U. S. ARMY RESEARCH INSTITUTE

# FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the

Deputy Chief of Staff for Personnel

| | |
|---|---|
| | L. NEALE COSBY |
| EDGAR M. JOHNSON | Colonel, IN |
| Technical Director | Commander |

This report, as submitted by the contractor, has been cleared for release to Defense Technical Information Center (DTIC) to comply with regulatory requirements. It has been given no primary distribution other than to DTIC and will be available only through DTIC or other reference services such as the National Technical Information Service (NTIS). The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

## REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | | |
|---|---|---|
| 1. REPORT NUMBER<br>Research Note 84-144 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |

4. TITLE (and Subtitle)
MOPADS Final Report

5. TYPE OF REPORT & PERIOD COVERED
Final Report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
J. Polito - Pritsker & Associates, Inc.
K.R. Laughery - Calspan Corporation

8. CONTRACT OR GRANT NUMBER(s)
MDA-903-81-C-AA06

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Pritsker & Associates, Inc.
P.O. Box 2413
West Lafayette, IN 47906

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
2Q263739A793

11. CONTROLLING OFFICE NAME AND ADDRESS
U.S. Army Research Institute for the Behavioral and Social Sciences, 5001 Eisenhower Avenue, Alexandria, VA 22333-5600

12. REPORT DATE
November 1984

13. NUMBER OF PAGES
94

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
- -

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
- -

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
- -

18. SUPPLEMENTARY NOTES
Charles C. Jorgensen, contracting officer's representative.
This report is also published, with 2,351 pages of supporting appendices, as RN 85-145. MOPADS Final Report With Appendices.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Human Factors
Simulation
Air Defense
SAINT
MOPADS

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the MOPADS (Models of Operator Performance in Air Defense Systems) simulation system. The material is presented in four major sections. The first describes the Human Factors methodology that is being used to represent operator task performance and goal seeking behavior. Times for skill-based behaviors will be "moderated" by literature-based functions that reflect the influences of factors endogenous and exogenous to the operator. Operators sequence tasks by selecting activities that improve their goals. The

(over)

Computer program

(continued)
second section describes the models for the AN/TSQ-73 and IHAWK systems that comprise MOPADS. The operator goals and tasks that are represented are discussed. The third section discusses the computer implementation of the models. A data base is used to maintain problem parameters and to manage the communications among operators and AD units. The last section describes the documentation for MOPADS. Appendixes A - C contain reports that provide user documentation. They are mandatory reading for individuals who will design, perform, and analyze simulations using MOPADS. These documents provide sufficient information for a MOPADS user to exercise the models that exist in MOPADS. Appendixes D - J are a collection of documents for the MOPADS modeler who will design and develop MOPADS models of new air defense systems and integrate them with the rest of the MOPADS system. Appendixes K - AA are a collection of reference documents that describe the methodology and software modules of MOPADS. They are intended as reference reports of primary interest to the MOPADS modeler, although MOPADS users may find some of them interesting.

([illegible])
SAINT (Systems Analysis of Integrated Networks of Tasks).

## CONTENTS

## LIST OF APPENDIXES
(Appendixes each published under separate cover)

S. MOPADS FREE-FORMAT SYNTAX PROCESSOR (MOPADS/FFSP)

T. MOPADS USER INTERFACE (MOPADS/UI)

U. THE MOPADS DATA BASE CONTROL SYSTEM (MOPADS/DBCS)

V. MOPADS FREE-FORMAT INPUT PROGRAM (MOPADS/FFIN2)

W. DOCUMENTATION MANUAL FOR THE AN/TSQ-73 SYSTEM MODULE

X. DOCUMENTATION MANUAL FOR THE IHAWK SYSTEM MODULE

Y. THE MOPADS DATA BASE

Z. THE MOPADS DATA BASE APPLICATION PROGRAMS (MOPADS/DBAP)

AA. DOCUMENTATION MANUAL FOR THE MOPADS CONTROL MODULE (MOPADS/CNTRL) AND THE MOPADS COMMON SYSTEM MODULE PROGRAMS (MOPADS/CSMP)

## LIST OF FIGURES

## LIST OF TABLES

## CONTENTS

## LIST OF APPENDIXES
(Appendixes each published under separate cover)

S. MOPADS FREE-FORMAT SYNTAX PROCESSOR (MOPADS/FFSP)

T. MOPADS USER INTERFACE (MOPADS/UI)

U. THE MOPADS DATA BASE CONTROL SYSTEM (MOPADS/DBCS)

V. MOPADS FREE-FORMAT INPUT PROGRAM (MOPADS/FFIN2)

W. DOCUMENTATION MANUAL FOR THE AN/TSQ-73 SYSTEM MODULE

X. DOCUMENTATION MANUAL FOR THE IHAWK SYSTEM MODULE

Y. THE MOPADS DATA BASE

Z. THE MOPADS DATA BASE APPLICATION PROGRAMS (MOPADS/DBAP)

AA. DOCUMENTATION MANUAL FOR THE MOPADS CONTROL MODULE (MOPADS/CNTRL) AND THE MOPADS COMMON SYSTEM MODULE PROGRAMS (MOPADS/CSMP)

# LIST OF FIGURES

# LIST OF TABLES

# I. OVERVIEW OF MOPADS

## 1-0 INTRODUCTION TO THE MOPADS PROJECT

The MOPADS (Models of Operator Peformance in Air Defense Systems) Project has developed modeling tools to represent the performance of human beings in complex man-machine air-defense systems. The primary goals of MOPADS were to create an analysis vehicle that is:

1. flexible - in other words, able to model a wide variety of system configurations, human factors conditions, and air defense scenarios,
2. expandable - i.e., amenable to inclusion of additional elements of air defense systems and additional human factors considerations without disrupting previously existing features and with a reasonable effort, and
3. user-oriented - in other words, sufficiently easy to use so that a professional behavioral scientist can conduct meaningful experiments without performing programming or explicit computer modeling activities.

Success in achieving these goals results in a modeling tool that permits low cost analysis of human factors considerations in complex air defense situations. The analyses will be low cost to the behavioral scientist because the main expenditure of his/her time will be in creating the experimental design and in selecting the various input parameters for MOPADS. Expensive single purpose models will not need to be developed for each new application. Also, if the MOPADS system is expanded to provide new capabilities, the analyst will not have to learn a new modeling framework or new data requirements, formats, etc. Performing the experiments will be low cost also, since only a few minutes of computer time will be required for most MOPADS simulations.

## 2-0 HUMAN FACTORS MODELING IN MOPADS

Since operators are the main focus of MOPADS, the way in which human factors are represented in the models is central to the methodology. Human factors affect the simulation outcomes in two ways:

1. by affecting activity performance times, and

2. by affecting operator task sequencing.

Three types of operator activities are represented in MOPADS. The first is skill based behavior. This type of behavior involves actions requiring one or more skills such as tracking, detection, and fine manipulation. Skill based behaviors are simple control actions. Examples in the Air Defense setting are pressing appropriate buttons, entering alphanumeric values on a keyboard, and locating a symbol on a display. In MOPADS terminology, these actions are called "task elements"because they are the components of operator tasks. MOPADS task elements correspond to the lowest level of instruction information found in Army documentation. For example, when an AN/TSQ-73 operator performs a number hook, one of the actions required is to enter a track identifier on a keyboard. This activity obviously requires hand and arm motions, finger motions, eye motions, and head motion. MOPADS does not explicitly represent these components of the activity. Rather, the skills required for this action are specified, and the human factors modules compute the time required. The MOPADS model contains only a single modeling symbol that represents the keyboard entry. This means that there is a nearly one-to-one match between MOPADS model symbols and Army system documentation. Also, data collection is restricted to values which can be directly observed from operator actions.

The second type of behavior represented in MOPADS is rule based behavior. This type of behavior is typified by the performance of check lists. Much of the activity of air defense operators can be classified as rule based behavior. Operator tasks (sometimes called "critical tasks") are specified in Army documentation, and they are, in effect, check lists which the operators memorize. Operator tasks (or simply "tasks") involve skill based actions (task elements), and simple decisions. Examples include hooking a track, clearing alerts, and manually identifying a track.

Operator tasks are also obtained directly from Army documentation, and there is a nearly one-to-one correspondence between official documents and MOPADS representations of tasks. In the same way that a single MOPADS modeling symbol represents a single task element, a collection of such MOPADS symbols represents an operator task.

The third type behavior represented in MOPADS is knowledge based behavior. This type of behavior is strategic in nature. Air defense operators perform this type of behavior in selecting which tasks to perform in order to accomplish their mission. In particular, the operators must decide which operator tasks to perform and in what order. The MOPADS term for this activity is "task sequencing." MOPADS operators are represented as goal seekers. They evaluate the potential impact of available operator tasks on their goals when selecting the next task to perform.

It is more difficult to obtain Army documentation references for operator goals because they result from common practice, standard operating procedures and individual operator motivations. The approach taken in MOPADS has been to consult subject matter experts (in addition to Army documentation) to determine a sufficient set of operator goals.

It is clear that the way human factors modeling is performed in MOPADS will be the central concern of behavioral scientists and that no such methodology is sufficiently well established so as to elicit no controversy. Therefore, the human factors modules developed for MOPADS are stand-alone software modules. This means that other researchers will be able to experiment with the methodology in a context removed from the MOPADS project. Furthermore, the modules are sufficiently well documented so that other researchers can test alternate parameter values and even substitute human performance equations.

This approach has benefits to MOPADS in that alternate human factors representations can be readily tested within the MOPADS system, and it will hopefully provide a useful tool for behavioral scientists to evaluate theories in a unified framework.

## 3-0 SIMULATION METHODOLOGY IN MOPADS

The simulation methodology selected for MOPADS is discrete event simulation. This means that the computer explicitly represents each action and event ( to the level of detail selected by the modeler) that occurs in the system. This is in contrast to algebraic or differential equation models which aggregate and smooth individual events to obtain overall average performance measures.

The advantages, in the MOPADS context, of a discrete event simulation are:

1. An actual time history of events is produced by the simulation. This can be important for interfacing the simulation with real time hardware simulators or field equipment, since the simulator events will more closely approximate the events in the "live" systems.

2. Discrete event modeling provides the potential for a higher degree of fidelity than do more aggregated techniques. The degree of detail can be determined by the modeler, and individual subsystems can be selectively aggregated or disaggregated as required.

3. It allows the introduction of human factors considerations at the level at which they naturally occur. In other words, individual operator actions can be affected rather than some performan' ? measure aggregated over many actions.

The SAINT simulation language has been selected as the host language for the MOPADS operator models. SAINT is an acronym for Systems Analysis of Integrated Networks of Tasks. It was initially developed by Pritsker & Associates, Inc. for the U. S. Air Force to model human performance in man-machine systems and has had numerous applications including modeling of operators of remotely piloted vehicles.

The unique feature of SAINT is that it provides a formal capability to introduce human factors considerations. This is done using "moderator functions" which modify the nominal time to perform tasks. The modification, of course, is based upon the operator's ability to perform the task at that time. Thus, SAINT has features which make it immediately useful for constructing models in MOPADS.

## 4-0 MOPADS TERMINOLOGY

The remainder of this report discusses the above topics in greater detail, and it will be helpful if the reader familiarizes himself/herself with the "Standard MOPADS Terminology" contained in Table I-1.

Table I-1. Standard MOPADS Terminology

| | |
|---|---|
| AIR DEFENSE SYSTEM | A component of Air Defense which includes equipment and operators and for which technical and tactical training are required. Examples are IHAWK and the AN/TSQ-73. |
| AIR DEFENSE SYSTEM MODULE | Models of operator actions and equipment characteristics for Air Defense Systems in the MOPADS software. These models are prepared with the SAINT simulation language. Air Defense System Modules include the SAINT model and all data needed to completely define task element times, task sequencing requirements, and human factors influences. |
| AIR SCENARIO | A spatial and temporal record of aerial activities and characteristics of an air defense battle. The Air Scenario includes aircraft tracks, safe corridors, ECM, and other aircraft and airspace data. See also Tactical Scenario. |
| BRANCHING | A term used in the SAINT simulation language to mean the process by which TASK nodes are sequenced. At the completion of the simulated activity at a TASK node, the Branching from that node determines which TASK nodes will be simulated next. |
| DATA BASE CONTROL SYSTEM | That part of the MOPADS software which performs all direct communication with the MOPADS Data Base. All information transfer to and from the data base is performed by invoking the subprograms which make up the Data BAse Control System. |
| DATA SOURCE SPECIALIST | A specialist in obtaining and interpreting Army documentation and other data needed to prepare Air Defense System Modules. |

ENVIRONMENTAL STATE VARIABLE

An element of an Environmental State Vector.

ENVIRONMENTAL STATE VECTOR

An array of values representing conditions or characteristics that may affect more than one operator. Elements of Environmental State Vectors may change dynamically during a MOPADS simulation to represent changes in the environment conditions.

MODERATOR FUNCTION

A mathematical/logical relationship which alters the nominal time to perform an operator activity (usually a Task Element). The nominal time is changed to represent the operator's capability to perform the activity based on the Operator's State Vector.

MOPADS DATA BASE

A computerized data base designed specifically to support the MOPADS software. The MOPADS Data Base contains Simulation Data Set(s). It communicates interactively with MOPADS Users during pre- and post-run data specification and dynamically with the SAINT software during simulation.

MOPADS MODELER

An analyst who will develop Air Defense System Modules and modify/develop the MOPADS software system.

MOPADS USER

An analyst who will design and conduct simulation experiments with the MOPADS software.

MSAINT (MOPADS/SAINT)

The variant of SAINT used in the MOPADS system. The standard version of SAINT has been modified for MOPADS to permit shareable subnetworks and more sophisticated interrupts. The terms SAINT and MSAINT are used interchangeably when no confusion will result. See also, SAINT.

OPERATOR STATE VARIABLE

One element of an Operator State Vector.

OPERATOR STATE VECTOR

An array of values representing the condition and characteristics of an operator of an Air Defense System. Many values of the Operator State Vector will change dynamically during the course of a MOPADS simulation to represent changes in operator condition.

OPERATOR TASK

An operator activity identified during weapons system front-end analyses.

SAINT

The underlying computer simulation language used to model Air Defense Systems in Air Defense System Modules. SAINT is an acronym for Systems Analysis of Integrated Networks of Tasks. It is a well documented language designed specifically to represent human factors aspects of man/machine systems. See also MSAINT.

SIMULATION DATA SET

The Tactical Scenario plus all required simulation initialization and other experimental data needed to perform a MOPADS simulation.

SIMULATION STATE

At any instant in time of a MOPADS simulation the Simulation State is the set of values of all variables in the MOPADS software and the MOPADS Data Base.

SYSTEM MODULES

See Air Defense System Modules.

TACTICAL SCENARIO

The Air Scenario plus specification of critical assets and the air defense configuration (number, type and location of weapons and the command and control system).

TACTICAL SCENARIO COMPONENT

An element of a Tactical Scenario, e.g., if a Tactical Scenario contains several Q-73's, each one is a Tactical Scenario Component.

TASK

See Operator Task.

TASK ELEMENTS

Individual operator actions which, when grouped appropriately, make up operator tasks. Task elements are usually represented by single SAINT TASK nodes in Air Defense System Modules.

TASK NODE

A modeling symbol used in the SAINT simulation language. A TASK node represents an activity; depending upon the modeling circumstances, a TASK node may represent an individual activity such as a Task Element, or it may represent an aggregated activity such as an entire Operator Task.

TASK SEQUENCING MODERATOR FUNCTION

A mathematical/logical relationship which selects the next Operator Task which an operator will perform. The selection is based upon operator goal seeking characteristics.

# II. MOPADS HUMAN FACTORS MODELING

## 1-0 OVERVIEW OF HUMAN FACTORS IN MOPADS

MOPADS models consist of networks of tasks, and each task is a network of task elements. The way in which the tasks are connected in the network represents potential operator decisions on sequencing of tasks. Human factors will affect this system in two ways:

1. The time to perform task elements is affected ("moderated") by environmental and operator conditions. Thus, the values of independent variables such as lack of sleep, hours of continuous duty, amount of training, and tracking ability as well as environmental variables such as ambient temperature may affect the time required to perform tasks.

2. The order in which the operator performs operator tasks will be determined by the degree which a particular task will help to satisfy the operator's goals. When a task is completed, the operator will evaluate the state of each goal and estimate the impact on these goals from the various options at hand. A task will be selected based on the operator's objective and his estimates of the probable goal improvement.

## 2-0 TASK ELEMENT MODERATOR FUNCTIONS

### 2-1. The MOPADS Skills Taxonomy.

As discussed earlier, task elements are generally the lowest level of activity described in Army documentation. Since these actions are directly observable, it is relatively easy to obtain estimates of the nominal time to perform them. Such times, however, will be estimates of the "nominal" time under average conditions and by average operators. In order to systematically estimate the effect on operator performance of environmental and scenario specific conditions, the task element performance times must be known as functions of these variables.

MOPADS uses the concept of a "moderator function" to accomplish this. The moderator functions accept the nominal task element performance time and alter it ("moderate it") based on endogenous and exogenous environmental variables. In this way, an operator's performance is affected dynamically during the simulation by the changing conditions of the system. A major activity of the MOPADS project has been the development of the moderator functions. Single observations of operators is not a practical method to obtain such functions because of the many variables and conditions that would need to be controlled.

Consider the action of an AN/TSQ-73 operator. There are literally dozens of task elements which this operator performs. It would be a prohibitively expensive chore to develop separate moderator functions for each task element. To do so would require that operators be observed performing each task element under controlled conditions while varying the independent variables of interest. Then the data would need to be reduced to a functional form suitable for use in MOPADS models.

A moderator function approach was needed that would be generic in nature. In other words, one that would be applicable to more than one task element. The approach selected was to consider each task element as an activity that requires one or more operator skills. The human factors literature contains a good deal of data on how skill performance varies with a wide variety of independent variables. The idea was to develop skill moderator functions for the skills required by air defense operators, and then to combine these functions according to the skills required for a particular task element to obtain a combined moderator tailored for the task element. A skills taxonomy similar to the one presented in Finley, Obermeyer, Bertone, Meister, & Muckler(1970) was selected since it includes skills required by air defense operators. It is shown in Table II-1.

Moderator functions for each of these skills have been developed. Each of the operator task elements is considered to require a combination of one or more of the skills shown in Table II-1. A moderator function for a particular task element is evaluated by combining the moderators for the component skills as explained in Section 2-2 below. In this way, moderators for a wide variety of air defense operator actions are available from a relatively small set of skill moderator functions.

The skill moderator functions were developed from the open human factors literature. The following five steps were performed to develop them.

Table II-1. MOPADS Skills Taxonomy.

| | |
|---|---|
| 1 | Probability Estimation |
| 2 | Time Estimation |
| 3 | Long Term Memory-Sensory |
| 4 | Long Term Memory-Symbolic |
| 5 | Short Term Memory-Sensory |
| 6 | Short Term Memory-Symbolic |
| 7 | Numeric Manipulation |
| 8 | Recognition |
| 9 | Unused |
| 10 | Unused |
| 11 | Timesharing |
| 12 | Detection |
| 13 | Fine Manipulation |
| 14 | Gross Manipulation |
| 15 | Unused |
| 16 | General Physical Effort |
| 17 | Reaction Time |
| 18 | Tracking |
| 19 | Team Coordination |

1. Literature Review
2. Development of a Computer Data Base
3. Selecting Independent Variables
4. Developing Moderator Functions
5. Cross-checking the Moderator Functions with a Structural Model of the Human

Since the moderator functions are to refelct the current state-of-the-art in quantitative human performance modeling, the first step was to review the current literature. This was completed and documented in Laughery (1981a). The next step was to organize this literature in a meaningful way. One part of the organization was already defined by the skill categories. Each human performance model in an article was categorized by the skill it modeled. In some cases the model grouped several skill categories (e.g., detection and identification). In this event, it was assigned to multiple skill categories. Secondly, a model within an article could be characterized by the independent variables which moderated performance of the skill. By compiling these "data" on the literature, it was possible to select all articles involving a particular skill and examine the independent variables included in the models.

To facilitate rapid analysis of the literature data base, a set of computer programs and files for data base management were prepared (Laughery, 1981b). As the literature was reviewed and the data entered into the data base, no screening was performed on the information. In other words, each time a new independent variable was encountered, it was added to the list of variables. The extent of the literature review was constrained by resources and time available and by the skills taxonomy which bounded the set of pertinent literature. The entire data base and the data base programs have been delivered to the government for further developemnt if warranted.

When the literature search was complete, it was necessary to reduce the information collected to a coherent set of moderator functions. Not all of the independent variables represented in the data base would be relevant to MOPADS, and the issue of "sufficiency" needed to be addressed. In other words, was the set of independent variables sufficient to model air defense operators.

Some type of cross-check was required. Relying solely on the literature to identify which independent variables affect skills assumes that all relationships have been studied and reported in the open literature. Since this is surely not the case, a conceptual model of the human was developed from a slightly different perspective (Laughery & Ditzian, 1981). Rather than treating the human as a performer of different types of skills, a model was developed of the human with respect to functional systems (e.g., visual, auditory, memory). Those human systems involved in each skill category from the taxonomy were then intuitively identified and arrayed into a "human systems" by "skills" matrix. As the list of independent variables was developed, those independent variables which were intuitively expected to interact with the human systems were identified and documented in a "human systems" by "independent variables" matrix. Finally, a list of hypothesized independent variables was linked to every skill by crossing the "human systems" by "skills" matrix with the "independent variables" by "human systems" matrix, resulting in an "independent variables" by "skill categories" matrix. This matrix was compared with the moderator functions for each skill category to see if all theoretically related independent variables were included.

Promising articles in the data base were examined in greater detail and their performance models compared with the intuitive human function model. Two articles that significantly affected the final model were Siegel, Pfeiffer, Kopstein, Wilson, & Ozkaptan (1979) and Pew, Baron, Seehrer, & Miller (1977). The final set of independent variables for MOPADS is shown in Table II-2.

The three categories of independent variables specify the scope of each category of variables. Operator variables constitute the operator's state vector. Each operator has his own set of values for these variables.

The environmental variables form the environmental state vector. These variables affect all operators in the same environment. For example, both operators in an AN/TSQ-73 are affected by the same environmental state vector. Finally, task variables apply to any operator that performs the task, and each task has its own set of values for each of the variables.

Obviously, some of the independent variables do not apply to operators of the AN/TSQ-73 and IHAWK. They have been included, however, because the objective in developing the taxonomy was to characterize the skill requirements of most air defense operators. Thus, MOPADS can be expanded at a later time to include other air defense systems such as Redeye, Vulcan, etc. All of the independent variables shown in Table II-2 have been implemented in the current MOPADS software. Since only the AN/TSQ-73 and IHAWK are modeled, however, not all of the variables are used. This causes no difficulty in the present models, but it will greatly facilitate future expansions.

## 2-2. Computation of Combined Task Element Moderator Functions.

The current implementation of the MOPADS skill moderator functions affect only the mean task element time. The software has been developed in such a way that, if at some time appropriate data become available, then the standard deviation and distribution function can also be moderated.

Consider the operator task shown in Figure II-1 for the AN/TSQ-73, and the aggregate SAINT task model shown in Figure II-2. The trapezoids labeled 48 in Figure II-1 corresponds to the SAINT "task node" numbered 48 (with LABL: NUMHOOK) in Figure II-2. Task node 48 in Figure II-2 is the SAINT modeling symbol that corresponds to the task elements "ENTER TRACK NUMBER, FIRE UNIT, OR SITE ADDRESS ON A/N KEYBOARD" and "PRESS TASK FUNCTIONS-NUMBER HOOK".

Table II-2. MOPADS Independent Variables.

OPERATOR STATE VARIABLES

| | |
|---|---|
| 1 | CORE TEMPERATURE |
| 2 | CIO VALUE |
| 3 | TIME ON TASK |
| 4 | DAYS OF DUTY |
| 5 | SEARCH DIMENSIONS |
| 6 | NUMBER FIRE UNITS |
| 7 | PERCENTAGE RECOVERY |
| 8 | PREVIOUS WORK |
| 9 | PREVIOUS REST |
| 10 | FLASH INTENSITY |
| 11 | TARGET SPEED |
| 12 | TARGET TYPE |
| 13 | TARGET SIZE |
| 14 | TARGET COLOR |
| 15 | SEARCH AREA |
| 16 | BINOCULAR USAGE |
| 17 | SLANT RANGE TO TARGET |
| 18 | TARGET TRAJECTORY |
| 19 | TARGET BACKGROUND COMPLEXITY |
| 20 | NUMBER BACKGROUND CHARACTERS |
| 21 | MESSAGE BACKLOG |
| 22 | SIGNALS PER MINUTE |
| 23 | HOURS WORKED PER WEEK |
| 24 | DAYS WITHOUT SLEEP |
| 25 | DAYS OF NIGHT DUTY |
| 26 | SIMULTANEOUS TASKS |
| 27 | CONTRAST RATIO |
| 28 | AVERAGE HOURS SLEEP |
| 29 | OBJECTIVE FUNCTION |
| 30 | GOALS CONSIDERED |
| 31 | TARGET BRIGHTNESS |
| 32 | NIGHTS |
| 33 | SKY GROUND RATIO |
| 34 | AIRCRAFT ALTITUDE |
| 35 | METEOROLOGICAL RANGE |
| 36 | THRESHOLD CONTRAST |
| 37 | TARGET HEIGHT |
| 38 | TARGET WIDTH |
| 39 | TARGET DEPTH |
| 40 | HORIZONTAL RANGE |
| 41 | NUMBER OF RESOLUTION ELEMENTS |
| 42 | NUMBER OF SUSPECT AREAS |
| 43 | AIRCRAFT SPEED |
| 44 | FIELD OF VIEW |
| 45 | OBSERVER OFFSET |

Table II-2 (continued)

| | |
|---|---|
| 46 | DISPLAY TARGET LOCATION |
| 47 | TARGET LOCATION |
| 48 | DISPLAY RESOLUTION |
| 49 | DISPLAY BACKGROUND HEIGHT |
| 50 | DISPLAY BACKGROUND WIDTH |
| 51 | DISPLAY BACKGROUND DEPTH |
| 52 | DISTANCE TO DISPLAY |
| 53 | DISPLAY HEIGHT |
| 54 | DISPLAY WIDTH |
| 55 | TARGET NOISE LEVEL |
| 56 | TARGET DURATION |
| 57 | EXPERIENCE |
| 58 | SIGNAL PROBABILITY |
| 59 | REST PERIODS |
| 60 | DAYS SINCE PRACTICE |
| 61 | SENSE OF DIRECTION |
| 62 | SKIN TEMPERATURE |
| 63 | TIME IN TEMPERATURE |
| 64 | PREVIOUS SKIN TEMPERATURE |

ENVIRONMENTAL VARIABLES

| | |
|---|---|
| 1 | DRY BULB TEMPERATURE |
| 2 | RELATIVE HUMIDITY |
| 3 | AIR MOVEMENT RATE |
| 4 | NOISE LEVEL |
| 5 | WORK AREA ILLUMINATION |
| 6 | NUMBER ON DUTY |
| 7 | VIBRATION |
| 8 | AMBIENT VAPOR PRESSURE |
| 9 | NOISE PREDICTABILITY |

TASK RELATED VARIABLES

| | | |
|---|---|---|
| 1 | | KILOCALORIES/MINUTE |
| 2 | | NUMBER OF BRANCHES OUT |
| 3 | * | STIMULUS MODE 1 |
| 4 | * | STIMULUS MODE 2 |
| 5 | * | RESPONSE MODE |
| 6 | * | OBSERVER TARGET POSITION |
| 7 | | CONTROL DISTANCE |
| 8 | | CONTROL WIDTH |
| 9 | | NUMBER OF DISPLAYS |
| 10 | | NUMBER OF ALTERNATIVES |
| 11 | | NUMBER SHORT TERM MEMORY ITEMS |

* See Legend

Table II-2 (continued)

Legend

Stimulus Mode 1

A two-digit number
10's digit - 0 - no visual
1 - visual
1's digit 0 - no auditory
1 - auditory

Stimulus Mode 2

A three-digit number as above
100's digit - 0 - no olfactory
1 - olfactory
10's digit 0 - no kinesthetic
1 - kinesthetic
1's digit 0 - no tactile
1 - tactile

Response Mode
A two-digit number
10's digit - 0 - no vocal
1 - vocal
1's digit 0 - no tactile
1 - tactile

Observer Target Position
1 - ground to ground
2 - air to ground
3 - ground to air
4 - air to air
5 - at a display

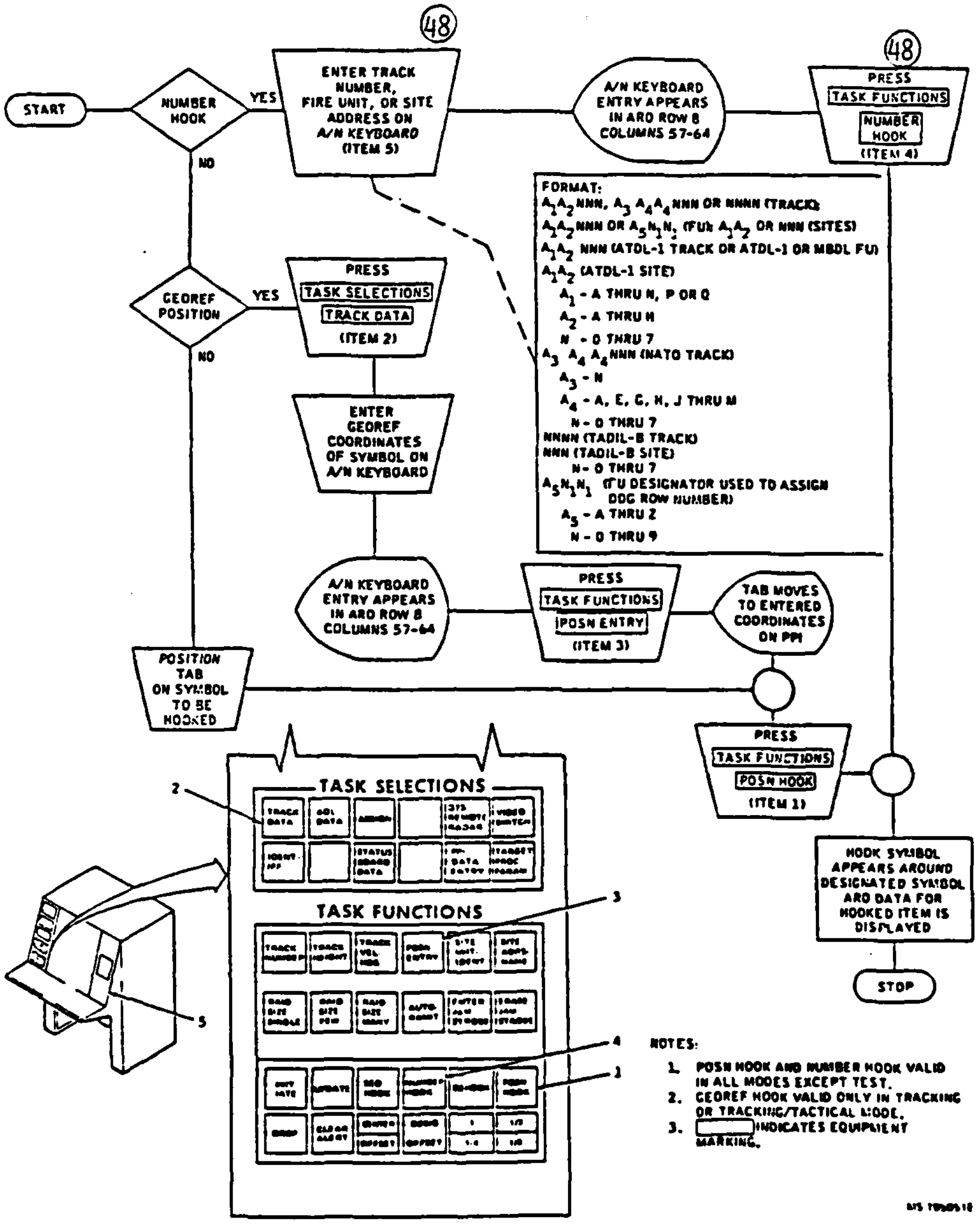
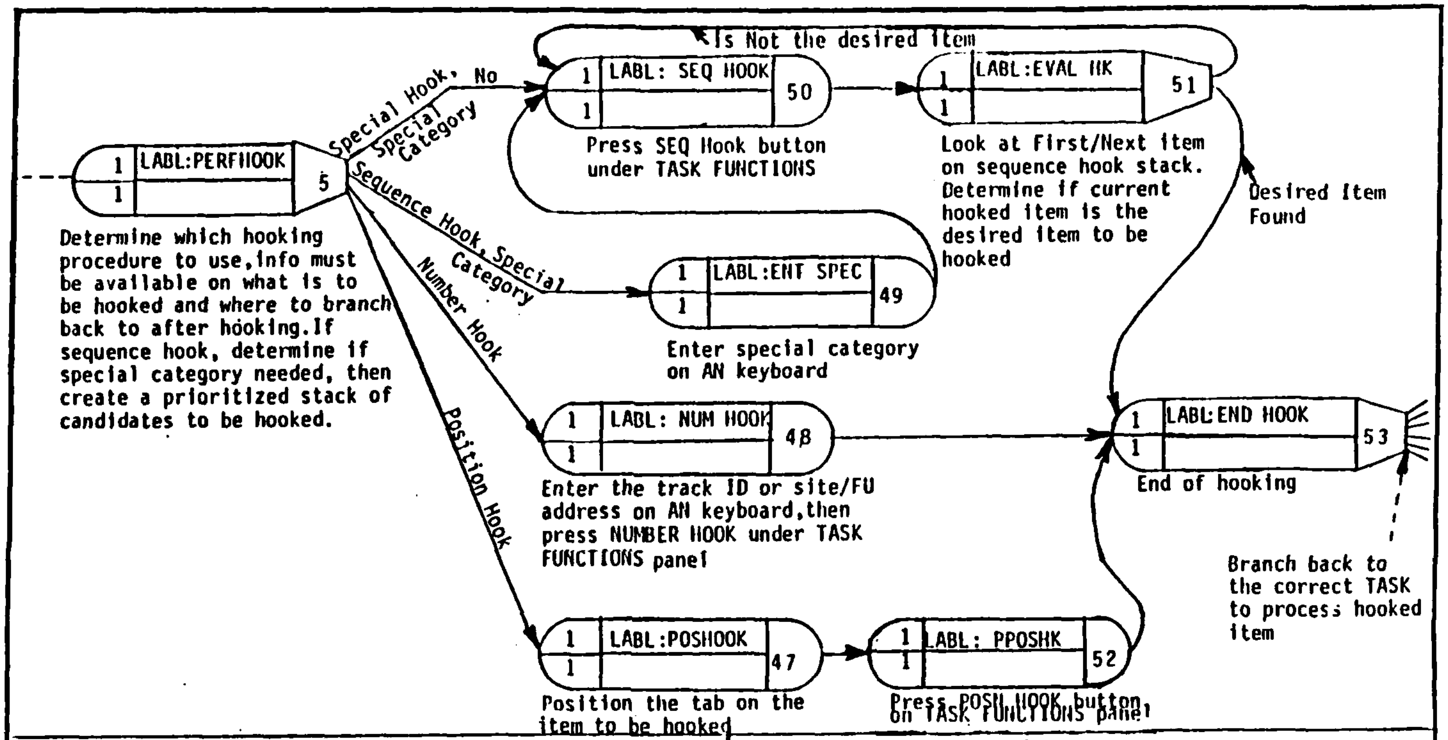

48
48
START
NUMBER HOOK
YES
NO
ENTER TRACK NUMBER, FIRE UNIT, OR SITE ADDRESS ON A/N KEYBOARD (ITEM 5)
A/N KEYBOARD ENTRY APPEARS IN ARO ROW 8 COLUMNS 57-64
PRESS TASK FUNCTIONS NUMBER HOOK (ITEM 4)
FORMAT:
$A_1A_2$NNN, $A_3$ $A_4A_4$NNN OR NNNN (TRACK)
$A_1A_2$NNN OR $A_5N_1N_1$ (FU); $A_1A_2$ OR NNN (SITES)
$A_1A_2$ NNN (ATDL-1 TRACK OR ATDL-1 OR MBDL FU)
$A_1A_2$ (ATDL-1 SITE)
$A_1$ - A THRU N, P OR Q
$A_2$ - A THRU H
N - 0 THRU 7
$A_3$ $A_4$ $A_4$NNN (NATO TRACK)
$A_3$ - N
$A_4$ - A, E, G, H, J THRU M
N - 0 THRU 7
NNNN (TADIL-B TRACK)
NNN (TADIL-B SITE)
N - 0 THRU 7
$A_5N_1N_1$ (FU DESIGNATOR USED TO ASSIGN DDG ROW NUMBER)
$A_5$ - A THRU Z
N - 0 THRU 9
GEOREF POSITION
YES
NO
PRESS TASK SELECTIONS TRACK DATA (ITEM 2)
ENTER GEOREF COORDINATES OF SYMBOL ON A/N KEYBOARD
A/N KEYBOARD ENTRY APPEARS IN ARO ROW 8 COLUMNS 57-64
PRESS TASK FUNCTIONS POSN ENTRY (ITEM 3)
TAB MOVES TO ENTERED COORDINATES ON PPI
POSITION TAB ON SYMBOL TO BE HOOKED
PRESS TASK FUNCTIONS POSN HOOK (ITEM 1)
HOOK SYMBOL APPEARS AROUND DESIGNATED SYMBOL ARO DATA FOR HOOKED ITEM IS DISPLAYED
STOP
TASK SELECTIONS
TRACK DATA
TASK FUNCTIONS
POSN ENTRY
NUMBER HOOK
POSN HOOK
2
3
4
1
5
NOTES:
1. POSN HOOK AND NUMBER HOOK VALID IN ALL MODES EXCEPT TEST.
2. GEOREF HOOK VALID ONLY IN TRACKING OR TRACKING/TACTICAL MODE.
3. □ INDICATES EQUIPMENT MARKING.
MS 1960518

*Figure 4-14. Console Hooking Procedures – Number, GEOREF, and Position Hook*

Figure II-1. AN/TSQ-73 Console Hooking Procedures.


Is Not the desired Item
1 LABL:PERFHOOK 1 5
Determine which hooking procedure to use, info must be available on what is to be hooked and where to branch back to after hooking. If sequence hook, determine if special category needed, then create a prioritized stack of candidates to be hooked.
Special Hook, No Special Category
Sequence Hook, Special Category
Number Hook
Position Hook
1 LABL: SEQ HOOK 1 50
Press SEQ Hook button under TASK FUNCTIONS
1 LABL:EVAL HK 1 51
Look at First/Next item on sequence hook stack. Determine if current hooked item is the desired item to be hooked
Desired Item Found
1 LABL:ENT SPEC 1 49
Enter special category on AN keyboard
1 LABL: NUM HOOK 1 48
Enter the track ID or site/FU address on AN keyboard, then press NUMBER HOOK under TASK FUNCTIONS panel
1 LABL:END HOOK 1 53
End of hooking
Branch back to the correct TASK to process hooked item
1 LABL:POSHOOK 1 47
Position the tab on the item to be hooked
1 LABL: PPOSHK 1 52
Press POSH HOOK button on TASK FUNCTIONS panel

| TASK NAME AND DESCRIPTION | OPERATOR | REFERENCE(S) |
|---|---|---|
| 5/ Perform Hooking Procedure | TD or TDA | TM 10-3, Figure 4-13, 4-14 |
| NAME: JACK WALKER<br>DATE: 7/22/82 | AIR DEFENSE SYSTEM MODULE: AN/TSQ-73<br>PROJECT: MOPADS | |
| AGGREGATE SAINT TASK MODELS | | PAGE 1 OF 1 |

Figure II-2. Aggregate SAINT Task Model of Console Hooking.

Sample data for this task element are shown below:

| | | |
|---|---|---|
| Nominal Mean Time: | 4.5 seconds | |
| Skills Required: | Fine Manipulation | 75% |
| | Short-Term Memory-Symbolic | 15% |
| | Detection | 10% |

During a MOPADS simulation, when an operator is to perform this task element, MOPADS will call the moderator functions for the three skills shown above. The moderator functions have access to the information in the operator state vector, the environmental state vector, and the task data. With this information, the moderator functions will each compute a change to the mean of 4.5 seconds.

Let $D_1$, $D_2$, and $D_3$ be the computed changes from the moderator functions for fine manipulation, short-term memory-symbolic, and detection, respectively. See Laughery & Ditzian (1982) & Laughery (1981) for these skill moderator functions. The combined mean is determined from

$$\mu_{moderated} = 4.5 + (.75D_1 + .15D_2 + .10D_3)$$

The obvious interpretation is that approximately 75% of the time performing the task element is in fine manipulation, etc., and minimal interaction between skills occurs. In the absence of data, these are relatively mild assumptions to achieve computable moderators for a wide variety of activities.

The moderated mean may then be used in the simulation to determine the time required for the operator to perform the task element. MOPADS allows several options in this regard:

1. Deterministic, unmoderated - always use the nominal mean time

2. Deterministic, moderated - use $\mu_{moderated}$ for the task element time

3. Stochastic, unmoderated - make a random draw from the specified distribution function whose mean is the nominal mean. Do not use the moderator functions.

4. Stochastic, moderated - make a random draw from the specified distribution function whose mean is $\mu_{moderated}$

MOPADS supports the following distribution functions:

Constant

Normal

Uniform

Erlang -1 (Exponential)

Lognormal

Beta

Gamma

The Gamma distribution is the default distribution, but the user can select any of the above.

### 2-3. Software Implementation.

As stated earlier, the human factors moderator function module has been developed in a way that allows it to be separated from the rest of the MOPADS software and used independently. This has been accomplished through the following design considerations:

1. Every moderator function subprogram has an identical calling sequence.

2. Moderator functions request data from the operator state vectors, environmental state vectors, and the task data through standard subprogram calls.

The implication of the above are that non-MOPADS software environments can use the moderator functions by calling them in their standard way and by providing utility programs that the moderator functions will call to access operator, environmental, and task data. MOPADS documents Laughery & Ditzian (1982) & Laughery (1981) contain the technical details.

## 3-0 TASK SEQUENCING METHODOLOGY

### 3-1. Task Sequencing Considerations.

In order to adequately model the actions of an air defense operator, the simulation must "perform" operator tasks in a way that responds to the air defense scenario. In other words, the simulated operators must perform tasks that tend toward accomplishing the mission of the air defense system. Developing a methodology that will exhibit this behavior is more difficult than simulating the operator tasks, because the simulation must represent the knowledge, experience, and motivations of the operators.

Furthermore, the simulation methodology needs to be accessible to the user. In other words, input parameters for the task sequencing algorithm should be intuitively meaningful and reasonably easy to determine. With these ideas in mind, the objectives in developing a task sequencing procedure were as follows:

The procedure should be:

1. consistent with the literature,

2. intuitively meaningful so that a MOPADS analyst can specify operator oriented parameters rather than abstract parameters that are obtained from some curve fitting procedure, and

3. relatively goal independent. This means that the parameters associated with one operator objective are nearly independent of the parameters associated with all other objectives.

Utility function and goal seeking approaches were considered. The multi-attribute utility function procedure was discarded because it did not satisfy objectives (2) and (3) above. A goal seeking approach, on the other hand, can be implemented in a way that satisfies (2) and (3) above and is consistent with the Newell & Simon (1972) view of humans as goal seekers. The principal advantage of the goal seeking approach is that the utility or "goal priority" of each goal can be computed independently of all other goals (this of course, is an assumption but a relatively mild one).

In particular, goal priority functions can be defined that are ordinal in nature, so that changes in parameters for one goal do not affect the parameters of other goals. The priority, or degree of goal satisfaction, for each goal can then be compared. This is in contrast to a multi-attribute utility approach in which a cardinal utility value is computed as a function of all goal states. The usual approach in utility theory is to develop a utility function of several variables (the goal states) from a curve fitting procedure. This is a cumbersome method for use in MOPADS because changes to individual goals for individual operators cannot be easily incorporated into the model.

## 3-2. The Task Sequencing Procedure.

The goal seeking behavior of the operators is characterized by the following:

1. The operators have a set of goals which they desire to satisfy simultaneously. They are capable of determining the value or state of each goal. For example, the operator may desire to maximize the distance from a critical asset to any hostile aircraft. The goal state is the minimum distance to any hostile track.

2. The operators can rank the importance of their goal states. In other words, they can assign priorities to their goals based upon their current states. For example, an AN/TSQ-73 operator can determine whether an uncleared alert message or a hostile aircraft within 30 miles of a critical asset should be attended to next.

3. The operators are capable of estimating the changes that will occur in their goal states if a particular task is performed.

4. Operators are limited in their ability to satisfy their goals. They may not be able to consider all of their goals at once.

These concepts are implemented in the following ways. For each operator goal, the goal state (denoted GS) must be explicitly specified in a way that allows GS to be assigned a unique value (e.g., GS = the number of unassigned hostile tracks). Then a goal priority function, GP, is specified for the goal that assigns a non-negative value to each value of GS. Figure II-3 shows an hypothetical example. The goal is satisfied when the goal state, GS, is between m and M. This is signified by GP = 0 when $m \leq GP \leq M$. If the goal state is less than m or greater than M, then the priority of the goal (i.e., its degree of dissatisfaction) increases linearly.

The meaning of the particular goal priority function in Figure II-3 is that the operator is satisfied and indifferent to any value of the goal state between m and M. Downside deviations (i.e., values of GS less than m) are more important than upside deviations (i.e., values of GS greater than M) since the slope for downside deviations is greater than the slope for upside deviations.

Each goal that an operator has may have a different priority function. See Figure II-4 for examples. The values of the goal priorities for each goal are compared in an ordinal fashion during task sequencing to determine the most dissatisfied goal or goals. This scheme allows the parameters for a goal priority function to be specified or changed without affecting the priority functions for other goals. Of course, complete independence is not obtained because the modeler must always be aware of the priority functions of the rest of the goals.

Determination of the goals and the goal priority functions must be accomplished with close coordination with subject matter experts since no open literature is available. Recall that the operators will select tasks in order to improve their goal states. Therefore, the modeler must specify one or more goals whose states are affected by each task. Goal priority functions may be determined by giving pairwise comparisons to subject matter experts (e.g., given states for two goals, which goal would have the highest priority?). Since the scale of the goal priority is arbitrary, the modeler can ordinally rank the responses to achieve correct rankings of the goals. In the current MOPADS implementation, a goal priority of 10 has been used to imply an extreme emergency, so goal priority values generally fall in the range zero to 10.

The operators seek to achieve one of the following two objectives when selecting the next task.

1. Maximize the expected reduction in the average goal priority of the NG largest goal priorities.

2. Maximize the expected reduction per unit tim: of the average goal priority of the NG largest goal priorities.

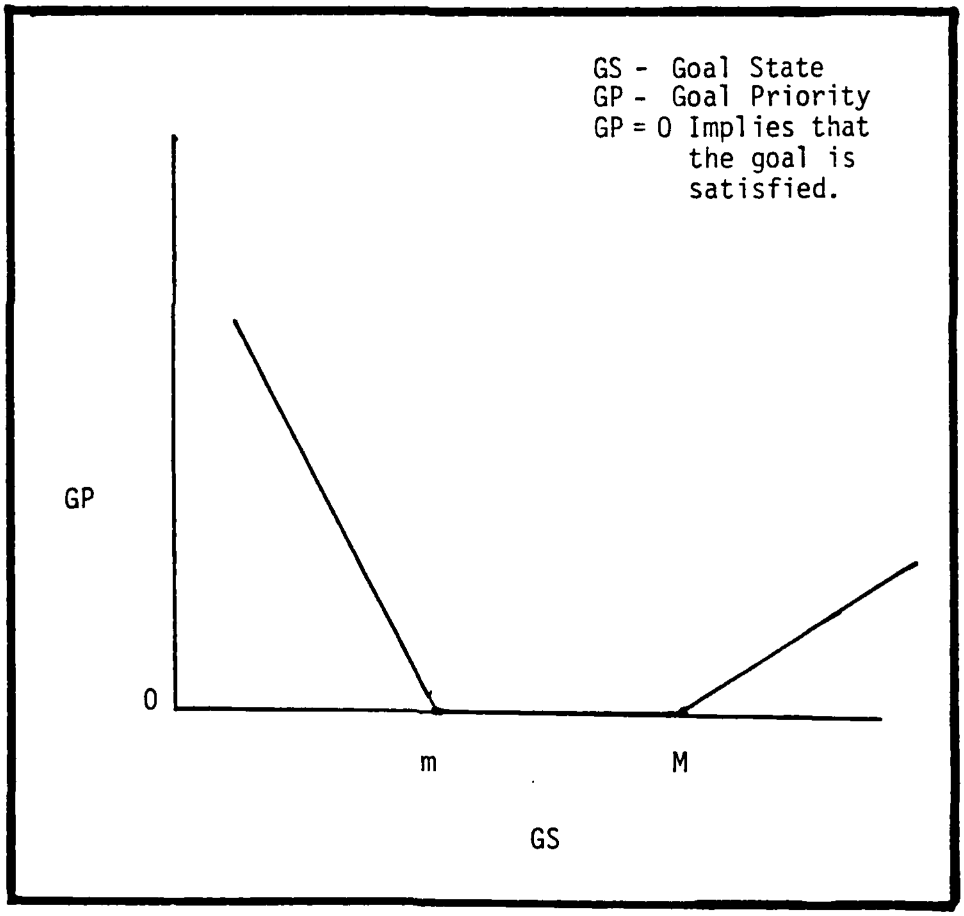


Figure II-3. Example Goal Priority Function.

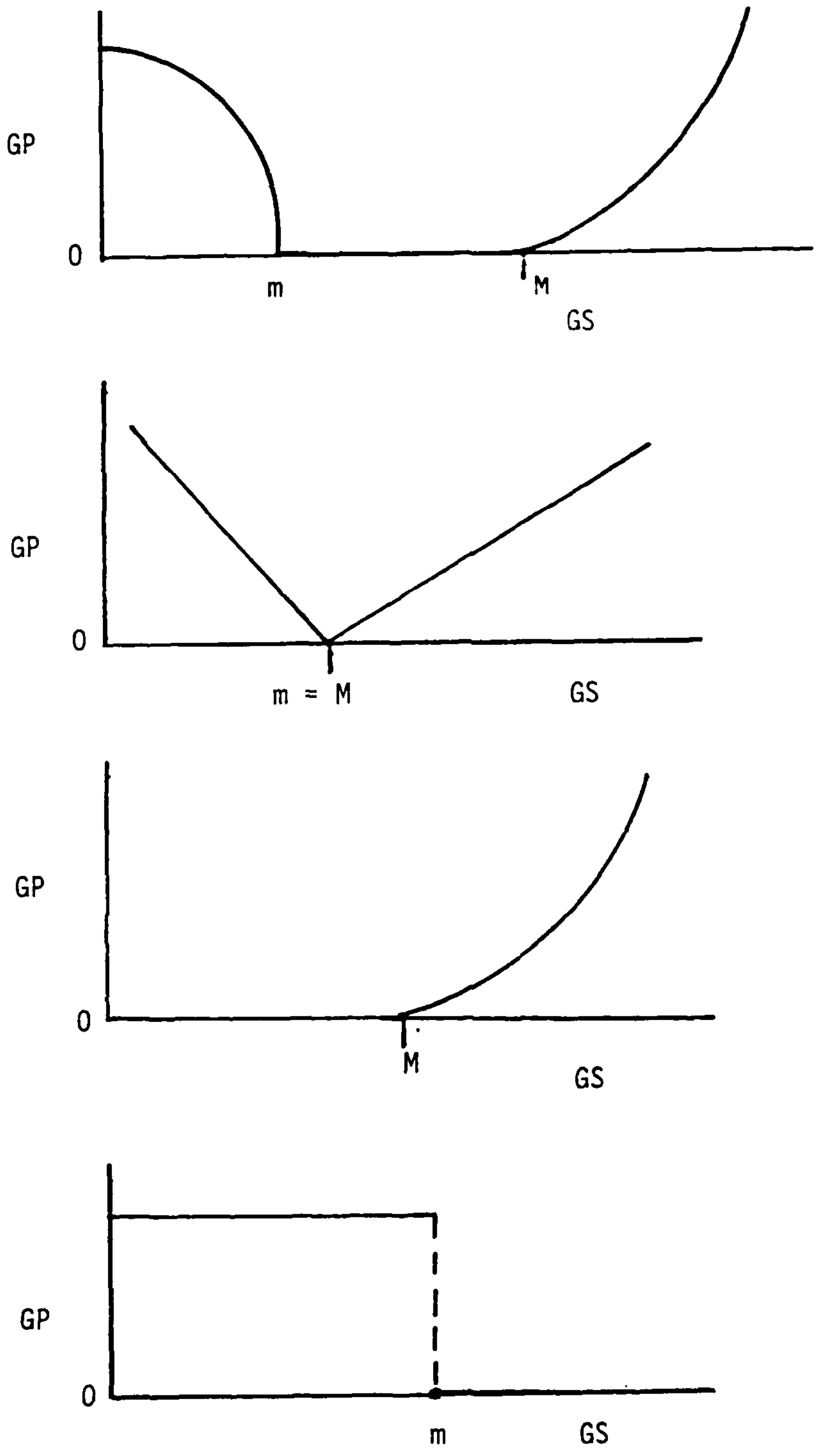


Figure II-4. Example Goal Priority Function Forms.

In the first case, the operator computes the average goal priority of the NG most dissatisfied goals. NG is a parameter of the operator which may be set by the MOPADS user. Note that if NG is one, the operator "puts out the biggest fire first." As NG increases, the operator is modeled as being able to take a more and more global view of his tasks. At the present time, NG remains fixed for each operator during the entire simulation. If relevant data were available, it would be possible to dynamically vary NG in response to operator work load or other conditions during the simulation.

The second objective function is similar to the first, but, in addition, the operator estimates the time it will take to perform each option available. With this information, the operator estimates the change in average goal priority that will occur per unit time and selects the one which gives the most rapid improvement.

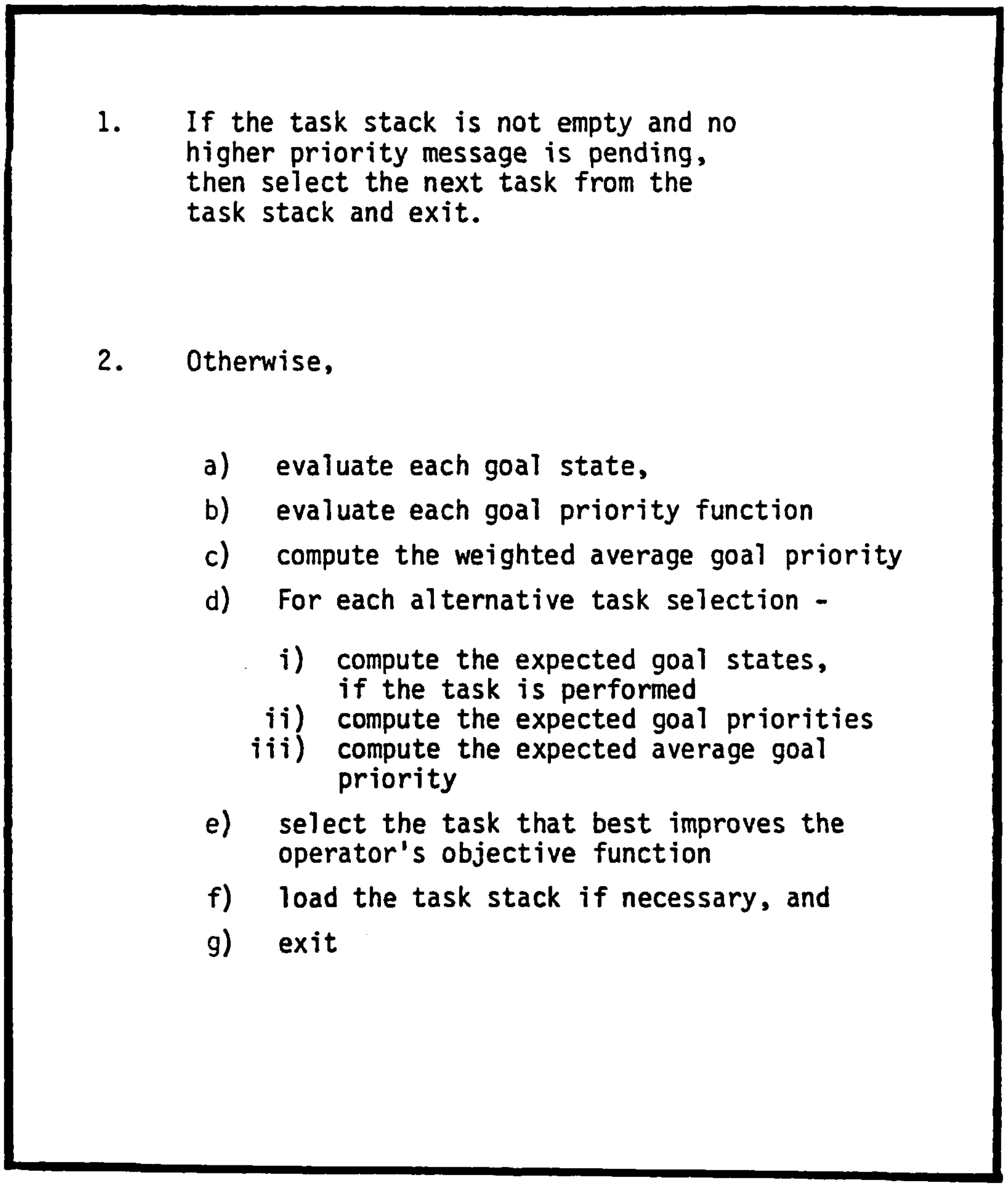
Finally, a last-in-first-out task stack is maintained for each operator. Some options available to the operator may involve performing several operator tasks before re-evaluating goals again. For such a case, the tasks which will be performed in sequence are loaded in the operator's task stack. When one task is completed, the operator will immediately perform the next task on his stack. Goal evaluation is performed only when the stack is empty or a high priority alert message has been received. In the case of a high priority message, a complete goal evaluation occurs.

The procedure for goal evaluation and task selection is shown in Figure II-5. The special weighted average of the goal priorities is computed as follows (here assume that the goal priorities have been arranged in decreasing order).

$$AVNG = \sum_{i}^{NG} \left( w_i \ GP_i \right)$$

where $$w_i = \frac{GP_i}{\sum_{i}^{NG} GP_i}$$

Thus, the maximum weight is given to the most dissatisfied goal (i.e., the one with the largest goal priority). This method prevents certain distortions in task selection that might result from the use of a simple average. For example, suppose the two



1. If the task stack is not empty and no higher priority message is pending, then select the next task from the task stack and exit.

2. Otherwise,

   a) evaluate each goal state,

   b) evaluate each goal priority function

   c) compute the weighted average goal priority

   d) For each alternative task selection -

      i) compute the expected goal states, if the task is performed

      ii) compute the expected goal priorities

      iii) compute the expected average goal priority

   e) select the task that best improves the operator's objective function

   f) load the task stack if necessary, and

   g) exit

Figure II-5. Goal Evaluation Procedure.

most dissatisfied goals (with $\underline{N}G = 2$) are $(GP_1, GP_2) = (10,5)$. The simple average goal priority, $\overline{X}$, is 7.5 while AVNG = 8.33. ($w_1 = .67$, $w_2 = .33$). Suppose there are two options whose expected results are shown below (the $w_i$ are not recomputed).

| | $GP_1$ | $GP_2$ | $\overline{X}$ | AVNG |
|---|---|---|---|---|
| Option 1 | 10 | 1 | 5.5 | 7.0 |
| Option 2 | 6 | 5 | 5.5 | 5.67 |

The simple average of the goal priorities is indifferent to the two options since they both result in a reduction of four in the sum of the goal priorities. Option 1, however, makes no improvement in the most dissatisfied goal. Selection of option 1 would represent an operator who attempts to improve the second most dissatisfied goal when there is an available option that improves the most dissatisfied goal. Using the special weighted average AVNG ensures that the operator will always pay most attention to the most dissatisfied goals even though he is attempting to consider more than one goal simultaneously.

Finally, note that the goal seeking procedure explained above is a simplified special case of utility theory. The utility function is simply a composition of the univariate goal priority functions which are combined through the AVNG function to a single value for each set of goal states. It would be a simple matter to recast the mathematical expression of the procedure in a utility theoretic framework. The goal seeking method simply assumes that a utility surrogate(the goal priority) can be expressed as a function of a subset of the set of the goal states (i.e., the priority of a goal is a function only of its own goal state).

### 3-3. Operator Goals for the AN/TSQ-73 and IHAWK.

The goals identified for the operators of the AN/TSQ-73 and the IHAWK arise from the basic goals of self preservation and a desire to accomplish their mission. The translation of these basic goals to operative goal statements results in goals that lead the operators to attack aircraft that present threats to themselves and the sites they are assigned to protect and to follow standard procedures to accomplish their missions. The statements of the goals for the AN/TSQ-73 and IHAWK operators are shown in Tables II-3 and II-4, respectively.

### 3-4. Software Implementation of Task Sequencing.

As is the case in all of the MOPADS software designs, the structure is intended to allow for future expansions (see Figure II-6). The common programs include those parts of the task sequencing

Table II-3. Operator Goal for the AN/TSQ-73.

| GOAL NUMBER | DESCRIPTION | EVALUATION OF GOAL STATE |
|---|---|---|
| 1 | Self Defense- maximize the minimum time to arrive of any threatening track. | The minimum time for any unassigned, not-receding, hostile or unknown track to arrive if it immediately turned inbound. |
| 2 | Protect Critical Assets - maximize the minimum time to arrive at a critical asset of any threatening track. | The minimum time for any unassigned, not-receding, hostile or unknown track to arrive at any protected site if it immediately turned inbound to the site. |
| 3 | Attack Threatening Tracks | The number of unassigned, visible, hostile or unknown, not-receding tracks and the number of covered tracks. |
| 4 | Minimize the number of unidentified tracks | The number of unidentified tracks visible or visible to owned fire units. |
| 5 | Minimize the number of uninitiated video contacts | The number of visible video contacts |
| 6 | Conserve ammunition | The number of tracks being engaged by more than one fire unit |

| | | | |
|---|---|---|---|
| NAME: | Joseph Polito | AIR DEFENSE SYSTEM MODULE: | AN/TSQ-73 |
| DATE: | 8/25/83 | PROJECT: | MOPADS |

OPERATOR GOALS DEFINITIONS Page 1 of 2

Table II-3 (continued)

| GOAL NUMBER | DESCRIPTION | EVALUATION OF GOAL STATE |
|---|---|---|
| 7 | Respond to Communications | The maximum priority of any outstanding message that the operator may receive and that no one else is processing. |
| 8 | Protect Friendly Tracks | The number of friendly tracks engaged or assigned. |

NAME: Joseph Polito
DATE: 8/25/83

AIR DEFENSE SYSTEM MODULE: AN/TSQ-73
PROJECT: MOPADS

OPERATOR GOALS DEFINITIONS

Page 2 of 2

Table II-4. Operator Goal for the IHAWK.

| GOAL NUMBER | DESCRIPTION | EVALUATION OF GOAL STATE |
|---|---|---|
| 1 | Engage Assigned Tracks | The minimum time for any not receding, in-range, assigned, but not engaged Track to arrive at the IHAWK or its protected site if it immediately turned inbound. |
| 2 | Self Defense | The minimum time for any hostile or unknown, not receding, in-range, unassigned Track to arrive at the IHAWK if it immediately turned inbound. |
| 3 | Protect Critical Assets | The minimum time for any hostile or unknown, *not receding*, *in-range*, unassigned Track to arrive at any protected site if it immediately turned inbound. |
| 4 | Minimize the number of unidentified Tracks. | The number of unidentified tracks. |
| 5 | Minimize the *maximum* priority of outstanding messages | The maximum priority of any outstanding message that the operator may receive and no one else is processing. |

NAME: Joseph Polito
DATE: 8/25/83

AIR DEFENSE SYSTEM MODULE: IHAWK
PROJECT: MOPADS

OPERATOR GOALS DEFINITIONS

Page 1 of 2

Table II-4 (continued)

| GOAL NUMBER | DESCRIPTION | EVALUATION OF GOAL STATE |
|---|---|---|
| 6 | Maximize the number of missiles available | The total number of hot and cold missiles available. |
| 7 | Protect friends | The number of friendly tracks being engaged. |
| 8 | Minimize the number of unidentified, low altitude tracks. | The number of unidentified, low altitude tracks. |

| | | | |
|---|---|---|---|
| NAME: | Joseph Polito | AIR DEFENSE SYSTEM MODULE: | IHAWK |
| DATE: | 8/25/83 | PROJECT: | MOPADS |

OPERATOR GOALS DEFINITIONS Page 2 of 2

procedure that are independent of the system module and operator type. This includes statistics collection, branching, and certain utility programs. Each system module (designated SM1, SM2, etc. in Figure II-6) has its own entry point subprogram which will, in turn, call a program to process the goals for each operator type in the system. The task sequencing programs labeled SM1, SM2, etc. and their descendents are documented as part of the system module documentation. In this way, new system modules can be added by "plugging in" the task sequencing programs without disturbing existing system modules.

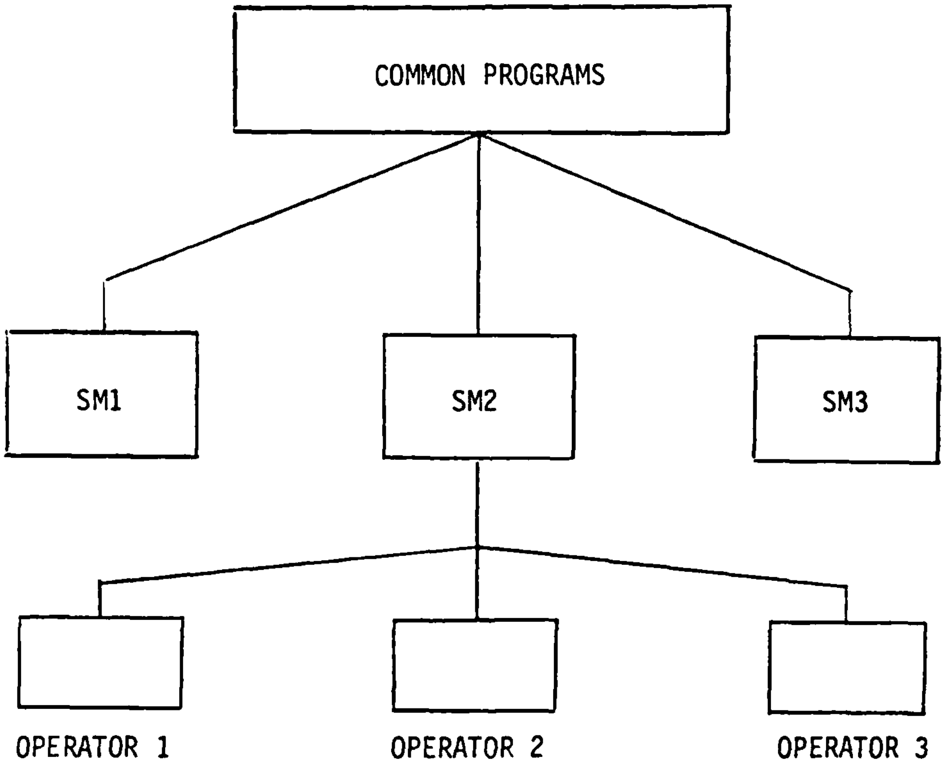


Figure II-6. Schematic Structure of MOPADS Task Sequencing Software.

# III. MOPADS AIR DEFENSE MODELS

## 1-0 DEVELOPMENT METHODOLOGY FOR AIR DEFENSE SYSTEM MODULES

MOPADS is intended to be a long lived software system that will evclve as new system modules are developed. Since the development of system modules is the most complex maintenance activity that will be performed on MOPADS and since it is likely that many individuals will be involved in developing the modules, it is essential that a systematic development and documentation methodology be followed. Procedures for these activities have been created (Walker & Polito, 1982a,b).

The procedure for developing system modules is summarized in Figure III-1. Steps 1, 2, and 3 are data collection functions in which the MOPADS analyst and those who aid him/her collect information and systematically characterize operator and equipment modeling requirements. In steps 4, 5, and 6, task sequencing considerations and constraints are identified and formalized. Human factors are minimally or nominally represented at this stage. The purpose is to ensure that infeasible or unrealistic sequencing does not occur.

At step 11, the MOPADS modeler enters data for the new system module into the MOPADS data base. The MOPADS user interface has facilities to support this activity. At step 12, the MOPADS simulations are performed with a minimal set of existing MOPADS models to test the new system module. When this is completed, the new system model is integrated with the full set of existing MOPADS models and tested. These are steps 13 and 14.

Step 15 is the final documentation effort. A systematic procedure for documentation has been specified to aid in module development and to ensure that adequate documentation of system modules is maintained. This is crucial because it is certain that when a new module is developed, the analyst will have to refer to the documentation of other system modules. This will be necessary or desirable in steps 4, 9, 10, 12, and 13.

Each document describing a system model will be organized in the same way. Figure III-2 is a typical table of contents. Standard forms have been provided to aid in modeling which will be included in Section III of the documents. Examples of these forms are shown in Sections 2-0 and 3-0 below.

Strict adherence to these documentation standards will result in a maintainable analysis tool with a long useful life.

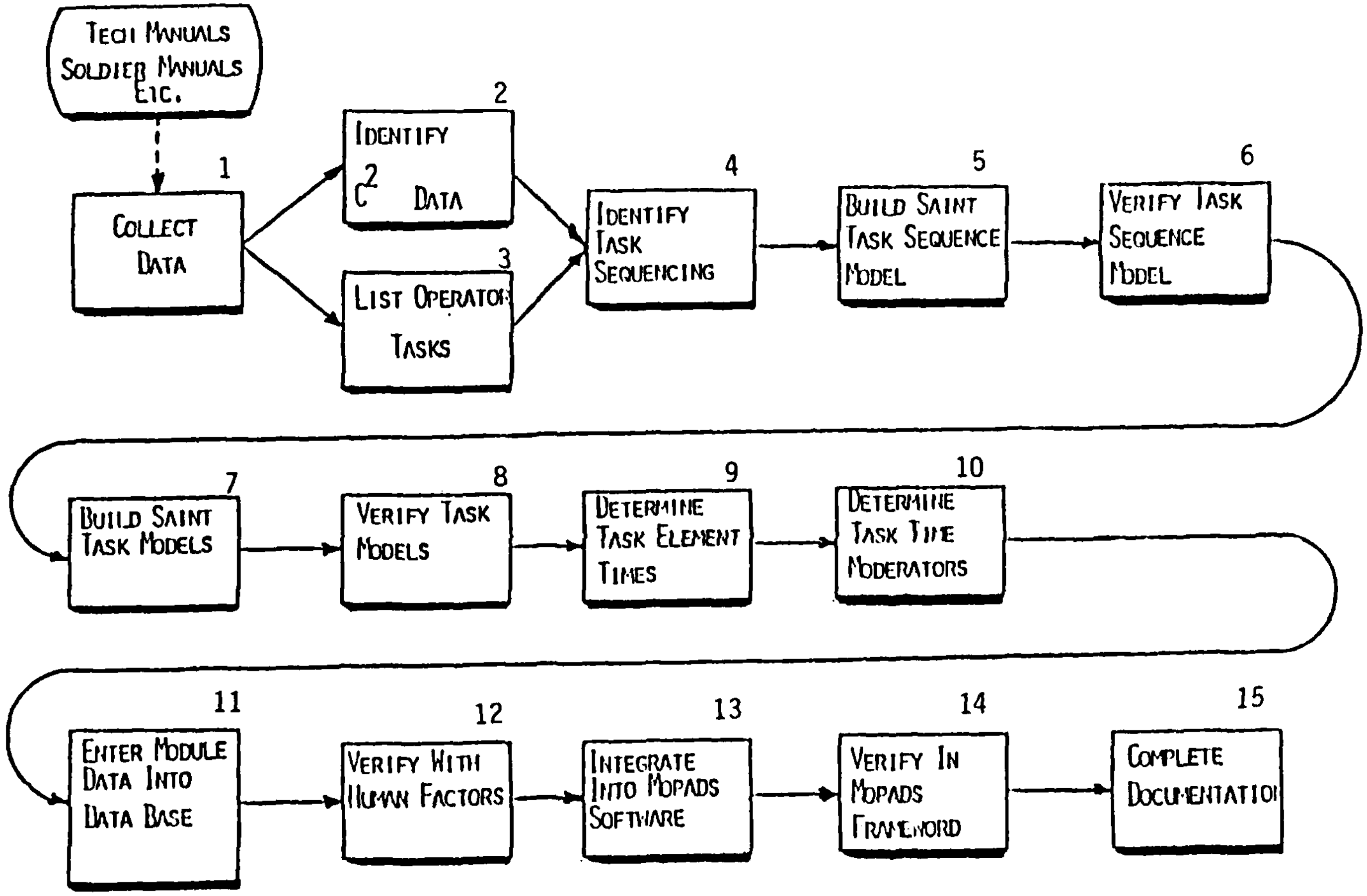

TECH MANUALS
SOLDIER MANUALS
ETC.
1
COLLECT DATA
2
IDENTIFY C2 DATA
3
LIST OPERATOR TASKS
4
IDENTIFY TASK SEQUENCING
5
BUILD SAINT TASK SEQUENCE MODEL
6
VERIFY TASK SEQUENCE MODEL
7
BUILD SAINT TASK MODELS
8
VERIFY TASK MODELS
9
DETERMINE TASK ELEMENT TIMES
10
DETERMINE TASK TIME MODERATORS
11
ENTER MODULE DATA INTO DATA BASE
12
VERIFY WITH HUMAN FACTORS
13
INTEGRATE INTO MOPADS SOFTWARE
14
VERIFY IN MOPADS FRAMEWORD
15
COMPLETE DOCUMENTATIO

Figure III-1. Development of System Modules.

I. SYSTEM DESCRIPTON

II. OVERVIEW OF THE SAINT MODEL

III. MODEL DESCRIPTION FORMS
1-0 Entities
2-0 Resources
3-0 Variables
4-0 Monitors
5-0 Task Descriptions
6-0 Statistics
7-0 User Functions
8-0 Moderator Functions

IV. USER WRITTEN SUBPROGRAMS
1-0 Index
2-0 Subprogram Descriptions
3-0 Subprogram Listings

V. LISTING OF SAINT NETWORK DATA INPUT

VI. NON-SAINT DATA REQUIREMENTS
1-0 Data Requirements
2-0 Data Source and Description

VII. OUTPUT REPORTS
1-0 Output Options
2-0 Sample Output

Figure III-2. Table of Contents for MOPADS System Module Documentation.

## 2-0 THE AN/TSQ-73 SYSTEM MODULE

### 2-1. AN/TSQ-73 Operator Tasks.

Table III-1 contains the operator tasks that are represented in the AN/TSQ-73 system module. An important part of developing MOPADS operator models is to determine which operator tasks are required to be represented. Many tasks are irrelevant to the air battle scenario. These include, for example, tasks related to the set-up, take-down, and transportation of the equipment. Furthermore, there may be duplication of task information among the task descriptions in official system documentation. Also, occasionaly it may be necessary to add a task to account for standard or common procedures that are not explicitly discussed in the official documentation.

Development of the task lists is a substantial activity that requires the MOPADS modeler to become familiar with all of the tasks in the official documentation so that informed decisions can be made in selecting those that must be modeled. Official documentation for the AN/TSQ-73 (U. S. Army Technical Manual, TM9-1430-652-10-3), Change 6, 1981) contains flow charts for the operator tasks. An example for hooking is shown in Figure III-3. These or similar representations must be obtained from the pertinent Army documentation and a preliminary task list determined. This activity is step 3 of Figure III-1. Since MOPADS involves models of combat operations, the operator tasks describing set-up and routine maintenance of equipment can usually be excluded at this point from further consideration.

For systems that have more than one operator, it may be necessary to develop task lists for each operator. An example of this case will be seen in Section 3-0 for the IHAWK system module. For the AN/TSQ-73, all tasks can be performed from each console, although it is customary for the operators to perform separate tasks based on their authority. For the AN/TSQ-73, this task split is represented structurally in the models rather than by developing separate task lists for each operator.

An important objective in developing the task list is to select the minimum set necessary to represent the operator interaction with the system to the required detail. Considerable effort is expended in expanding each operator task into an MSAINT representation. Therefore, the task list must be examined with a critical eye to ensure that unnecessary effort is not expended. As an example, consider the AN/TSQ-73. The usual operator configuration is to have a Tactical Director (TD) who has authority to order engagements and a Tactical

Table III-1. AN/TSQ-73 Operator Tasks Represented in MOPADS.

| DESCRIPTION |
|---|
| Idle Time (Scan the Displays) |
| Cancel Secondary Assignment |
| Send Terminate Commands |
| Clear Hold Fire, Effective, Status |
| Perform Hooking Procedure |
| Enter ID and IFF Data |
| Interrogate a Target or Sector |
| Send Command Message |
| Assign Weapons/Battalions |
| Receive Commands |
| Clear Alerts |
| Receive Miscellaneous Messages |

Director Assistant (TDA) who does not have such authority. The TDA will perform tracking and identification tasks, and the TD will order and monitor engagements. It is possible, however, to configurate the system with two operators that each have engagement authority (two "TD's"). In this case, each operator can perform all tasks. The modeler must decide which configuration(s) will be included in the model and specify task lists accordingly.

### 2-2. AN/TSQ-73 Messages.

Step 2 in Figure III-1 is to "Identify $C^2$ Data." Essentially, this step involves determining how operators of this equipment communicate with superior, subordinate, and lateral units. The communications occur through voice and data link messages.

MOPADS models of air defense systems (components) mimic this structure by communicating with messages sent through the MOPADS data base. It is necessary for the MOPADS modeler to explicity identify all communications between the air defense system being modeled and all other systems already modeled. For the current implementation, this means that the communication between the group and battalion AN/TSQ-73 and the IHAWK battery must be identified. Once again, a judicious elimination of messages that are not needed in the MOPADS context will greatly reduce later work.

The messages used in the current implementation are shown in Table III-2. For each such message, the sender, receiver, message characteristics, and message contents must be specified. A form has been developed to facilitate specification and documentation of this data. An example is shown in Figure III-4.

### 2-3. AN/TSQ-73 Operator Goals.

Once all of the messages and tasks for the operator have been determined, initial development of the task sequencing procedures for the operators can begin. The goals for the AN/TSQ-73 operators have been shown already in Table II-3.

The particular parameters used for each goal priority function are specified in Goodin & Polito (1983b).

### 2-4. AN/TSQ-73 Operator Task Models.

A SAINT task node model analogous to Figure II-2 has been developed for each of the operator tasks shown in Table III-1. This procedure involves defining the "entities" that will flow through the networks. In this case,the entities are operators. Each entity has a list of characteristics called attributes that identify it. Each of the operators is defined on a form as in Figure III-5. The attributes of the operators have been defined. These attributes are used in branching decisions to determine which Task Node will be performed next and in FORTRAN programs written by the MOPADS modeler to represent the operators' actions.

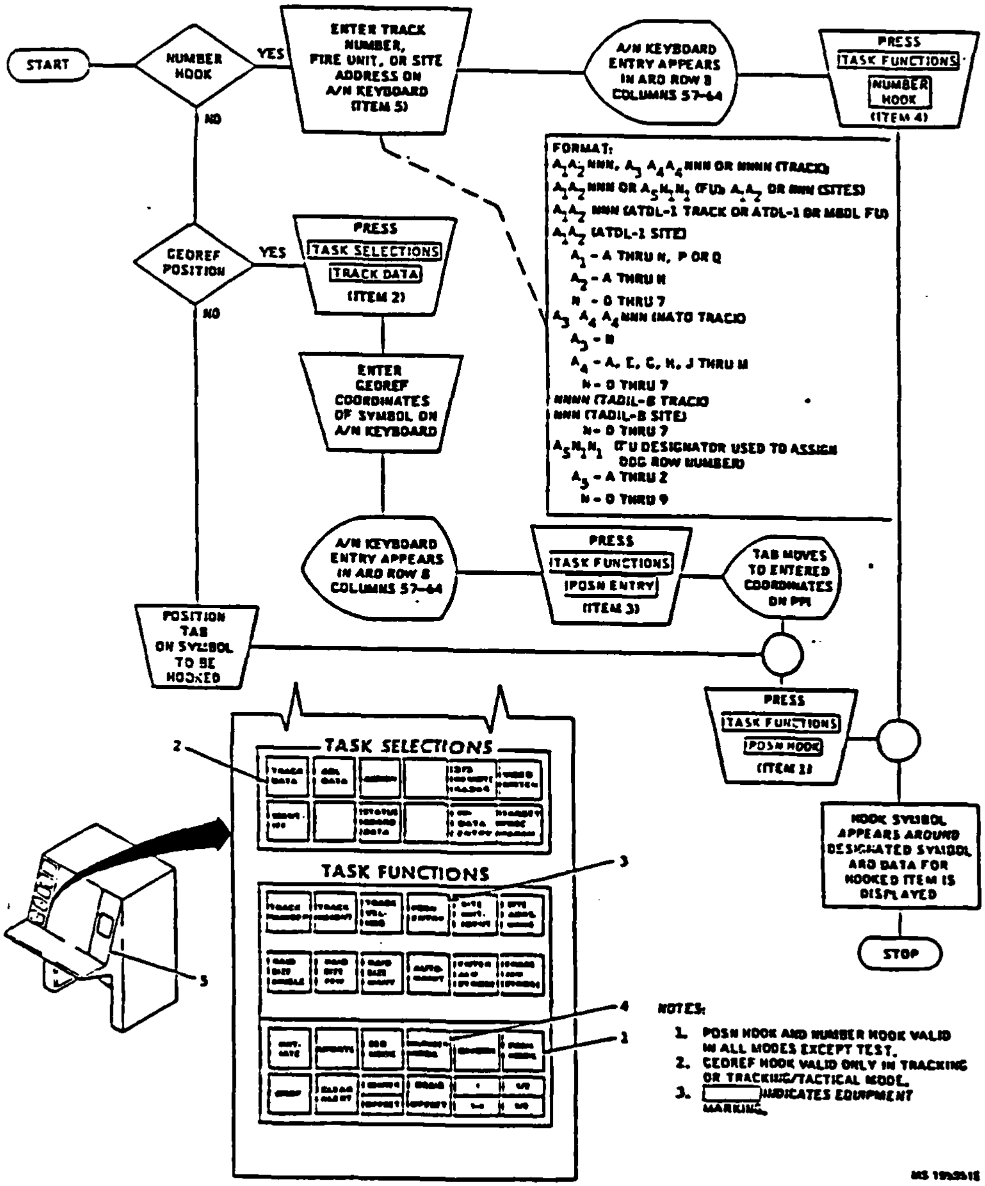

START
NUMBER HOOK
YES
NO
ENTER TRACK NUMBER, FIRE UNIT, OR SITE ADDRESS ON A/N KEYBOARD (ITEM 5)
A/N KEYBOARD ENTRY APPEARS IN ARD ROW 8 COLUMNS 57-64
PRESS TASK FUNCTIONS NUMBER HOOK (ITEM 4)
FORMAT:
$A_1A_2$NNN, $A_3$ $A_4A_4$NNN OR NNNN (TRACK)
$A_1A_2$NNN OR $A_5N_1N_1$ (FU) $A_1A_2$ OR NNN (SITES)
$A_1A_2$ NNN (ATDL-1 TRACK OR ATDL-1 OR MBDL FU)
$A_1A_2$ (ATDL-1 SITE)
$A_1$ - A THRU N, P OR Q
$A_2$ - A THRU H
N - 0 THRU 7
$A_3$ $A_4$ $A_4$NNN (NATO TRACK)
$A_3$ - B
$A_4$ - A, E, C, H, J THRU M
N - 0 THRU 7
NNNN (TADIL-B TRACK)
NNN (TADIL-B SITE)
N - 0 THRU 7
$A_5N_1N_1$ (FU DESIGNATOR USED TO ASSIGN DDG ROW NUMBER)
$A_5$ - A THRU Z
N - 0 THRU 9
GEOREF POSITION
YES
NO
PRESS TASK SELECTIONS TRACK DATA (ITEM 2)
ENTER GEOREF COORDINATES OF SYMBOL ON A/N KEYBOARD
A/N KEYBOARD ENTRY APPEARS IN ARD ROW 8 COLUMNS 57-64
PRESS TASK FUNCTIONS POSN ENTRY (ITEM 3)
TAB MOVES TO ENTERED COORDINATES ON PPI
POSITION TAB ON SYMBOL TO BE HOOKED
PRESS TASK FUNCTIONS POSN HOOK (ITEM 1)
HOOK SYMBOL APPEARS AROUND DESIGNATED SYMBOL AND DATA FOR HOOKED ITEM IS DISPLAYED
STOP
TASK SELECTIONS
TRACK DATA
TASK FUNCTIONS
1
2
3
4
5
NOTES:
1. POSN HOOK AND NUMBER HOOK VALID IN ALL MODES EXCEPT TEST.
2. GEOREF HOOK VALID ONLY IN TRACKING OR TRACKING/TACTICAL MODE.
3. □ INDICATES EQUIPMENT MARKING.
MS 195351S

Figure III-3. Example AN/TSQ-73 Operator Task Flow Chart.

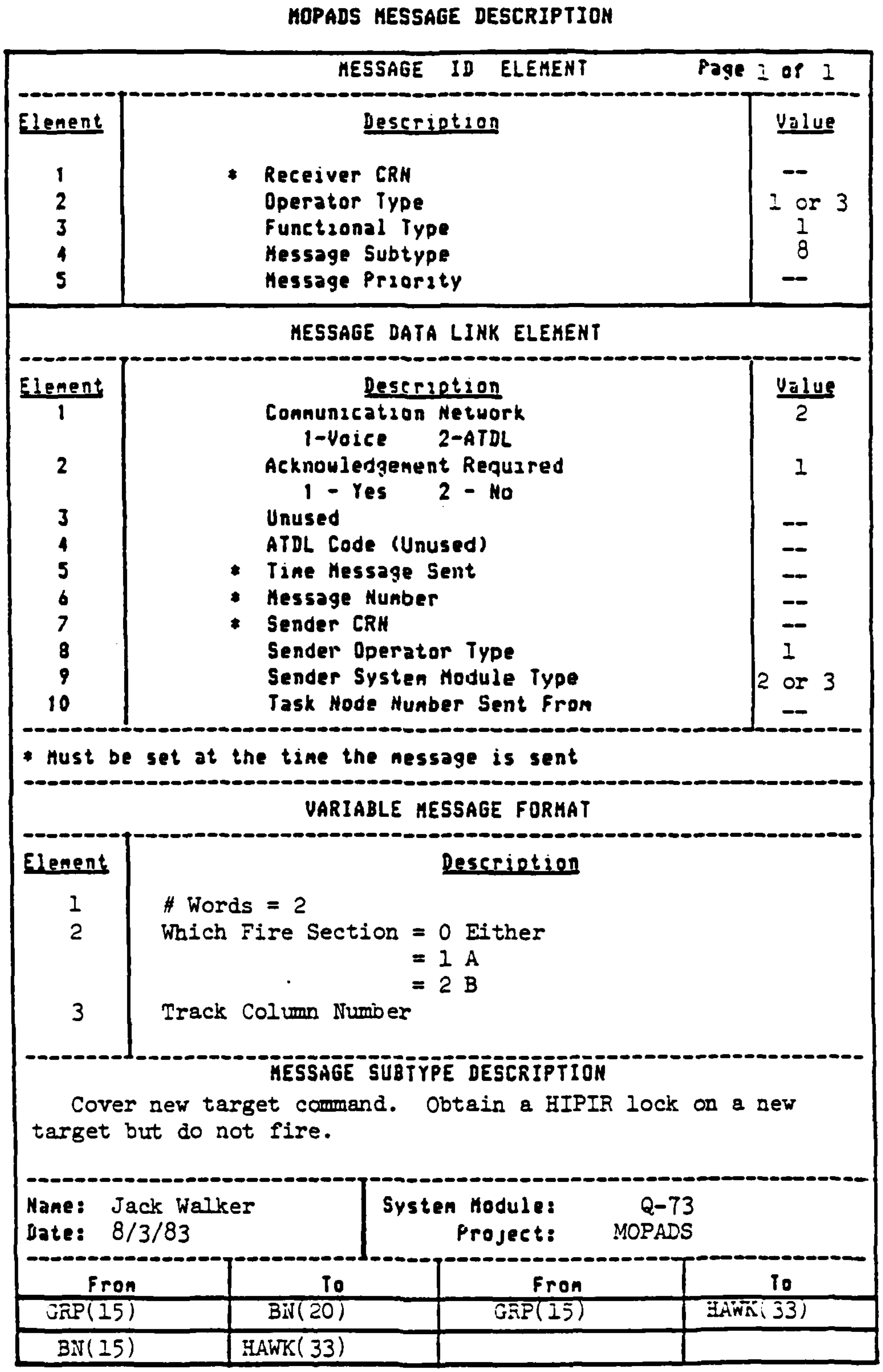
Table III-2. Messages for the AN/TSQ-73 and IHAWK.

| | Command Messages | |
|---|---|---|
| 1 | Hold Fire | |
| 2 | Cease Fire | |
| 3 | Cease Engagement | |
| 4 | Engage | On track already assigned |
| 5 | Cover | |
| 6 | Engage Ripple | |
| 7 | Engage New Track Assignment | |
| 8 | Cover New Track Assignment | |
| 9 | Engage Ripple Assignment | |
| 10 | Investigate/Assign Assignment | |
| 11 | Cancel Alert | |
| 12 | Track Assignment Status | |
| 13 | Change Targets | |
| 14 | Method of Fire | |
| 15 | Order No Kill | |
| 16 | Order Break Lock | |

| | Request for Information Messages |
|---|---|
| 1 | Request Cancel of TCC Alert |

| | Reporting Status Messages |
|---|---|
| 1 | FU Out of Action |
| 2 | FU Expended Hot Missiles |
| 3 | FU Effective/Not Effective |
| 4 | FU Engaging Pop-up Target |
| 5 | New ASO Target |
| 6 | IHIPIR Lock Status |
| 7 | Raid Size Report |
| 8 | Temporary No Kill |

| | Acknowledgement Messages | |
|---|---|---|
| 1 | Will Comply | Battalion Q-73 to Group |
| 2 | Have Complied | |
| 3 | Can't Comply | |
| 4 | Will Comply | IHAWK to Battalion Q-73 |
| 5 | Can't Comply | |
| 6 | Acknowledge ASO Target | |
| 7 | Accept ASO Target | |



MOPADS MESSAGE DESCRIPTION

MESSAGE ID ELEMENT — Page 1 of 1

| Element | Description | Value |
|---|---|---|
| 1 | * Receiver CRN | -- |
| 2 | Operator Type | 1 or 3 |
| 3 | Functional Type | 1 |
| 4 | Message Subtype | 8 |
| 5 | Message Priority | -- |

MESSAGE DATA LINK ELEMENT

| Element | Description | Value |
|---|---|---|
| 1 | Communication Network<br>1-Voice 2-ATDL | 2 |
| 2 | Acknowledgement Required<br>1 - Yes 2 - No | 1 |
| 3 | Unused | -- |
| 4 | ATDL Code (Unused) | -- |
| 5 | * Time Message Sent | -- |
| 6 | * Message Number | -- |
| 7 | * Sender CRN | -- |
| 8 | Sender Operator Type | 1 |
| 9 | Sender System Module Type | 2 or 3 |
| 10 | Task Node Number Sent From | -- |

* Must be set at the time the message is sent

VARIABLE MESSAGE FORMAT

| Element | Description |
|---|---|
| 1 | # Words = 2 |
| 2 | Which Fire Section = 0 Either<br>= 1 A<br>= 2 B |
| 3 | Track Column Number |

MESSAGE SUBTYPE DESCRIPTION

Cover new target command. Obtain a HIPIR lock on a new target but do not fire.

| | |
|---|---|
| Name: Jack Walker<br>Date: 8/3/83 | System Module: Q-73<br>Project: MOPADS |

| From | To | From | To |
|---|---|---|---|
| GRP(15) | BN(20) | GRP(15) | HAWK(33) |
| BN(15) | HAWK(33) | | |

Figure III-4. Example Message Sent From the AN/TSQ-73.

| NODE(S) WHERE CREATED | DESCRIPTION | INFORMATION ATTRIBUTES | | | RESOURCE REQUIREMENTS |
|---|---|---|---|---|---|
| | | ATTRIBUTE NUMBER | DEFINITION | INITIAL VALUE (IF APPROPRIATE) | |
| 31(BN)<br>31 (GRP)<br>32 | TD-Officer for handling all engagement and FU assignment tasks (operator types 1(BN), 8(GRP), and 9(GRP)) | 1 | Operator ID | - | |
| | | 2 | Copy Row Number | - | |
| | | 3 | Operator Type (1-TD, 2-TDA) | - | |
| | | 4 | Current Track Column Pointer (Ø if none) | - | |
| | | 5 | Internal Mark Time (For Op.Task Time Statistics) | 0.0 | |
| 32(BN) | TDA-Handles tracking and identification tasks (Operator type = 2) | 6 | Self-Clearing Indicator (Ø-No Self Clearing >Ø-Node to Clear to) | 0 | |
| | NOTE: Both TD's and TDA's have the same IA definitions. | 7 | Last Task Node Branched from (set and used only at release times) | - | |
| | | 8 | Hooking Indicator (Ø-None,Branch to hooking, >Ø-Task Node to branch back to) | 0 | |
| | | 9 | Item to be hooked (<Ø site or FU; =CRN; =Ø No hooking;>Ø Track column pointer) | | |

Name: Riley Goodin
Date: 11/15/83

AIR DEFENSE SYSTEM MODULE: AN/TSQ-73
PROJECT: MOPADS

SAINT ENTITIES
Page 1 of 2

Figure III-5. Example SAINT Operator Definition Form.

| NODE(S) WHERE CREATED | DESCRIPTION | INFORMATION ATTRIBUTES | | | RESOURCE REQUIREMENTS |
|---|---|---|---|---|---|
| | | ATTRIBUTE NUMBER | DEFINITION | INITIAL VALUE (IF APPROPRIATE) | |
| | | 10-12 | Unused currently | - | |
| | | 13-14 | Branching (Task Node Specific uses) | - | |
| | | 15 | Information passed in or out of task sequencing (operator task specific uses) | | |

Name: Riley Goodin
Date: 11/15/83

AIR DEFENSE SYSTEM MODULE: AN/TSQ-73
PROJECT: MOPADS

SAINT ENTITIES

Page 2 of 2

Figure III-5. (continued)

When operator characteristics are specified, the task networks like Figure II-2 are expanded to describe each Task Node. Figures III-6, 7, 8, and 9 are examples. On these forms are specified the skill and skill weight requirements (see Table II-1), the task related variable data (see Table II-2), and the mean, standard deviation and distribution type of the nominal performance time. Any resource requirements are also specified.

The MOPADS operator task number is also given, so a cross-reference is possible between SAINT task nodes and operator tasks. Given a task node number, it is possible to find the operator task that contains it by looking at the forms in Figures III-6 to 9. Conversely, all of the task nodes that make up an operator task model can be found in the forms shown in Figure II-2.

Specification and verification of all of the data in Figures III-5 through III-9 completes steps 7 to 10 in Figure III-1. This leaves step 11 which involves entering data interactively into the MOPADS data base to support the system module. SAINT task models have been developed for each operator task, and nominal task performance times and skill requirements have been determined for each task node.

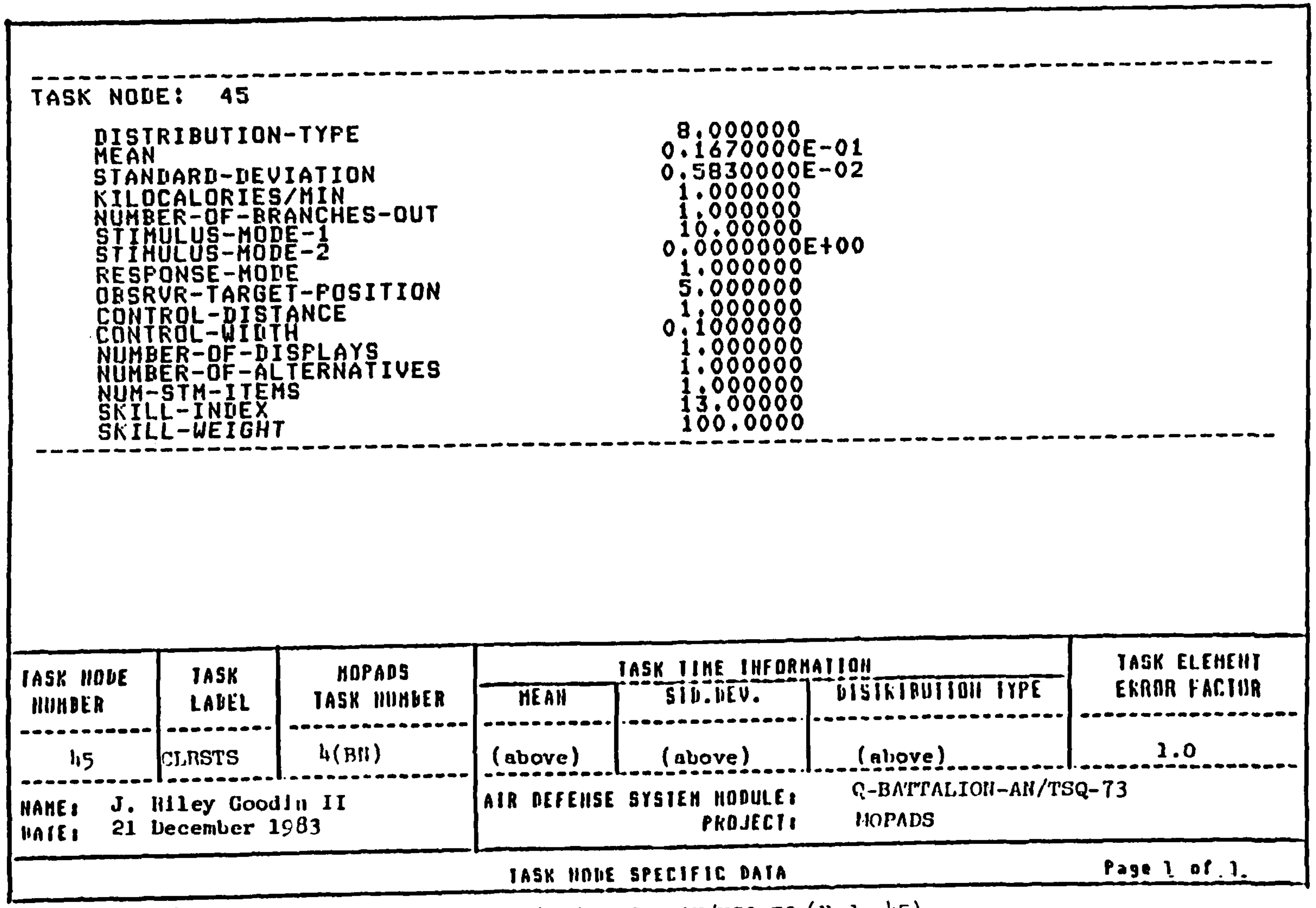
## 3-0 THE IHAWK SYSTEM MODULE

### 3-1. IHAWK Operator Tasks.

The IHAWK operator tasks that are represented in MOPADS are shown in Table III-3. A separate task list has been prepared for each IHAWK operator, because, in contrast to the AN/TSQ-73,the various operators perform substantially different functions. By far, the most complex activities are performed by the Tactical Control Officer (TCO). All communications to other air defense components are represented as passing to and from the TCO.

The other operators perform relatively straightforward tasks. In other words, the task sequencing for these operators is relatively simple. Their activities are primarily rule-based.

The process of developing the IHAWK task lists was analogous to that for the AN/TSQ-73. It was somewhat more difficult to develop the lists and network models, however, because the Army documentation (U. S. Army Technical Manual, TM9-1430-1526-12-1, June 30, 1979) does not contain task flow charts analogous to Figure III-3. The activities are described in text that needed to be examined and put into network form.



```
-------------------------------------------------------------------------------
TASK NODE:  45

    DISTRIBUTION-TYPE                        8.000000
    MEAN                                     0.1670000E-01
    STANDARD-DEVIATION                       0.5830000E-02
    KILOCALORIES/MIN                         1.000000
    NUMBER-OF-BRANCHES-OUT                   1.000000
    STIMULUS-MODE-1                          10.00000
    STIMULUS-MODE-2                          0.0000000E+00
    RESPONSE-MODE                            1.000000
    OBSRVR-TARGET-POSITION                   5.000000
    CONTROL-DISTANCE                         1.000000
    CONTROL-WIDTH                            0.1000000
    NUMBER-OF-DISPLAYS                       1.000000
    NUMBER-OF-ALTERNATIVES                   1.000000
    NUM-STM-ITEMS                            1.000000
    SKILL-INDEX                              13.00000
    SKILL-WEIGHT                             100.0000
-------------------------------------------------------------------------------
```

| TASK NODE NUMBER | TASK LABEL | MOPADS TASK NUMBER | TASK TIME INFORMATION MEAN | TASK TIME INFORMATION STD.DEV. | TASK TIME INFORMATION DISTRIBUTION TYPE | TASK ELEMENT ERROR FACTOR |
|---|---|---|---|---|---|---|
| 45 | CLRSTS | 4(BN) | (above) | (above) | (above) | 1.0 |

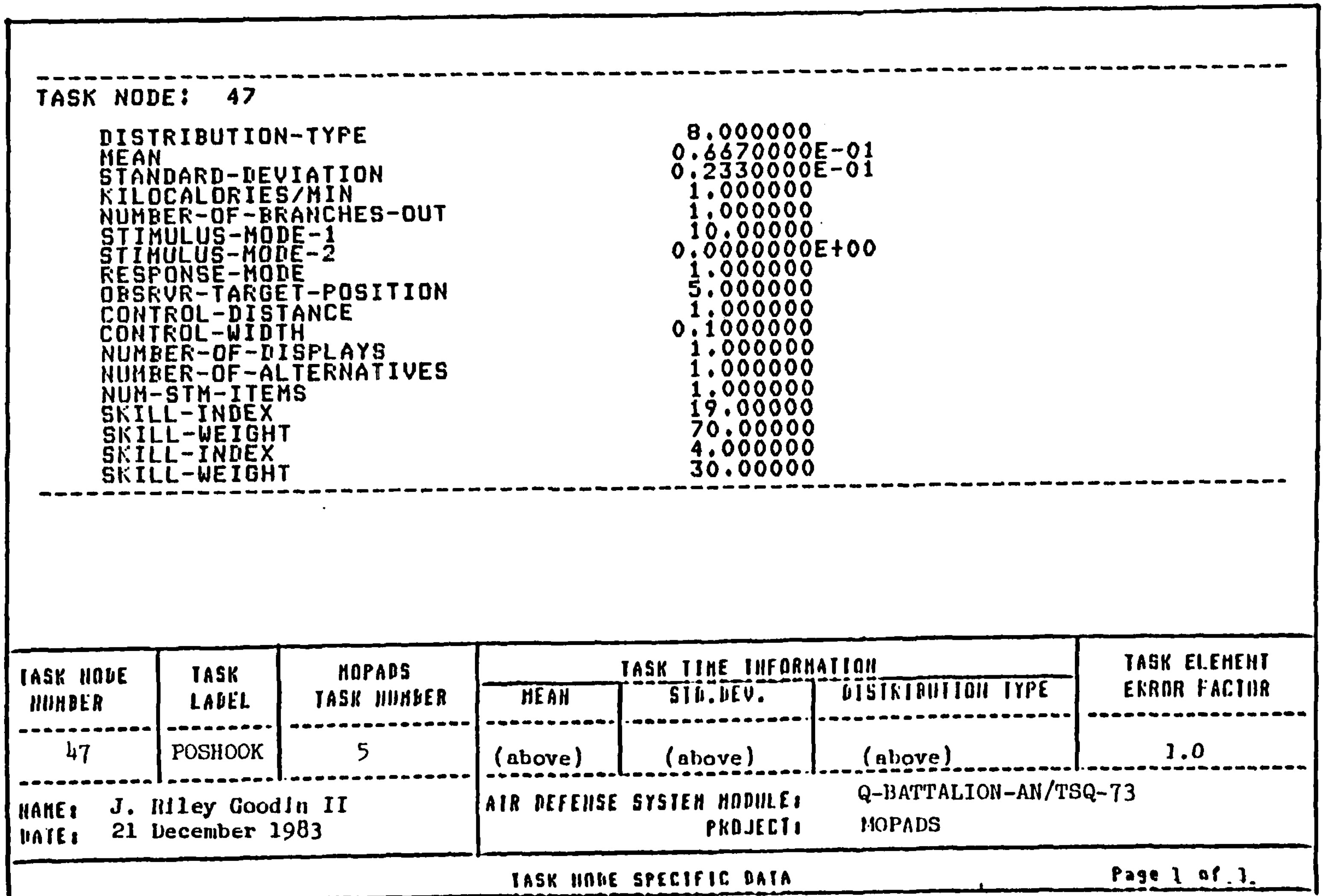
NAME: J. Riley Goodin II
DATE: 21 December 1983

AIR DEFENSE SYSTEM MODULE: Q-BATTALION-AN/TSQ-73
PROJECT: MOPADS

TASK NODE SPECIFIC DATA

Page 1 of 1.

Figure III-6. Example SAINT Task Description for AN/TSQ-73.(Node 45)



```
-------------------------------------------------------------------------------
TASK NODE:   47
     DISTRIBUTION-TYPE                              8.000000
     MEAN                                           0.6670000E-01
     STANDARD-DEVIATION                             0.2330000E-01
     KILOCALORIES/MIN                               1.000000
     NUMBER-OF-BRANCHES-OUT                         1.000000
     STIMULUS-MODE-1                                10.00000
     STIMULUS-MODE-2                                0.0000000E+00
     RESPONSE-MODE                                  1.000000
     OBSRVR-TARGET-POSITION                         5.000000
     CONTROL-DISTANCE                               1.000000
     CONTROL-WIDTH                                  0.1000000
     NUMBER-OF-DISPLAYS                             1.000000
     NUMBER-OF-ALTERNATIVES                         1.000000
     NUM-STM-ITEMS                                  1.000000
     SKILL-INDEX                                    19.00000
     SKILL-WEIGHT                                   70.00000
     SKILL-INDEX                                    4.000000
     SKILL-WEIGHT                                   30.00000
-------------------------------------------------------------------------------
```

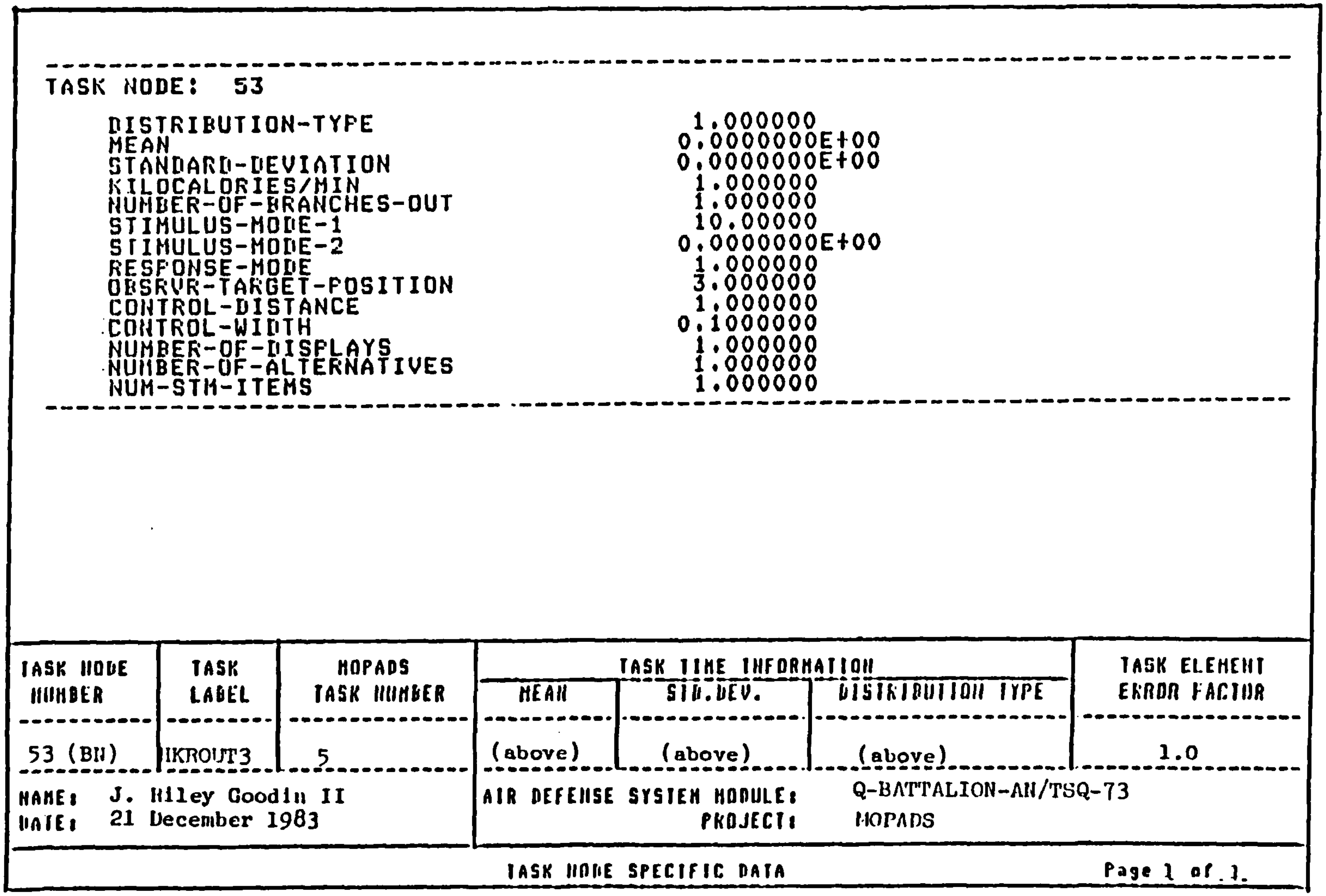
| TASK NODE NUMBER | TASK LABEL | MOPADS TASK NUMBER | TASK TIME INFORMATION |  |  | TASK ELEMENT ERROR FACTOR |
|---|---|---|---|---|---|---|
|  |  |  | MEAN | STD.DEV. | DISTRIBUTION TYPE |  |
| 47 | POSHOOK | 5 | (above) | (above) | (above) | 1.0 |

NAME: J. Riley Goodin II
DATE: 21 December 1983

AIR DEFENSE SYSTEM MODULE: Q-BATTALION-AN/TSQ-73
PROJECT: MOPADS

TASK NODE SPECIFIC DATA Page 1 of 1.

Figure III-7. Example SAINT Task Description for AN/TSQ-73. (Node 47)



```
---------------------------------------------------------------------------------
TASK NODE:  53
    DISTRIBUTION-TYPE                                   1.000000
    MEAN                                               0.0000000E+00
    STANDARD-DEVIATION                                 0.0000000E+00
    KILOCALORIES/MIN                                    1.000000
    NUMBER-OF-BRANCHES-OUT                              1.000000
    STIMULUS-MODE-1                                     10.00000
    STIMULUS-MODE-2                                    0.0000000E+00
    RESPONSE-MODE                                       1.000000
    OBSRVR-TARGET-POSITION                              3.000000
    CONTROL-DISTANCE                                    1.000000
    CONTROL-WIDTH                                      0.1000000
    NUMBER-OF-DISPLAYS                                  1.000000
    NUMBER-OF-ALTERNATIVES                              1.000000
    NUM-STM-ITEMS                                       1.000000
---------------------------------------------------------------------------------
```

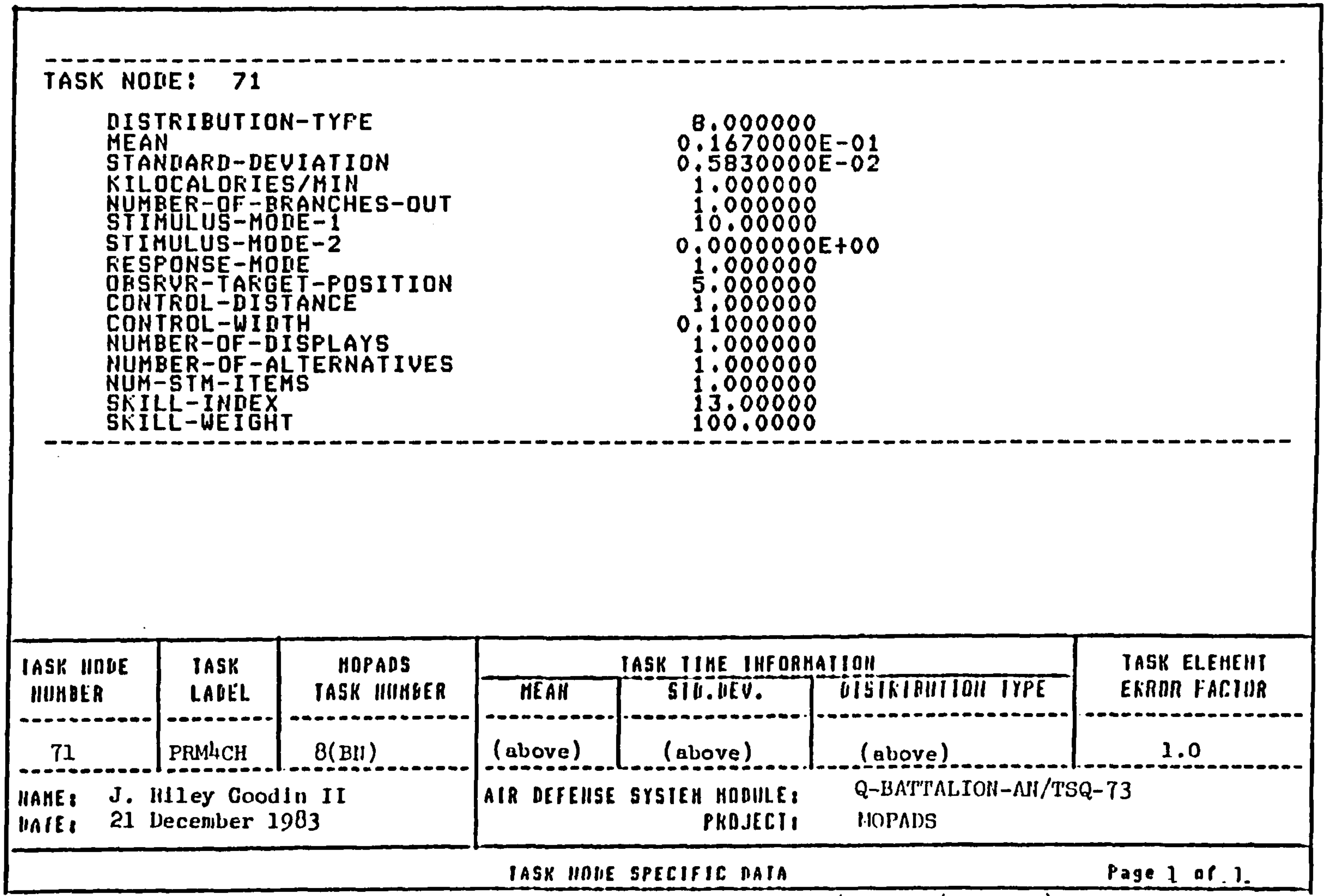
| TASK NODE NUMBER | TASK LABEL | MOPADS TASK NUMBER | TASK TIME INFORMATION<br>MEAN | <br>STD.DEV. | <br>DISTRIBUTION TYPE | TASK ELEMENT ERROR FACTOR |
|---|---|---|---|---|---|---|
| 53 (BN) | IKROUT3 | 5 | (above) | (above) | (above) | 1.0 |

NAME: J. Riley Goodin II
DATE: 21 December 1983

AIR DEFENSE SYSTEM MODULE: Q-BATTALION-AN/TSQ-73
PROJECT: MOPADS

TASK NODE SPECIFIC DATA    Page 1 of 1.

Figure III-8. Example SAINT Task Description for AN/TSQ-73. (Node 53)



```
--------------------------------------------------------------------------------
TASK NODE:  71

     DISTRIBUTION-TYPE                          8.000000
     MEAN                                       0.1670000E-01
     STANDARD-DEVIATION                         0.5830000E-02
     KILOCALORIES/MIN                           1.000000
     NUMBER-OF-BRANCHES-OUT                     1.000000
     STIMULUS-MODE-1                            10.00000
     STIMULUS-MODE-2                           0.0000000E+00
     RESPONSE-MODE                              1.000000
     OBSRVR-TARGET-POSITION                     5.000000
     CONTROL-DISTANCE                           1.000000
     CONTROL-WIDTH                             0.1000000
     NUMBER-OF-DISPLAYS                         1.000000
     NUMBER-OF-ALTERNATIVES                     1.000000
     NUM-STM-ITEMS                              1.000000
     SKILL-INDEX                                13.00000
     SKILL-WEIGHT                               100.0000
--------------------------------------------------------------------------------
```

| TASK NODE NUMBER | TASK LABEL | MOPADS TASK NUMBER | TASK TIME INFORMATION | | | TASK ELEMENT ERROR FACTOR |
|---|---|---|---|---|---|---|
| | | | MEAN | STD.DEV. | DISTRIBUTION TYPE | |
| 71 | PRM4CH | 8(BH) | (above) | (above) | (above) | 1.0 |

NAME: J. Riley Goodin II

DATE: 21 December 1983

AIR DEFENSE SYSTEM MODULE: Q-BATTALION-AN/TSQ-73

PROJECT: MOPADS

TASK NODE SPECIFIC DATA

Page 1 of 1.

Figure III-9. Example SAINT Task Node Description Form for AN/TSQ-73.(Node 71)

Table III-3. IHAWK Operator Tasks Represented in MOPADS.

| Description |
|---|
| ASO TASKS |
| ASO Standby, Wait for Action |
| Detect New Target at CWTDC |
| Establish Target Priority |
| Preempt Lower Priority Target |
| TCC Alert |
| Mark Target as Accepted by TCC |
| FCO A FCO B TASKS |
| FCO Standby, Wait for Action |
| Track Target |
| Obtain Lock on Target Manuallv |
| Put Fire Section out of Action |
| Estimate Raid Size |
| Select Launcher |
| Fire Missiles |
| Evaluate Target Intercept |
| Process Change Targets |
| Put Fire Section back in Operation |
| TCA TASKS |
| TCA Standby, Wait for Action |
| Accept CWTDC Target From ASO |
| IFF Challenge |
| Mark on Reflection Plotter (Friend, Hostile, Unknown) |
| TCO TASKS |
| TCO Standby, Wait for Action |
| Detect IPAR ADP Target |
| Manually Assign Targets |
| IHIPIR Tracking |
| TCO-IHIPIR Does not Acquire Lock |
| Higher Priority Target to be Assigned to Firing Section |
| Process a Hostile Target |
| Assigned Target Determined Friendly |
| Process Friend |
| Evaluate Whether More Missiles Are To Be Fired |
| Give ASO Permission to Cancel Alert |

Table III-3. (continued)

TCO MESSAGE TASKS

- Accept Q-73 Target (ENGAGE MODE)
- Accept Q-73 Target (COVER MODE)
- Accept Change Targets Command
- Receive "HOLD FIRE" Command
- Receive "CEASE FIRE" Command
- Receive "CEASE ENGAGE" Command
- Issue "PRIORITY, PRIORITY" CALL TO Q-73
- Receive "ROGER ENGAGE" Command
- Send Cannot Comply Message to Q-73

### 3-2. IHAWK Messages.

The messages used in the current implementation were shown in Table III-2. This table includes IHAWK messages. MOPADS uses the message concept to represent voice communication within a component as well as data link and voice between units. This mechanism is, then, used in a uniform way to represent all direct communication between individuals in the MOPADS models. Figure III-10 is an example of a voice message specifying the method of fire which is sent between operators of an IHAWK unit. This particular message is sent by the TCO to the Fire Control Officer (FCO) to specify the method of fire for attacking a particular track.

In all other ways, specification of the messages for the IHAWK is analogous to the process discussed in Section III, 2-2.

### 3-3. IHAWK Operator Goals.

The IHAWK operators' goals were shown in Table II-3.

MOPADS MESSAGE DESCRIPTION

MESSAGE ID ELEMENT Page 1 of 1

| Element | Description | Value |
|---|---|---|
| 1 | * Receiver CRN | -- |
| 2 | Operator Type | 6 or 7 |
| 3 | Functional Type | 1 |
| 4 | Message Subtype | 14 |
| 5 | Message Priority | -- |

MESSAGE DATA LINK ELEMENT

| Element | Description | Value |
|---|---|---|
| 1 | Communication Network<br>1-Voice 2-ATDL | 1 |
| 2 | Acknowledgement Required<br>1 - Yes 2 - No | 2 |
| 3 | Unused | -- |
| 4 | ATDL Code (Unused) | -- |
| 5 | * Time Message Sent | -- |
| 6 | * Message Number | -- |
| 7 | * Sender CRN | -- |
| 8 | Sender Operator Type | 3 |
| 9 | Sender System Module Type | 4 |
| 10 | Task Node Number Sent From | -- |

* Must be set at the time the message is sent

VARIABLE MESSAGE FORMAT

| Element | Description |
|---|---|
| 1 | # Words = 2 |
| 2 | Track Column Number |
| 3 | Method of Fire = 1 Shoot-Look-Shoot<br>= 2 Ripple Fire |

MESSAGE SUBTYPE DESCRIPTION

Method of Fire Command. Shoot-Look-Shoot means shoot one then evaluate before shooting another. Ripple means shoot two missiles.

Name: Jack Walker
Date: 8/4/83

System Module: IHAWK
Project: MOPADS

| From | To | From | To |
|---|---|---|---|
| ICO(27) | FCO | | |
| | | | |

Figure III-10. Example IHAWK Message.

### 3-4. IHAWK Operator Task Models.

Task models for the IHAWK have been developed in the same way as those for the AN/TSQ-73. Figure III-11 is the operator definition for the TCA. Each of the operators, tasks, and task nodes has been specified in the same way as for the AN/TSQ-73.

All of the careful data collection and model development information for the AN/TSQ-73 and IHAWK system modules have been delivered to the Army in four volumes (Goodin & Polito (1983a,b; Goodin & Walker, 1983a,b).

## 4-0 AIR SCENARIOS

In addition to representing the air defense system, MOPADS must have a model of the air battle that the air defense system fights. The MOPADS software has facilities to represent the salient features of the air battle environment.

The data required for the air scenario representation are the following.

1. The coordinate system reference point.
2. The locations of all air defense units and protected sites (critical assets).
3. The assignments of critical assets to the fire units who are to protect them.
4. The characteristics of each radar or observer that acquires information about aircraft.
5. The characteristics and flight paths of all aircraft.

### 4-1. The Coordinate System.

MOPADS assumes a flat earth and uses rectangular coordinates. The coordinates, (x, y, z), of all points in the system are given with respect to a user specified reference point. This point may be specified as a longitude and latitude if a specific terrain system is to be specified. Once the reference point is chosen, all other coordinates are specified with reference to it. The units of the x and y coordinate are nautical miles, and the units of z are feet. The +x axis is east, the +y axis is north, and +z is up.

| NODE(S) WHERE CREATED | DESCRIPTION | INFORMATION ATTRIBUTES | | | RESOURCE REQUIREMENTS |
|---|---|---|---|---|---|
| | | ATTRIBUTE NUMBER | DEFINITION | INITIAL VALUE (IF APPROPRIATE) | |
| 46 | Tactical Control Assistant | 1 | Operator ID | - | - |
| | | 2 | Copy Row Number | - | |
| | | 3 | Operator Type | 4 | |
| | | 4 | Current Track Column Pointer | 0 | |
| | | 5 | Internal Mark Time | 0 | |
| | | 6 | Self-Clearing Indicator | 0 | |
| | | 7 | Last Task Node Branched From | 0 | |
| | | 8 | Unused | 0 | |
| | | 9 | IFF Mode (1=Manual,2=Auto) Not currently used (All IFF is manual) | 1.0 | |
| | | 10 | Unused | 0.0 | |
| | | 11 | Unused | 0.0 | |
| | | 12 | Unused | 0.0 | |
| | | 13<br>14<br>15 | Branching | 0.0 | |

Name: Riley Goodin
Date: 11/3/83

AIR DEFENSE SYSTEM MODULE: IHAWK
PROJECT: MOPADS

SAINT ENTITIES

Page 1 of 1

Figure III-11. Example IHAWK Operator Definition Form.

### 4-2. Locations of Air Defense Units, Critical Assets, and Asset-Fire Unit Assignments.

Figure III-12 shows an example layout of the reference point and of protected assets. With this basic configuration, a variety of air defense configurations and air battles can be simulated. One possible configuration is shown in Figure III-13. This figure shows a group AN/TSQ-73 with two battalions. The battalions each have two IHAWK fire units. The critical assets are assigned to fire units as follows: IHAWK 1 protects site 1, IHAWK 2 protects site 2, and IHAWKS 3 and 4 both protect site 3. The circles around the units represent their area of radar coverage.

Many possible configurations could be arranged to protect the three critical assets, and MOPADS allows the MOPADS user to specify the location and assignments of the air defense components.

### 4-3. Characteristics of Viewers.

Each air defense unit may "own" one or more "viewers." Viewers are usually radars (and in the current implementation, they are always radars), but in the case of REDEYE or VULCAN, for example, the viewer might be a human observer.

Each viewer has the following characteristics.

1. maximum range
2. minimum and maximum altitude
3. probability of detection
4. barriers to view
5. a sector of interest

The characteristics above serve to restrict a viewer's ability to detect aircraft. The maximum range and altitude restrictions are self explanatory. The probability of detection is the probability that the viewer will detect an aircraft that is otherwise in its field of view. MOPADS assumes that once an aircraft is detected it remains detected so long as it is in the viewer's field of view.

The MOPADS user may specify barriers-to-view that block out part of a viewer's ability to detect aircraft. The barriers approximate terrain and other limitations that preclude a radar or observer from seeing everything within range. Figure III-14 shows how barriers may be specified. Two types of barriers may be specified: line barriers and wedges.

# COORDINATE AND ASSET DATA

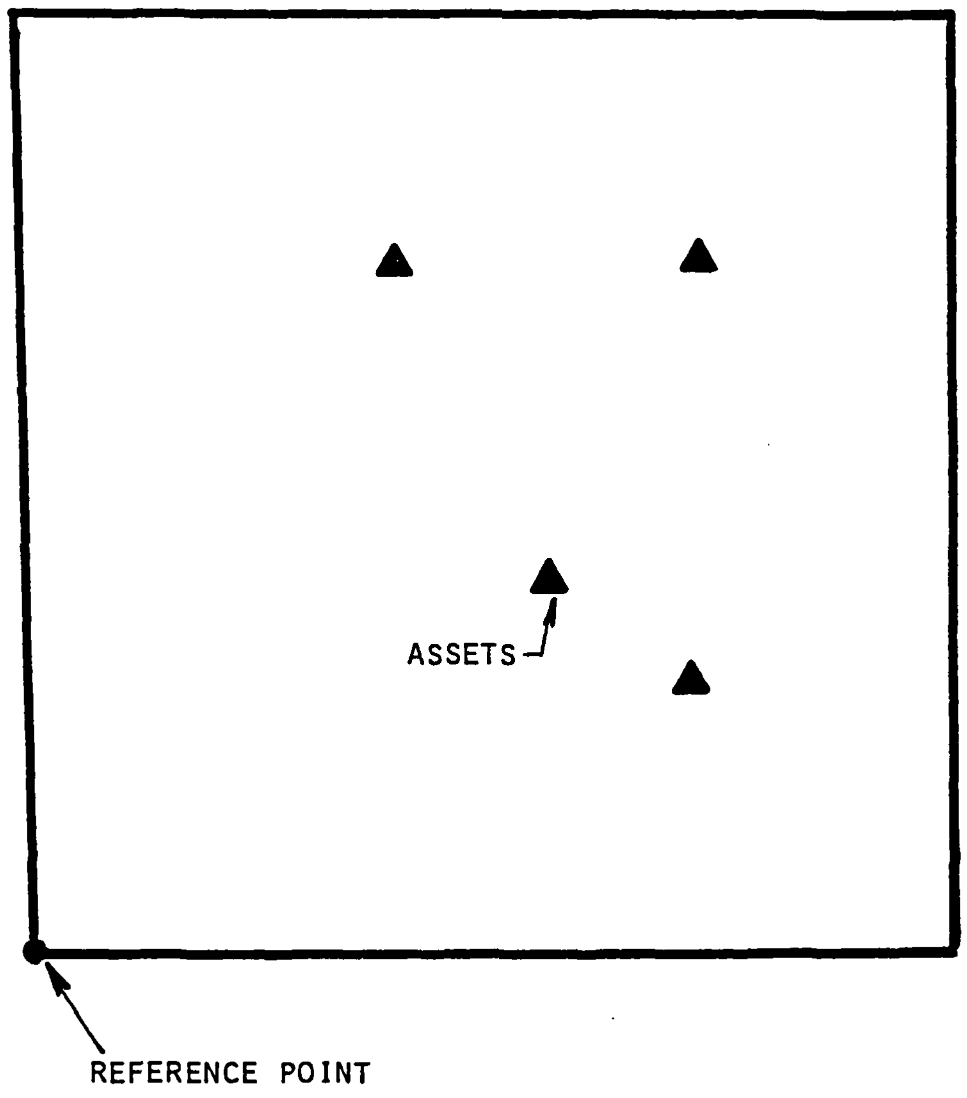
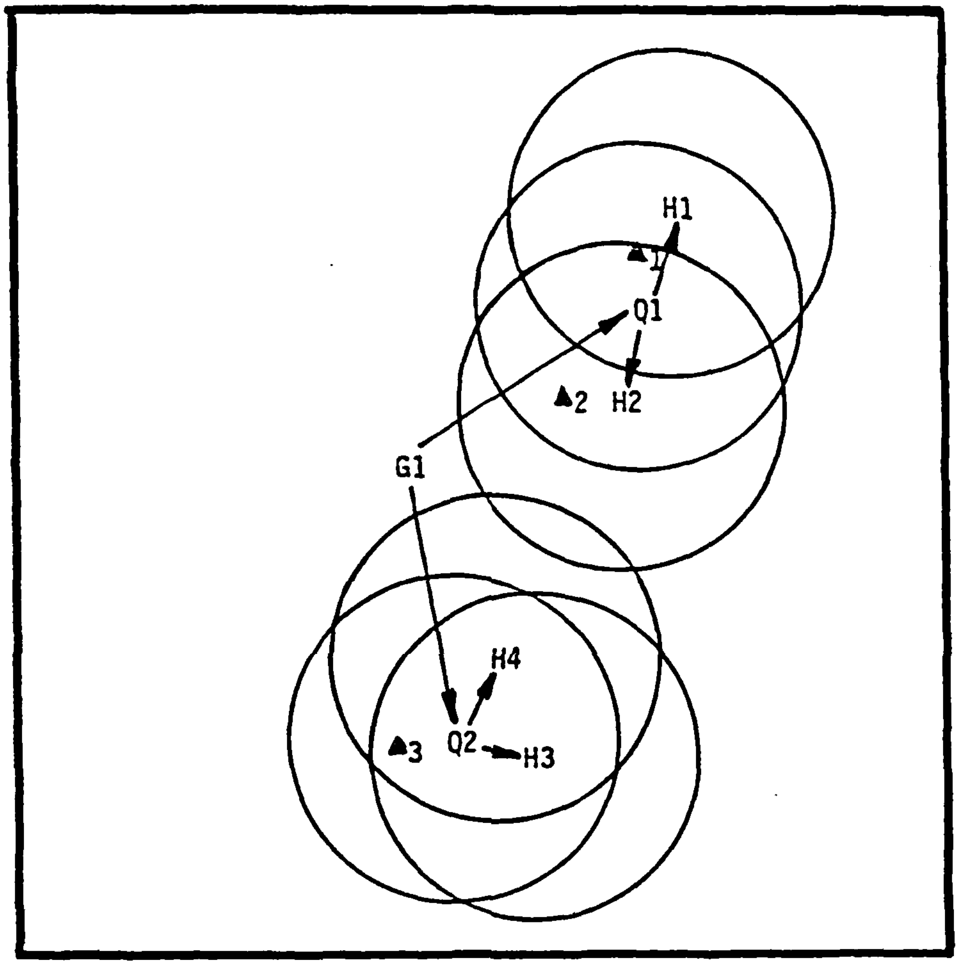


Figure III-12.Example Critical Asset Configuration.

AIR DEFENSE CONFIGURATION

1 GROUP(G) 2 Q-73's(Q) 4 IHAWKS(H)



▲ - CRITICAL ASSET

Figure III-13. Example Air Defense Configuration

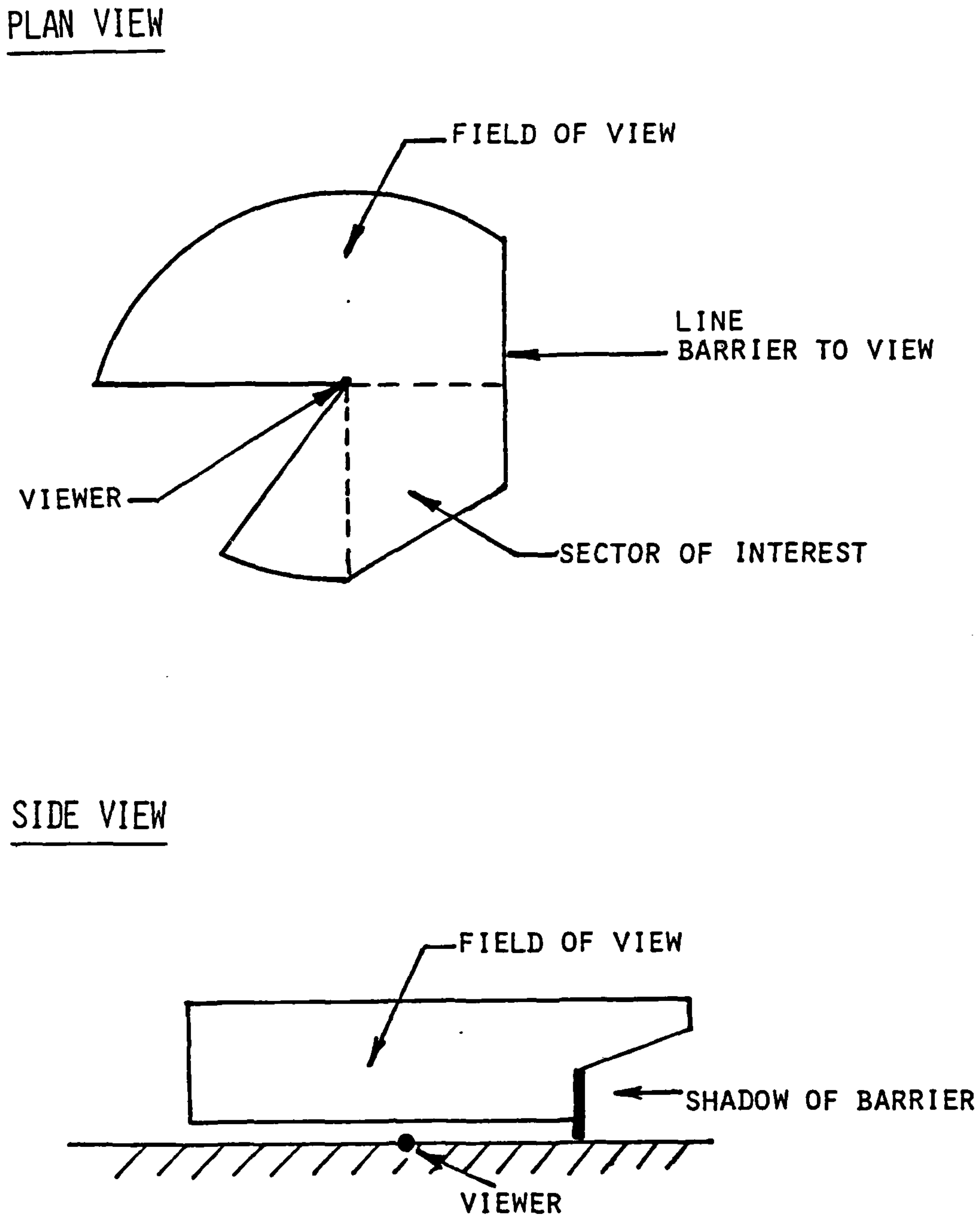


Figure III-14. Viewers and Barriers-to-View.

In the plan view of Figure III-14, two line barriers are shown to the east and southeast of the viewer. Everything farther away from the viewer than the line barrier is hidden from view if it is in the shadow of the barrier. The side view in Figure III-14 demonstrates this. The area to the east of the barrier and below the shadow is hidden from the viewer (note, the side view does not show a complete detail of the viewer in the plan view).

Line barriers may be positioned anywhere in the maximum range of the viewer and the end points may have different altitudes. MOPADS assumes a linear variation of altitude from one end of the barrier to the other. Using multiple line barriers, it is possible to create complex viewing areas that approximate actual radar viewing patterns.

Wedge barriers are simply pie shaped sections in which nothing is visible. An example is shown in Figure III-14 in the plan view to the west of the viewer. The user specifies the start and end compass headings of the wedge.

The system of restricting the field of view described above is not a perfect representation of radar vision limits, but it permits a reasonable approximation for modeling purposes.

The IHAWK permits the operators to specify a "sector of interest" which is a pie shaped segment of its viewing area. The IHAWK computer will automatically process tracks in this sector. The sector of interest has no effect on the radar's ability to acquire aircraft. It serves only to delineate a high interest area to the computer. MOPADS has a facility to specify a sector of interest for each viewer. In the current implementation, the sector of interest is used only by IHAWK.

### 4-4. Characteristics and Flight Paths of Aircraft.

Aircraft flight paths are represented as sequential piece-wise linear segments. The start and end coordinates for each segment are specified along with the speed of the aircraft on the segment. The x, y, and z velocities are computed by MOPADS and assumed constant along the segment. The MOPADS modeler may specify as many segments for an aircraft as desired, and as many aircraft as desired may be included in an air scenario. Curvilinear tracks may be approximated as sequences of linear segments.

The following data are specified for each aircraft:

a. Category A track may be "hostile," "friendly," or "other." An "other" track is a track that cannot be classified as friendly.

b. Identifying Number The user may specify a unique number for each track.

c. Multiplicity The number of aircraft in the flight.

d. Aircraft Type Code values for a variety of aircraft types are provided. See Table III-4.

e. Whether The End Point Of The Segment Is A Target (for hostile tracks only) - This item specifies that the end point of a flight segment is an air defense unit or a critical asset that the air defense system is attempting to protect.

f. Probability of Destruction (for hostile tracks only) - If the end of the segment is a target, the hostile track will attack it. This item is the probability that the site is destroyed.

g. Jamming On A Segment The user may specify that a hostile track is employing ECM on a segment. This information is currently not used by MOPADS. It is provided for future enhancements.

Table III-4. Aircraft Type Codes.

TARGET TYPE CODES

| CODE NO. | NAME | COUNTRY | MISSION |
|---|---|---|---|
| 1 | F4C | USA | |
| 2 | F100 | USA | |
| 3 | T33 | USA | |
| 4 | OTHER JET | | |
| 5 | U1A | USA | |
| 6 | U6A | USA | |
| 7 | OTHER PROP | | |
| 8 | 01A | USA | |
| 9 | OH23 | USA | |
| 10 | OTHER HELICOPTER | | |
| 11 | TANK | | |
| 12 | JEEP | | |
| 13 | TROOP | | |
| 14 | APC | | |
| 15 | TRUCK | | |
| 16 | ZERO | | |
| 17 | HALFTRACK | | |
| 18 | F14 | USA | |
| 19 | F15 | USA | |
| 20 | F16 | USA | |
| 21 | F18 | USA | |
| 22 | MIG21 | USSR | |
| 23 | MIG23 | USSR | |
| 24 | MIG25 | USSR | |
| 25 | SOLDIER(FOOT) | ANY | |
| 26 | MIG-27 | USSR | |
| 27 | SU-17 | USSR | |
| 28 | QIANG Ji-5 | PRC | Ground Attack |
| 29 | R-235G | FRANCE | Military surveillance |
| 30 | MIRAGE 3E | FRANCE | Fighter |
| 31 | MIRAGE F1 | FRANCE | Fighter |
| 32 | MIRAGE 2000 | FRANCE | Fighter |
| 33 | MIRAGE 4000 | FRANCE | Fighter |
| 34 | MIRAGE 4 | FRANCE | Bomber |
| 35 | MIRAGE 5 | FRANCE | Ground Support |
| 36 | MIRAGE 50 | FRANCE | Fighter |
| 37 | AW.660 | GREAT BRITAIN | Military Transport |
| 38 | 698 | GREAT BRITAIN | Bomber |
| 39 | HS748 | GREAT BRITAIN | Military Transport |
| 40 | HS.780 | GREAT BRITAIN | Military Transport |
| 41 | P.1099 | GREAT BRITAIN | Ground Attack |

Table III-4. (continued)

| | | | |
|---|---|---|---|
| 42 | IAI202 | ISRAEL | Military Transport |
| 43 | MIRAGE 3C | ISRAEL | Fighter |
| 44 | KfirC2 | ISRAEL | Fighter |
| 45 | G.222 | ITALY | Military Transport |
| 46 | F1045 | ITALY | Interceptor |
| 47 | MB.326K | ITALY | Strike |
| 48 | S.M.1019E | ITALY | Military STOL |
| 49 | F-1 | JAPAN | Fighter |
| 50 | C.207A | SPAIN | Military Transport |
| 51 | SF-5A | SPAIN | Fighter |
| 52 | HA-220 | SPAIN | Ground Attack |
| 53 | 35XD | SWEDEN | Ground Attack |
| 54 | JA37 | SWEDEN | Fighter |
| 55 | J-1 | YUGOSLAVIA | Strike |
| 56 | Light (e.g.blinking) | | |
| 57 | Digit (digit on a display) | | |
| 58 | MIG-17 | USSR | |
| 59 | MIG-19 | USSR | |
| 60 | SU-7 | USSR | |
| 61 | SU-9PM | USSR | |
| 62 | SU-11 | USSR | |
| 63 | SU-15 | USSR | |
| 64 | SU-19 | USSR | |
| 65 | SU-20 | USSR | |
| 66 | YAK-28P | USSR | |
| 67 | YAK-36 | USSR | |
| 68 | TU-28P | USSR | |
| 69 | IL-28 | USSR | |
| 70 | M-4 | USSR | |
| 71 | TU-16 | USSR | |
| 72 | TU-20 | USSR | |
| 73 | TU-22 | USSR | |
| 74 | TU-26 | USSR | |
| 75 | IL-38 | USSR | |
| 76 | TU-126 | USSR | |
| 77 | MIG-15 | USSR | |
| 78 | L-29 | USSR | |
| 79 | L-39 | USSR | |
| 80 | J-5 | USSR | |
| 81 | J-6 | USSR | |
| 82 | J-7 | USSR | |
| 83 | J-8 | USSR | |
| 84 | H-5 | USSR | |
| 85 | H-6 | USSR | |
| 86 | CJ-6 | USSR | |
| 87 | Y-11 | USSR | |
| 88 | MIRAGE III | FRANCE | |

The organization of scenario information in the data base is shown in Figure III-15. A directory is created called CRITICAL-ASSET-CONFIGURATION which contains the specification of the coordinate reference point and the location of all critical assets. This information is contained in an entry COORDINATE-AND-ASSET-DATA. Then one or more air scenarios (which consist of aircraft tracks) can be specified. Each air scenario contains records for hostile, friendly, and other tracks. Thus, for a single configuration, the MOPADS modeler can create a menu of air scenarios that attack it (see Figure III-16). The MOPADS user then selects from this menu, the scenario that he desires to simulate.

Similarly, more than one CRITICAL-ASSET-CONFIGURATION can be created in the data base, so the user also has a menu of such configurations to choose from. When a simulation is designed, the MOPADS user will select the CRITICAL-ASSET-CONFIGURATION and the air scenario within the asset configuration that he desires to simulate.

### 4-5. The Control System Module.

Every MOPADS simulation will automatically contain a system module that does not represent an air defense unit. This system module, called the Control System Module, is a SAINT model that drives the air scenario. This small SAINT model initiates the tracks at the proper time and simulates their flight paths to their termination points. It also performs statistics collection and information management on all track data.

The control system module also ends the simulation at the time specified by the user. The module consists of six SAINT task nodes and appropriate FORTRAN support programs.

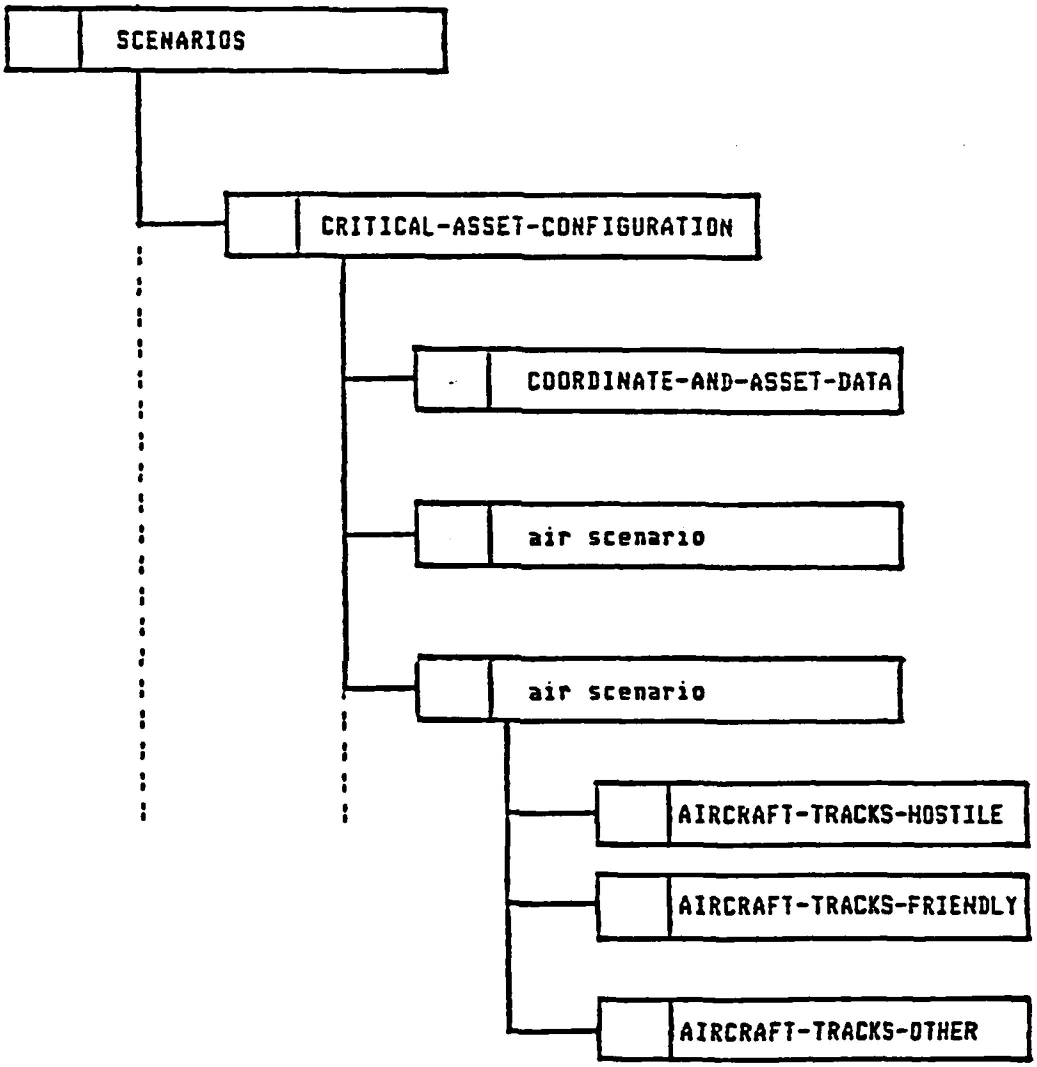


Figure III-15. Data Base Organization of Air Scenario Information.

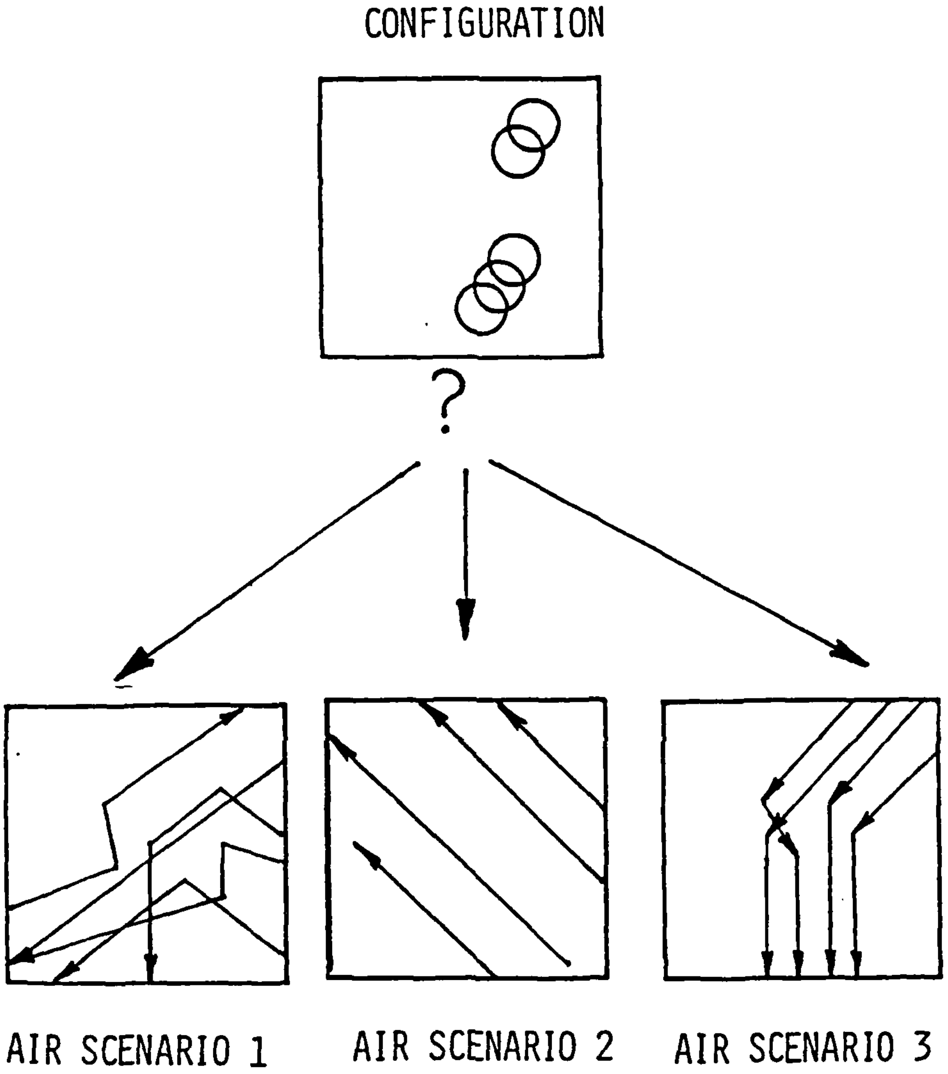


Figure III-16. Selection of Air Scenarios.

# IV. MOPADS PROGRAM IMPLEMENTATION

## 1-0 MSAINT

The SAINT simulation language needed to be modified to accommodate the MOPADS requirement for "run-time configurable" air defense systems. In other words, it must be possible for the MOPADS user to specify the number of air defense units to be simulated and their hierarchical structure. As has been discussed, the logical issues related to this requirement were resolved by modeling the intercommunication among components as messages which are sent through the MOPADS data base. At the computer programming level, however, several issues still needed to be addressed.

Since the MOPADS modeler cannot know in advance how many copies of, say, the IHAWK system module may be required by a MOPADS user, a software scheme is needed to replicate the system modules an arbitrary number of times. SAINT task nodes are numbered (e.g., 1, 2, 3...), so that merely duplicating the task networks would result in duplicate node numbers which are unacceptable to the SAINT processor. The most straightforward approach to this problem is to develop a preprocessor that would renumber the nodes as the network duplicates are created. On the surface, this appears to be an acceptable solution, but it has several drawbacks:

a. Large amounts of computer memory are used for network storage. Replicating essentially identical information will require large computing resources to run MOPADS models.

b. Technical problems result in matching data base information (much of which is keyed to individual nodes) to the (now unpredictable) node number of replicated networks.

c. Complex cross-indexing schemes would have to be developed for the FORTRAN insert programs that are an integral part of each system module so that it would be possible to know which copy was being processed at any given time.

d. Problems (b) and (c) above are compounded when more than one system module type are included in the network.

Using the approach just described would have required a complex programming activity, complex indexing designs for the data base, and complex programming in the FORTRAN inserts. Also, it would have heavily taxed the computing resources.

An alternative scheme is to modify SAINT to "share" a single network representation among multiple realizations of "copies" of the network (Polito & Walker (1982)). This saves substantial computing resources and allows the same node numbers to be used for each copy. This approach has been implemented and, in fact, the node numbers for each system module may be selected independently of the node numbers of all other system modules. The programming to perform this task was, of course, complex, but this task will be performed only one time, and it greatly simplified the design and programming of virtually all other software modules. In particular, the development of air defense system modules is made much simplier because:

a. node numbers may be selected freely without knowledge of node numbers used in other system modules,

b. FORTRAN insert programs do not have to cross index node numbers to determine the copy, and

c. complex cross-indexing of data base information is not required.

This scheme represents the multiple copies of system modules in an elegant manner that simplifies model development and substantially reduces requirements for computing resources.

## 2-0 MOPADS DATA BASE

The MOPADS software must maintain a great deal of information about a variety of subjects:

a. data which describes the Air Scenario to be simulated,

b. data which describes the tactical scenario which includes location and characteristics of air defense units, the command and control system, and the coordinate reference system,

c. data which describes the characteristics of the operators of air defense systems and the environment in which they function,

d. data which describes the dynamic relationships of operator tasks, and

e. data which represents statistics collected during simulations.

All of this information must be maintained in an easily accessed and edited form, and it must be addressed in a way that permits multiple data sets to co-exist without confusion.

The most effective way to accomplish this is with a unified data management system or a data base. The Data Base Control System (DBCS) module of the MOPADS software performs this function.

The DBCS is a non-traditional data base manager. However, the organization of the MOPADS data base file most nearly resembles a hierarchical data base. It is non-traditional in the sense that rapid access of datum elements is needed during MOPADS simulations. The DBCS utilizes a data address that is passed into and out of the DBCS which permits rapid access of required data. The address precludes most hierarchical searches in the data base file to locate data; thus, it eliminates many disk accesses. Furthermore, the DBCS dynamically retains the most frequently accessed data (not merely the most recently accessed) in main memory, so disk reads are minimized. These features make the DBCS a reasonably fast tool for storage and retrieval of data.

The DBCS is a collection of subprograms which any application program may use to create and manipulate a data base file. The DBCS has no main program, so it has no stand-alone capability. It was designed to provide rapid access to data for the MOPADS system. The DBCS can be used in settings other than MOPADS, because it has no built in structure that is unique to MOPADS. It does not, however, have many traditional data base features because of its specialized target environment. It also imposses a somewhat greater burden on application programs than other similar data base systems. The DBCS requires the application program to remember data base address information which is used by the DBCS to implement fast data access.

The DBCS provides capabilities to perform the following functions on data base files:

1. Open/Close DB Files
2. Add/Delete/Rename Directories
3. Add/Delete/Extend/Shorten Data Lists
4. Read/Write Data Lists
5. Set/Change the Data Base Protection Mode
6. Search Directories for Particular Data Lists or Directories.
7. Set/Access/Change Labels of Data Lists and Data List Elements
8. Read/Write External Format Data Files for Portability of Data Base Files
9. Set/Access Various DBCS Options
10. Print Contents of Data Lists and Directories.

In order to reduce the number of disk accesses, the DBCS computes an access frequency for each record, and then keeps the most frequently accessed records in core. Since patterns of record accessing may change over the life of a data base, an exponentially smoothed access frequency (SAF) is calculated and used as the basis for core residence decisions.

Every record in main memory (whether it has been resident for some time or has just been read) has a SAF value which is approximately comparable. In other words, it represents the smoothed access frequency of the record based upon nearly the same criteria. The DBCS uses these values to determine which records will have a tendency to be retained in main memory.

A set of data base application programs (DBAP) have been written to implement the particular data base used by MOPADS. As mentioned earlier, DBCS is generic and has no structural elements that reflect the details of the contents of the MOPADS data base. The DBAP, however, is specialized to MOPADS and utilizes the DBCS to manipulate and manage the data. The DBAP provides many high level data base access functions for the other MOPADS software modules. A schematic of the software structure is shown in Figure IV-1. In particular, DBAP programs are provided for the Human Factors Moderator Functions to access operator characteristics, so the moderator functions do not need to "know" structural information of how MOPADS stores data.

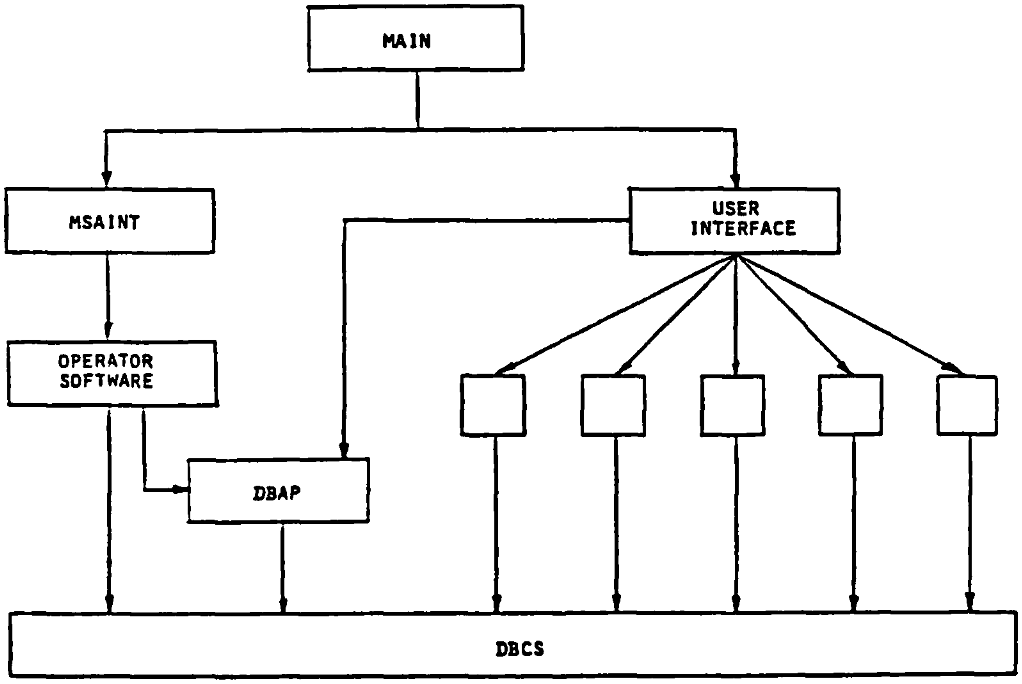


Figure IV-1. DBCS and MOPADS.

Examples of DAB programs are:

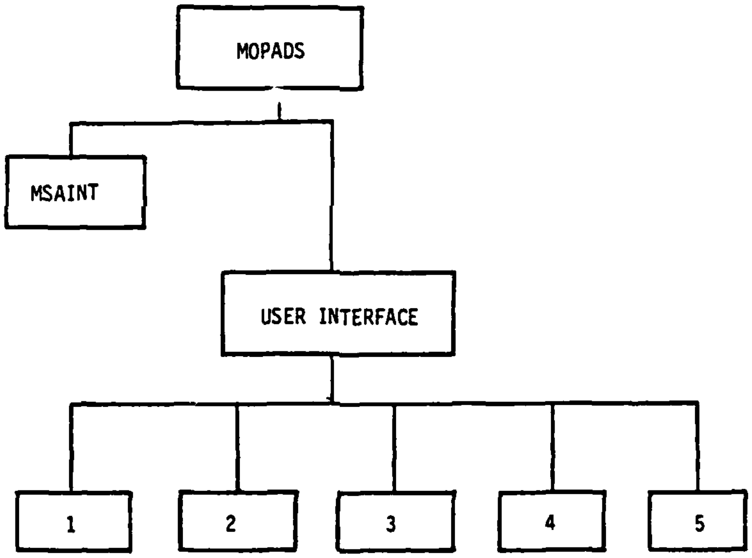
1. Create a MOPADS data base on a new data base file.
2. Copy portions of one MOPADS data base to another.
3. Store/retrieve elements of an operator's state vector.
4. Generate MOPADS error messages related to the data base.
5. Locate all air defense components in a specified simulation data set.

The DBCS is documented in Polito (1983d). The DBAP is documented in Polito (1983e).

## 3-0 THE MOPADS USER INTERFACE

### 3-1. Organization of the MOPADS User Interface.

The MOPADS user interface has five subprocesses which are shown schematically in Figure IV-2. Each of the subprocesses provides facilities for using and modifying the MOPADS data base.



Subprocesses:

1 - Create Simulation Data Set
2 - Set Up Simulation Run Data
3 - Playback Statistics
4 - Create/Edit Air Scenario
5 - Create/Edit Reference System Module

Figure IV-2. Structure of the MOPADS User Interface.

Three of the subprocesses are for use primarily by the MOPADS user. These are the first three: Create Simulation Data Set, Set Up Simulation Run Data, and Examine Statistics. The remaining two subprocesses, Create/Edit Air Scenario and Create/Edit Reference System Module, are primarily for the use of the MOPADS modeler. The functions of the subprocesses are:

1. Create Simulation Data Set - This subprocess allows the MOPADS user to specify a command and control configuration with individually edited copies of reference air defense system modules.

2. Set Up Simulation Run Data - This subprocess allows the MOPADS user to specify data for a MOPADS simulation. The user specifies a simulation data set (previously constructed with subprocess one above), selects the air scenario which will be used, and assigns critical assets to air defense units to be defended.

3. Examine Statistics - This subprocess allows the user to review and selectively print outputs from a MOPADS simulation.

4. Create/Edit Air Scenario - This subprocess provides facilities for the MOPADS modeler to create new air scenarios for the MOPADS user to simulate.

5. Create/Edit Reference System Module - This subprocess provides facilities for the MOPADS modeler to create new system modules for the use of MOPADS users. This process allows the menu of available air defense system modules to be expanded.

Each subprocess is an interactive, command-driven processor.

## 3-2. Create Simulation Data Set.

Using this subprocess, the MOPADS user can create an air defense configuration that may be simulated. Figure IV-3 shows a schematic of the data base operations performed in this subprocess.

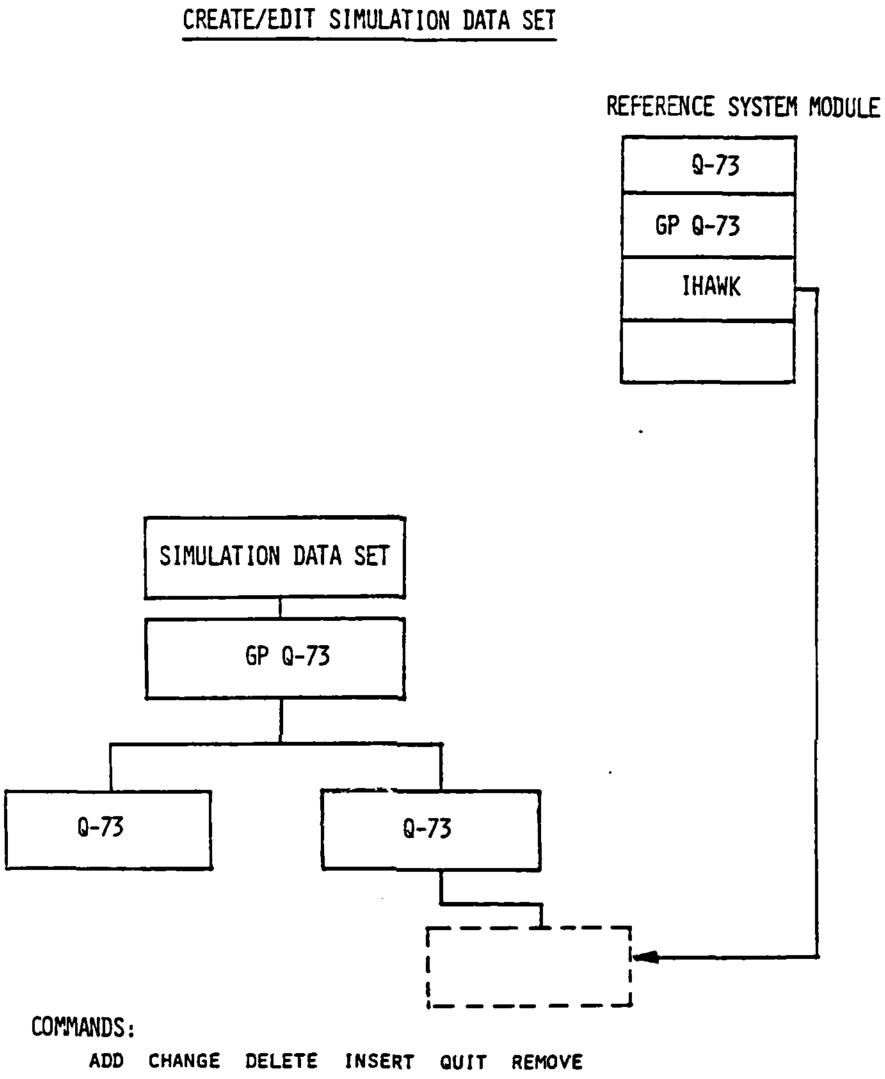


Figure IV-3. Data Base Functions of the "Create Simulation Data Set" Subprocess.

The menu of reference system modules is created by MOPADS modelers using the Create/Edit Reference System Module process. The result is a set of complete system specifications for an AN/TSQ-73, IHAWK, etc. The reference system module contains default values for all human factors and system parameters.

When using the Create Simulation Data Set, the user issues commands that cause copies of the reference system module to be placed in the command and control structure devised by the user. Thus more than one IHAWK or AN/TSQ-73 may be represented in the command and control structure. Each of these copies is called a working air defense system module. The working system modules can be individually edited to reflect differences in human factors parameters or system options.

The commands unique to this subprocess are shown below:

ADD - Create a new simulation data set

CHANGE - Edit the parameters of a working system module

DELETE - Delete an entire simulation data set and all of its working system modules

INSERT - Copy a reference system module to a specified position in the simulation data set

REMOVE - Delete a specified working system module and all of its subordinate units

QUIT - Leave this subprocess of the user interface

Multiple simulation data sets can be created and stored simultaneously in the MOPADS data base. This provides a capability to create a menu of air defense configurations that can be simulated.

3-3. Set Up Simulation Run Data.

With this subprocess, the user completes the information needed to perform a simulation. Figure IV-4 shows the data basic operations. An entry called a Tactical Scenario is created that belongs to a particular Simulation Data Set. The user selects a particular critical asset configuration and air scenario for that asset configuration. Furthermore, he assigns critical assets to fire units for defense and specifies certain technical parameters such as the start and end times of the simulation. Once this information is stored in the data base, it is necessary only to specify the simulation data set and the tactical scenario identifiers to MOPADS in order to perform a simulation.

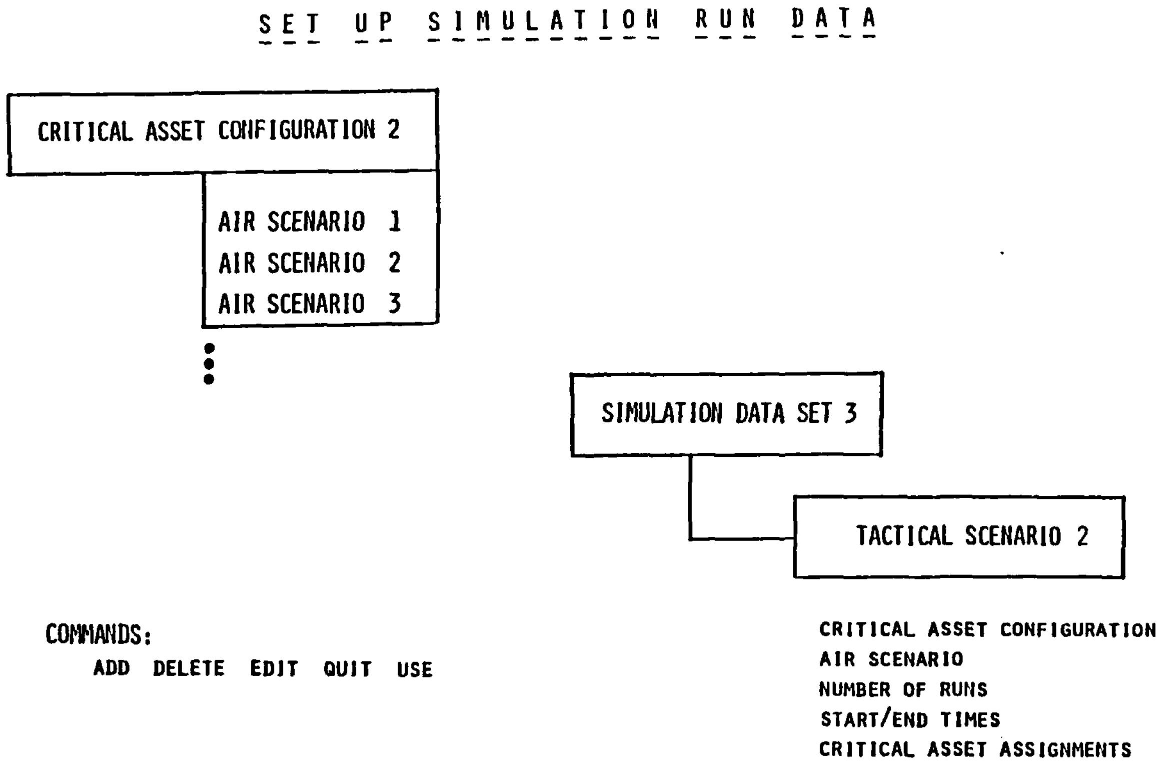


Figure IV-4. Data Base Functions of the "Set Up Simulation Run Data" Subprocess.

The commands unique to this subprocess are:

| | |
|---|---|
| ADD | - Create a Tactical Scenario entry |
| DELETE | - Delete a Tactical Scenario entry |
| EDIT | - Set/revise parameters specified in the Tactical Scenario entry |
| USE | - Select a Simulation Data Set |
| QUIT | - Leave this subprocess |

More than one Tactical Scenario entry may exist in one Simulation Data Set. Also, the statistics produced by a simulation are stored in the Tactical Scenario entry, so the output of a simulation is always stored with the data specification that produced it.

### 3-4. Examine Statistics.

This subprocess is used by MOPADS users to preview and print the results of MOPADS simulations. MSAINT writes row statistical data to the data base after each simulation run. It is physically stored in entries of the Tactical Scenarios. This subprocess has commands that permit the user to selectively print statistics. For example, it is possible to print task node statistics for a particular IHAWK or for all IHAWKs. Similarly, the user can print all operator statistics for one Q-73 or operator statistics for one type (e.g., TD) for all Q-73's.

The commands unique to this subprocess are:

| | |
|---|---|
| DISPLAY | - Print the labels of all Operators and all Working System Modules. |
| PRINT | - Print statistics. |
| SHOW | - Print the Current Simulation Data Set Number, Tactical Scenario Number, and Run Number. |
| USE | - Change the Current Simulation Data Set, Tactical Scenario, and/or Run Number. |

## 3-5. Create/Edit Air Scenario.

This subprocess is used by the MOPADS modeler to create new air scenarios for the MOPADS user. Figure III-15 is a schematic of the organization of the data base information for Air Scenario data. A Critical Asset Configuration entry contains the basic data such as the coordinate reference point and the locations of all critical assets. Then one or more air scenarios can be created with reference to this coordinate and asset system. Each air scenario may contain tracks classified as hostile, friendly, and other.

Commands unique to this subprocess are:

ADD-AIR - Creates a new air scenario entry with tracks

ADD-ASSETS- Creates a new critical asset configuration entry

DELETE - Deletes a single air scenario or an entire Critical Asset Configuration entry

GET - Copies an air scenario or an entire Critical Asset Configuration from a reference data base to the working data base.

RENAME - Renames an air scenario or Critical Asset Configuration

SAVE - Copies an air scenario or an entire Critical Asset Configuration from the working data base to a reference data base.

The user interface contains provisions to maintain reference type information such as scenario data in a reference data base so that it is not subject to accidental loss. It can be copied into working data base files as needed. The GET and SAVE commands perform this function. Use of this subprocess is described in Polito (1983a).

## 3-6. Create/Edit Reference System Module.

This subprocess is used by the MOPADS modeler to enter data for new models of air defense systems into the data base. Figure IV-5 shows the organization of the reference system module information in the data base. When a new system module is being created, and all of the data have been prepared, then this subprocess is used to enter default human factors and system option parameters for the module. This effectively increases the menu of system modules available to the MOPADS user for performing simulations.

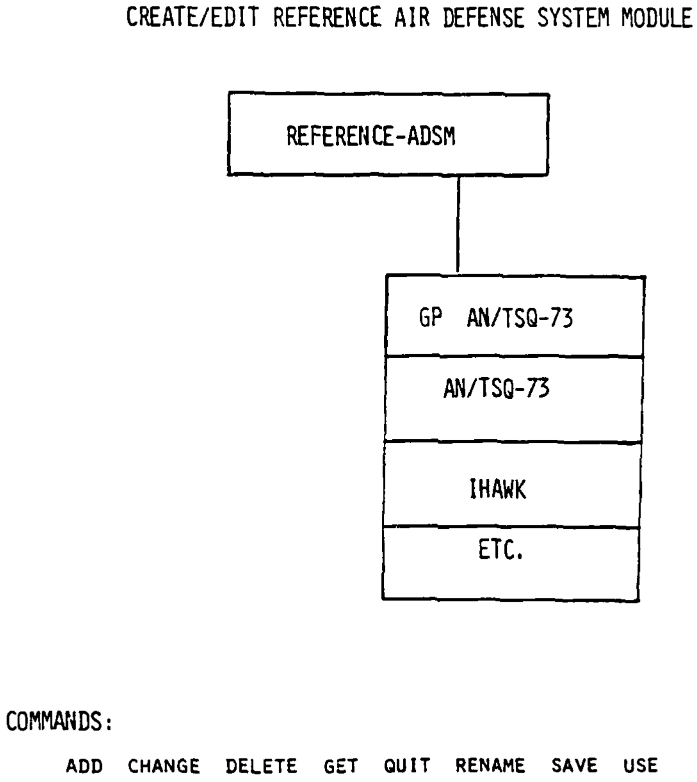

CREATE/EDIT REFERENCE AIR DEFENSE SYSTEM MODULE
REFERENCE-ADSM
GP AN/TSQ-73
AN/TSQ-73
IHAWK
ETC.
COMMANDS:
ADD CHANGE DELETE GET QUIT RENAME SAVE USE

Figure IV-5. Data Base Organization for the "Create/Edit Reference System Module" Subprocess.

The commands unique to this subprocess are:

ADD - Create a new reference air defense system module

CHANGE - Edit the parameters of a reference system module

DELETE - Delete a reference system module

GET - Copy a reference system module from a reference data base to the working data base

RENAME - Rename a reference system module

SAVE - Copy a reference system module from the working data base to a reference data base

USE - Specify a particular reference system module as "current"

QUIT - Leave the subprocess

Use of this subprocess is described in Polito (1983b).

### 3-7. Basic Data Base Commands.

In addition to the commands discussed for each subprocess, there are several basic commands that can be entered from any subprocess. Those commands provide low level editing or data base handling services for the user. The basic data base commands are discussed below.

CLOSE - The CLOSE command will close either the working or reference data base. It can be used to switch to a new data base file.

CURRENT - The CURRENT command will display the label, ID or both of the current directory and/or data list on either data base.

DEPOSIT - DEPOSIT is a low level editing command that allows any element of the current data list to be changed. DEPOSIT interactively requests element numbers and new values.

DIRECTORY - DIRECTORY shows the contents (all owned directories and/or data lists) of the current directory on either data base. It shows the labels, ID's, and directory positions of the contents. This information is useful for the SELECT command.

EXAMINE - EXAMINE will display selected contents of the current data list to the terminal or to the MOPADS non-interactive output file. If the latter is selected, the data list label and other information will also be printed.

HELP - HELP will print the prompts and options for the prompts for the specified command.

MENU - MENU has no prompts. It will print all commands available in the current subprocess.

OPEN - OPEN will open a data base file as either the working or reference data base. OPEN will not automatically close the current data base. CLOSE must be used explicitly before OPEN to switch data base files.

PLINK - PLINK will change the current directory to the owner of the directory which was current when PLINK was issued.

SELECT - SELECT changes the current directory or data list to one that is owned by the directory that is current when SELECT is issued. The desired directory or data list is selected by specifying one (and only one) of the following: 1 - its directory position, 2 - its label, or 3 - its ID. This information is obtained with the DIRECTORY command.

TERMINATE - TERMINATE has no prompts. It will close all open data bases and terminate execution. This is the normal way to end a User Interface session.

## 3-8. Conversing With the MOPADS User Interface.

The User Interface is primarily a command driven processor that waits for the user to issue instructions. It does, however, have aspects of menu driven systems in that some commands result in menus being presented to the user. Also, the command processor (FFSP described in Goodin & Polito (1983a))permits menu-like presentations of commands.

The regular mode for entering commands is shown below.

```
command,prompt1=response1/prompt2=response2/...
```

The commands and prompts are keywords recognized by MOPADS. The responses are particular values for the prompts. For example, consider this:

```
OPEN,FILE=MOPADS.DBF/STATUS=OLD
```

OPEN is the command. FILE and STATUS are prompts recognized by MOPADS and MOPADS.DBF and OLD are values for the prompts.

Certain prompts for a command may have default values that will be used if the prompt is not entered. In the example above, another prompt, DB, specifies which data base is to be associated with the file. Its default is WORKING, so by not entering it on the command line, WORKING is automatically selected. If the default value is not desired, then the prompt must be explicitly entered on the command line.

If the user fails to enter responses to all required prompts, MOPADS will interactively prompt for them. For example,

```
OPEN,STATUS=OLD

FILE[NO DEFAULT] = MOPADS.DBF
```

While processing the OPEN command, MOPADS found that the required prompt, FILE, was not entered. It printed "FILE[NO DEFAULT]=" to prompt the user for a response. If the last non-blank character on a command line is a dash (-), MOPADS will interactively prompt for all unentered prompts, even those with defaults. For example,

```
OPEN,STATUS=OLD -

DB[WORKING] = REFERENCE

FILE[NO DEFAULT] = MOPADS.DBF
```

The dash caused "DB[WORKING] = " to be printed. The value between the brackets is the default for the prompt. The default can be selected by typing "DEF" as the response. DEF can also be entered on the command line; e.g.,

```
OPEN,DB=DEF/STATUS=OLD/FILE=MOPADS.DBF
```

The above demonstrates that the prompt-response groups can be entered in any order.

Also, a command can be cancelled at any time by typing "CANC" as a response or a prompt. For example,

```
OPEN,CANC

OPEN,FILE=CANC
```

Note that DEF and CANC are essentially reserved words. The user interface treats commas, blanks, and equal signs as interchangeable separators. Also, multiple separators are treated as a single separator. This means that the commas in the previous examples could be replaced by any combination of one or more blanks and commas. The same is true of the equal signs, but their use is recommended to make the command lines easy to read. The slashes are required separators between prompt-response groups, but they can be preceded or followed by blanks or commas.

A response may include separators (i.e., commas, blanks, equal signs, and slashes) if it is enclosed in quote marks ("). For example, on some computers file names contain embedded blanks, e.g.,

```
OPEN  FILE="MOPADS DBF"
```

Without the quote marks above, MOPADS will consider MOPADS DBF as two responses when only one is desired. (NOTE: A single prompt may have more than one response if the programmer specified it that way. In such a case, each response would be separated by a blank or comma. In the case above, however, where a single response is required, the quote marks must be used to embed the blank in the response.)

Any response may be enclosed in quotes, although there is no advantage in doing so unless a separator is to be embedded. Blank responses can be entered with "  " where at least one blank appears between the quotes.

A generalization of entering only some of the prompts is to enter only the command name:

```
OPEN

DB[WORKING]=DEF

FILE[NO DEFAULT]=MOPADS.DBF

STATUS[OLD]=DEF
```

The User Interface will prompt for all responses. This method can be selected if the user does not remember the prompts.

For commands which the user issues frequently, a concise mode can be selected by preceding the command with "C-". In this case, the prompt= part of the syntax may be omitted. For example,

```
C-OPEN  DEF/MOPADS.DBF
```

Responses must be entered in the same order as they are prompted in the command-name-only form. No response may be skipped, except that if all remaining responses have defaults and the defaults are desired, then the command line may be terminated (e.g., the STATUS response was omitted above since OLD was desired). The dash works in the concise mode in the same way as in other modes.

The following rules will formalize the previous discussion of how syntax is processed by FFSP.

1. The command-name-only form of a command may be used at any time by typing only the command name.

2. Blank responses and responses containing separators may be entered by enclosing them in quotes. To enter a blank response,type " " (including the quotes). At least one blank must be entered between the quote marks.

3. A command may be cancelled at any time by typing CANC for any prompt or response. You can not abbreviate CANC.

4. The user may elect to use the default value(s) by typing DEF for any response in a response list up to one field past the last response in the list.

5. Slashes (/) must be used to separate one prompt-response group from another. Blanks or commas may be used to separate all other fields. The equal sign should be used to separate prompts from their responses; however, it is not required.

6. Command and prompt names may be abbreviated to any non-ambiguous string of characters. For example, if there are two commands, DESIGN and DESCRIBE, they can be abbreviated DESI and DESC respectively. The commands may be abbreviated in longer forms. For example, the user may enter DESC, DESCR, DESCRI, DESCRIB, or DESCRIBE for the command DESCRIBE.

7. If a command line in regular or concise mode is ended with more than one dash, the last dash will signify to

the system to prompt the user for all the unentered responses. Other dashes will then be considered as part of a response.

8. Any multiple combination of commas and blanks is treated as a single separator. For example,

NAME = BILL WOLF and NAME = BILL , WOLF

are equivalent (here the response is a list of two character strings).

9. If the user enters an incorrect response or misues the syntax, FFSP will explain the error and prompt interactively for all remaining responses.

10. Concise mode is signified by preceding the command name with "C-" (without the quotes).

The user interface is designed to be easy to use and to support novice, intermediate, and expert users. The HELP and MENU commands and the command-name-only form provide extensive help for novice users. Conversely, the concise mode allows frequently used commands to be entered without excessive typing. Furthermore, the ability to use abbreviations for commands and prompts permits experienced users to reduce typing effort in even the regular mode. Examples of User Interface commands and terminal sessions are given in Polito (1983a,b).

## 4-0 UTILITIES AND SUPPORTING SOFTWARE

### 4-1. MOPADS Utilities.

A set of utility programs have been developed to support the rest of the MOPADS software (Polito & Goodin, 1983). These programs provide standard services in loading and copying arrays, managing auxiliary array storage, opening files, and encoding/decoding character strings in a machine independent scheme. These low level services are used by virtually all other MOPADS software modules.

### 4-2. FFSP and FFIN2.

Two software modules are used to provide interactive terminal functions for the user interface. FFIN2(Polito, 1983c) is a set of format free input programs that predate MOPADS. FFIN2 provides extensive error checking for input entered from the terminal, and it protects the user from abnormal termination due to mistyping.

FFSP (Free Format Syntax Process) uses FFIN2 to interpret the syntax of the user interface described in Section 3-7 above. The FFSP is general in nature in that is is entirely data driven. This allows the set of commands now recognized by MOPADS to be expanded easily if the need requires. Indeed, FFSP can be used to interpret commands in a setting entirely removed from MOPADS.

All form, content, and syntax error checking is performed at the lowest level in FFIN2 or FFSP. This relieves the user interface software which implements the commands from the chore of performing these checks for each separate command type. The user interface programs can avoid duplication and concentrate on detection errors in meaning rather than form.

# V. GUIDE TO MOPADS DOCUMENTATION

## 1-0 MOPADS VOLUME 1

This report comprises MOPADS Volume 1.1. It is the final report for the project and provides an introduction to the concepts used in MOPADS. Subsequent MOPADS volumes contain user and refernece material for the MOPADS modeler.

### 1-1. Volume 1 Contents.

| Volume | Title and Description |
|---|---|
| 1.1 | MOPADS Final Report<br>This report describes, in a non-technical manner, the MOPADS modeling system and the MOPADS software. The methodology is described and a simplified description of the software implementation is given. |

## 2-0 MOPADS VOLUME 2

There are no documents in MOPADS Volume 2.

## 3-0 MOPADS VOLUME 3

MOPADS Volume 3 contains reports that provide user documentation. They are mandatory reading for individuals who will design, perform, and analyze simulations using MOPADS. These documents provide sufficient information for a MOPADS user to exercise the models that exist in MOPADS.

### 3-1. Volume 3 Contents.

| Volume | Title and Description |
|---|---|
| 3.1 | User Guide for the AN/TSQ-73 System Module<br>This document describes the AN/TSQ-73 model. It provides information required by a MOPADS user such as a) the number and type of operators, b) the default human factors and system parameters, c) the operator goals, d) other data requirements, and 3) a discussion of the implementation. |
| 3.2 | User Guide for the IHAWK System Module<br>This document is analogous to Volume 3.1 except for the IHAWK system. |
| 3.3 | Performing MOPADS Simulations<br>This document contains user instructions on how to use the MOPADS User Interface to set up and perform MOPADS simulations. It also contains information on analyzing the outputs from simulation. It's intended audience is the MOPADS user. |

## 4-0 MOPADS VOLUME 4

MOPADS Volume 4 is a collection of documents for the MOPADS modeler who will design and develop MOPADS models of new air defense systems and integrate them with the rest of the MOPADS system.

4-1. Volume 4 Contents.

| Volume | Title and Description |
|---|---|
| 4.1 | MOPADS Architecture Manual<br>This document contains a description of the modeling framework and the software structure of MOPADS. It is prerequisite reading for the MOPADS modeler. |
| 4.2 | FORTRAN Style and Documentation Requirements<br>All new FORTRAN code written for the MOPADS project has been written following the standards specified in this document. It is recommended that all follow-on developments also use this standard. |
| 4.3 | Documentation Requirements and Development Guidelines for MOPADS Air Defense System Modules<br>This document describes the documentation procedures that have been used to develop the AN/TSQ-73 and IHAWK system modules. It is recommended that the procedures in this report be used for all follow-on developments also. |
| 4.4 | Development Methodology for MOPADS Air Defense<br>This document describes a step-by-step methodology for developing air defense system modules. It has been used for the AN/TSQ-73 and IHAWK system modules. It is recommended that these procedures be used in all follow-on developments. |
| 4.5 | MSAINT User's Guide: Changes and Additions to the SAINT User's Manual |

This document describes changes made to the SAINT simulation language to support MOPADS. It is written as a supplement and addendum to existing SAINT documentation. Therefore, the reader must be familiar with the SAINT language.

4.6 Creating Reference Air Defense System Modules

This document describes the procedures for using the MOPADS user interface to enter data for a new air defense system module into the MOPADS data base.

4.7 Creating MOPADS Air Scenarios

This document describes the procedures for using the MOPADS user interface to enter new air scenario and critical asset configuration data into the MOPADS data base.

## 5-0 MOPADS VOLUME 5

MOPADS Volume 5 is a collection of reference documents that describe the methodology and software modules of MOPADS. They are intended as reference reports of primary interest to the MOPADS modeler, although MOPADS users may find some of them interesting.

### 5-1. Volume 5 Contents.

Volume — Title and Description

5.1 A Summary of the Literature on Quantitative Human Performance Models

This document contains summary sheets of the literature search conducted of the human performance literature for MOPADS. It is comprised primarily of raw data used for the data base described in Volume 5.2.

5.2 A Data Base for Quantitative Human Performance Modeling

This document describes a computerized system for organizing the information described in Volume 5.1 and the use of the computer program to identify relevant literature to develop the MOPADS skills taxonomy.

5.3 Selected Operator Functions and Tasks for the AN/TSQ-73 Missile Minder

This working document contains raw task flow chart data extracted from AN/TSQ-73 system documentation.

5.4 Selected Operator Functions and Tasks for the Improved HAWK Missile Battery

This document is analogous to Volume 5.3 except for the IHAWK.

5.5 The Underlying Person Model Behind HOMO (Human Operator Model)

This document describes the skill taxonomy for modeling air defense operators.

5.6 HOMO Establishment of Performance Criteria for Non-Decision Making Tasks

This document describes the moderator function approach developed for MOPADS, and the way in which task times are moderated by breaking tasks down into component skills.

5.7 MOPADS Task Sequencing Structure

This document describes the methodology used for operator task sequencing. Operators are represented in MOPADS as goal seekers. This report describes the procedures used to model the goal seeking behavior.

5.8 MOPADS Documentation Style Manual

This report describes the documentation style used for writing and typing MOPADS reports.

5.9 MOPADS Utility Programs

This report documents the MOPADS software module UTIL which contains general program utilities.

5.10 Human Factors Moderator Functions

This report documents the MOPADS software module which implements the human factors moderator functions.

5.11 MOPADS Free-Format Syntax Processor (MOPADS/FFSP)

This report documents the MOPADS software module which interprets the MOPADS user interface command syntax.

5.12 MOPADS User Interface (MOPADS/UI)

This report documents the MOPADS software module that implements the user interface.

5.13 The MOPADS Data Base Control System (MOPADS/DBCS)

This report documents the MOPADS software module that manipulates the MOPADS data base file.

5.14 MOPADS Free-Format Input Program (MOPADS/FFIN2)

This report documents the MOPADS software module that performs low level format free input for FFSP (see Volume 5.11) and the user interface.

5.15 Documentation Manual for the AN/TSQ-73 System Module

This report contains detailed documentation of the AN/TSQ-73 system module. Complete SAINT subnetworks for each operator task are shown with cross references between SAINT task node numbers, operator task numbers, and references to Army documentation. Also, all SAINT user FORTRAN inserts are documented.

5.16 Documentation Manual for the IHAWK System Module

This document is analogous to Volume 5.15 except for the IHAWK system module.

5.17 The MOPADS Data Base

This report specifies the contents and format of the MOPADS data base file.

5.18 The MOPADS Data Base Application Programs (MOPADS/DBAP)

This report documents the MOPADS software module that contains utilities that interact with the MOPADS data base through the DBCS (Volume 5.13).

5.19 Documentation Manual for the MOPADS Control Module (MOPADS/CNTRL) and The MOPADS Common System Module Module Programs (MOPADS/CSMP).

This document is analogous to Volumes 5.15 and 5.16 for a system module called the Control System Module. This module is not an air defense system module. Rather it is a special SAINT subnetwork which manages aircraft tracks and drives the software that updates track position and ATDL information. In addition, certain programs common to all system modules are documented.

# VI. REFERENCES

Finley, D. L., Obermayer, R. W., Bertone, C. M., Meister, D., & Muckler, F. A. Human performance prediction in man-machine systems (NASA-CR-1614) August 1970.

Goodin II, J. R., & Polito, J. MOPADS free-format syntax processor (MOPADS/FFSP) (MOPADS Vol. 5.11). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1983 (a)

Goodin II, J. R., & Polito, J. User guide for the AN/TSQ-73 system module (MOPADS Vol. 3.1). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1983 (b)

Goodin II, J. R., & Polito, J. User's guide for the IHAWK system module (MOPADS Vol. 3.2). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1983 (c)

Goodin II, J. R., & Walker, J. L. Documentation manual for the AN/TSQ-73 system module (MOPADS Vol. 5.15). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1983 (a)

Goodin II, J. R., & Walker, J. L. Documentation manual for the IHAWK sytem module (MOPADS Vol. 5.16). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft.Bliss, TX, 1983 (b)

Laughery, K. R. A data base for quantitative human performance modeling (MOPADS Vol. 5.2). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1981.

Laughery, K. R. & Ditzian, J. L. The underlying person model behind HOMO (MOPADS Vol. 5.5). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1981.

Laughery, K. R. & Ditzian, J. L. A summary of the literature on quantitative human performance models (MOPADS Vol. 5.1). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1982.

Newell, A. & Simon, H. Human problem solving. New York: A Prentice Hall Bool, 1972.

Pew, R. W., Baron, S., Seehrer, C. E., & Miller, D. C. A critical review and analysis of performance models applicable to man-machine systems evaluation, Bolt-Beranek & Newman, Cambridge, MA (BBN-3446). Air Force Office of Scientific Research, Bolling Air Force Base, DC, 1977. (TR No. AFOSR-TR-77-0520)

Polito, J. Creating MOPADS air scenarios (MOPADS Vol. 4.7). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1983 (a)

Polito, J. Creating reference air defense system modules (MOPADS Vol. 4.6). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1983 (b)

Polito, J. MOPADS free-format input programs (MOPADS/FFIN2) (MOPADS Vol. 5.14). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1983 (c)

Polito, J. The MOPADS data base control system (MOPADS/DBCS) (MOPADS Vol. 5.13). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1983 (d)

Polito, J. The MOPADS data base application programs (MOPADS/DBAP) (MOPADS Vol. 5.18). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1983 (e)

Polito, J., & Goodin II, J. R. MOPADS utility programs (MOPADS/UTIL) (MOPADS Vol. 5.9). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1983.

Polito, J. & Walker, J. L. An implementation of "sharable' networks in a process oriented simulation language, presented at the Joint National ORSA/TIMS Conference, San Diego, CA, October 25-27, 1982.

Siegel, A. I., Pfeiffer, M. G., Kopstein, F. S., Wilson, L. G., & Ozkaptan, H. Human performance in continuous operations (ARI Research Product No. 808-4A) Applied Psychological Services, Inc., Wayne, Pennsylvania. Army Research Institute, Alexandria, VA, 1979. (NTIS No. AD-808-6131)

U. S. Army. Technical manual (TM 9-1430-652-10-3) Operator's manual: initialization and operating procedures, guided missile air defense system (AN/TSQ-73, August 1978, Change 6, July 9, 1981.

U. S. Army. Technical manual (TM 9-1430-1526-12-1) Operator and organizational maintenance manual improved guided missile battery control control AN/TSQ-11 1430-01-042-4910, improved HAWK air defense guided missile system, June 30, 1979.

Walker, J. L., & Polito, J. Development methodology for MOPADS air defense system modules (MOPADS Vol. 4.4). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1982 (a)

Walker, J. L., & Polito, J. Documentation requirements and development guidelines for MOPADS air defense system modules (MOPADS Vol. 4.3). Pritsker & Associates, Inc., prepared for the U. S. Army Research Institute Field Unit, Ft. Bliss, TX, 1983 (b)

# VII. CHANGE NOTICES