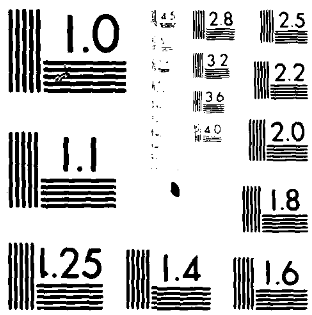
AD-A113 209 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
FUNCTIONAL DESIGN FOR AIR INTERCEPT CONTROLLER PROTOTYPE TRAINI--ETC(U)
DEC 81 E REGELSON, G SLEMON, R VERSTEEG N61339-78-C-0182
UNCLASSIFIED 8641-A009 NAVTRAEQUIPC-78-C-0182-8 NL

1 of 3
AD
A113209



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

Report NAVTRAEQUIPCEN 78-C-0182-8

FUNCTIONAL DESIGN FOR AIR INTERCEPT CONTROLLER PROTOTYPE TRAINING SYSTEM

E. Regelson, G. Slemon, R. VerSteeg, R. Halley
Logicon, Inc.
Tactical & Training Systems Division
Post Office Box 80158
San Diego, California 92138

December 1981

Final Report for Period 28 September 1978 -
28 June 1981

DOD Distribution Statement

Approved for public release;
distribution unlimited.

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NAVTRAEQUIPCEN 78-C-0182-8
2. GOVT ACCESSION NO.: AD-B063643
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): Functional Design for Air Intercept Controller Prototype Training System
5. TYPE OF REPORT & PERIOD COVERED: Final Report 28 SEP 78 - 28 JUN 81
6. PERFORMING ORG. REPORT NUMBER: 8641-A009
7. AUTHOR(s): E. Regelson, G. Slemon, R. VerSteeg, R. Halley
8. CONTRACT OR GRANT NUMBER(s): N61339-78-C-0182
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Logicon, Inc., Tactical & Training Systems Div. Post Office Box 80158 San Diego, California 92138
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 7796-2P3
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Training Equipment Center Orlando, Florida 32813
12. REPORT DATE: December 1981
13. NUMBER OF PAGES: 236
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:
19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
    VTAG
    Air Controller Exerciser (ACE)
    Air Intercept Controller (AIC) Prototype Training System Func. nal Design
20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

This report provides the technical plan for definition, design, development and implementation of the software for the Prototype Air Intercept Controller Training System. The introduction provides a description of the main characteristics of the software, placed in context with the purpose, design and operational concept of the overall training system. The discussion of Program Design addresses the functional and operational consideration of the software. System design discusses the equipments and their interactions and interfaces. The report is concluded in a discussion of training and system constraints.

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

FOREWORD

The Functional Design report provides detailed design data for the AIC experimental prototype system. It provides a translation into programming terminology of the Functional Requirements report. This Functional Design report is intended to serve as a guide for the design of future training systems which incorporate voice technology. The report provides a model of translating behavioral objectives into a fully automated system.

R. BREAUX
Scientific Officer

TABLE OF CONTENTS

TABLE OF CONTENTS - continued

TABLE OF CONTENTS - continued

LIST OF ILLUSTRATIONS

LIST OF EXHIBITS

LIST OF TABLES

SECTION I

INTRODUCTION

PURPOSE

This functional design report is one step in the evolutionary sequence of the experimental Prototype Air Intercept Controller (AIC) Training System or Air Controller Exerciser (ACE). Prior to the publication of this report, several other milestones have been attained. The behavioral objectives for the task of training basic air intercept controller skills have been identified, analyzed, and organized into the Behavioral Objectives report[1] and the Objectives Hierarchy report.[2] A student training course has been established by the Ordinal Syllabus report.[3] The functions that the system will require to support the training approach represented in the structure of the syllabus are identified in the Functional Requirements report.[4] The implementation of trainee testing was explained and test data reporting functions were identified in the Measurement of Student Achievement report.[5] A description of the system as it will be assembled for test and evaluation was provided in the Prototype Configuration report.[6]

The current report deals with the characteristics and design of ACE system functions. It contains the specification of those functions in terms of inputs and outputs, as well as the definition of the hardware, software, and courseware to be used in the implementation of those functions. The functions are identified and described in the main body of the report; the appendices contain detailed specifications and definitions for various individual design features.

---

1. Behavioral Objectives for Air Intercept Controller Prototype Training System, Report NAVTRAEQUIPCEN 78-C-0182-1 (Logicon, Inc.). Naval Training Equipment Center, Orlando, Florida; in press.

2. Objectives Hierarchy for Air Intercept Controller Prototype Training System, Report NAVTRAEQUIPCEN 78-C-0182-2 (Logicon, Inc.). Naval Training Equipment Center, Orlando, Florida; in press.

3. Ordinal Syllabus for Air Intercept Controller Prototype Training System, Report NAVTRAEQUIPCEN 78-C-0182-3 (Logicon, Inc.). Naval Training Equipment Center, Orlando, Florida; in press.

4. Functional Requirements for Air Intercept Controller Prototype Training System, Report NAVTRAEQUIPCEN 78-C-0182-4 (Logicon, Inc.). Naval Training Equipment Center, Orlando, Florida; in press.

5. Measurement of Student Achievement for Air Intercept Controller Prototype Training System, Report NAVTRAEQUIPCEN 78-C-0182-5 (Logicon, Inc.). Naval Training Equipment Center, Orlando, Florida; in press.

6. Prototype Configuration Report for Air Intercept Controller Prototype Training System, Report NAVTRAEQUIPCEN 78-C-0182-6 (Logicon, Inc.). Naval Training Equipment Center, Orlando, Florida; in press.

DESIGN

The design requirements were first identified in the Functional Requirement report and are categorized into three types. Training requirements pertain to those functions governing instruction: the training course structure and content, automated instructor capabilities, automated speech generation and recognition, simulations modeling, and the training enhancement console (TEC). System requirements apply to the hardware equipment used to implement the training system: the central computers and their peripheral devices, the interfaces between the system devices, the system input/output requirements, and the details of the man-machine interaction. System constraints relate the first two types of requirements to the requirements imposed by the real world upon the design implementation, e.g.: state-of-the-art limitations in speech generation and speech recognition; hardware limitations in the radar simulation; software constraints imposed by the operating system, programming language, and hardware limitations of the government furnished equipment (GFE) computer system; training limitations arising from the content, level of difficulty, and partial "psychomotor skill" nature of the course material; and the amount of contracted time, money, and GFE resources available to the project.

OPERATIONAL CONCEPT

The operational concept for ACE is a high technology, stand-alone, computer-based, prototype training system. The operational application for ACE is to provide basic training for AIC candidates, controlling air intercepts and training setups using simulated Naval Tactical Data System (NTDS) equipment. The operational application of ACE, as an integrated system, will utilize automated speech technologies; simulate the AIC and aircraft environments, radar, the NTDS console and communications; and apply automated adaptive training.

There are four basic goals for the training system. The first goal is the most pragmatic--to provide the required synthetics training to enable students to control live aircraft under supervision of an instructor. The other three goals are research and experimentation oriented.

The first research goal is to study the usefulness of automated speech technologies in military training. These technologies include computer generated speech, computer speech recognition, and computer speech recording and playback. Research in this area is very important in respect to training for jobs such as air intercept control where personnel are responsible for transmitting, receiving, and responding to verbal messages in a specialized vocabulary with structured and established syntactical constraints.

The second research goal is to study the applicability and effectiveness of instructional methods using automated, adaptive, performance based instruction. This project will research how well an automated instructor can measure a learner's performance and provide subsequent instruction customized to the learner's problems. The learner and the system will be able to make virtually all of the decisions necessary for completing this segment of the AIC candidate's training. The computer system will keep complete records of student performance and will continually adapt the instruction to support optimized learner progress through the specified curriculum. Human instructors will be included into the system, but their primary roles will be to provide system support and special instruction. It is important to note that this is a learner oriented system with every effort being made to make the system transparent to learner use. The product of the final research goal will be an expression of the lessons learned during the development of this project. The system development documentation of an assortment of effectiveness tests will provide data for the project goal--to provide suggestions and recommendations for the construction and development by the Navy of an operational air intercept controller and anti-submarine air controller training system.

SOFTWARE

ACE software is organized in a top-down, modular, hierarchical manner in three computers. The software is grouped functionally in the Instructor, Simulation, and Speech computers. The functions in each computer are implemented as a fixed set of concurrent processes. A process is synonymous with a user program and is a logically and physically complete execution path. A process may initiate other processes which are referred to as sons. The originating process is called the father. Within each process, functions may be divided into tasks. A task is a logically complete independent execution path through a process. Tasks may be further subdivided into one or more procedures, each of which is a logically and physically complete synchronous body of code that performs a minimal set of functions.

The computers communicate with each other via variable length messages sent over the Multiprocessor Communications Adapter (MCA). The MCA serves as a bi-directional high speed data link. Processes may send and receive variable length messages through ports. A port is a full duplex communications path to a process.

The Instructor Computer is responsible for the overall control of ACE, the automated instructor tasks, keyboard interactions with the human instructor and the student, measurement of student performance, and summary reports of system use and student and class performance. The Simulation Computer is responsible for simulating the radar display, interactions with the TEC, NTDS program simulation, and modelling aircraft and other aspects of the training environment. The Speech Computer is responsible for speech recognition and understanding, digitized speech recording and playback, and speech generation.

The following is a description of the program modules which will implement the ACE functions as described in the Functional Requirements report. For consistency, all modules are described as if they run in an Advanced Operating System (AOS) environment even though AOS, Real-time Disk Operating System (RDOS) and Real Time Operating System (RTOS) will also be used in ACE.

INSTRUCTOR COMPUTER. The Instructor Computer will be the locus of training control because the Adaptive Training Control Process will reside therein. At the deeper level of providing an environment for the training system processes, the Instructor Computer will be both the controller, in the sense of being the initiator of processing, and a link between ACE modules. These functions and others will be distributed among the five processes which reside in the Instructor Computer as follows: the resident Executive process shall maintain the training system environment; the Adaptive Training Control process will coordinate the student's experiences in training; the Performance Measurement process will detect errors in student performance; the System Summary process will collate data and report on system utilization; and the Student Summary process will summarize performance data and prepare reports about individual performance. The following paragraphs describe these processes in more detail.

Instructor Executive Process. This process shall provide the environment in which the training system processes will function. It shall be a sort of training system utility package which will handle the details of message routing, timing, file transfer, etc., for the applications processes. The features it will offer are described in more detail below.

Initialization. The Instructor Executive process shall perform a variety of initialization tasks including:

a. providing options selected by switch settings at runtime;

b. opening and reading the shared page file and initializing the shared page data;

c. creating and establishing communications with all son processes;

d. initiating executive tasks;

e. establishing communications with the other processors by initiating the sync message protocol;

f. starting the system clock.

Shared Page. The Instructor Executive shall cause the system shared data to be maintained in all three processors. The shared data page is divided into a block which is modified by Instructor Computer processes and a block which is modified by Simulation Computer processes. (The Speech Computer tasks will not modify the shared page data.) Once each second, the Instructor Executive shall transfer the portion of the shared page which it modifies to the Simulation Computer.

Receipt of these data shall cause the Simulation Executive to transfer the portion of the shared page it modifies back to the Instructor Computer. The Simulation Executive shall also transfer a subset of the shared page data to the Speech Computer at this time. Upon receipt, the shared page data will be read directly into the shared area and so will be immediately accessible to the son processes.

Message Routing. The executive process shall serve as the ACE postman in all three computers as follows: son processes within a computer will not communicate directly. Instead, all messages to other processes will be sent to the executive process. The executive will monitor an Interprocess Communications (IPC) port to receive these messages, and will determine their destination(s) by table lookup. It will formulate a destination code specific to each computer which is to receive the message. It will then send the message to any of its son processes which are to receive the message via the IPC mechanism (or inter-task message facility in the Speech Computer), and to any other computers which are to receive it via the MCA.

Another task in the message routing function shall monitor the MCA for incoming messages. It will use the destination code formulated by the sender to direct the message to the appropriate son process(es).

Keyboard Monitor. The Instructor Executive process shall control the instructor and student keyboards and cathode ray tube (CRT) displays. The Keyboard Monitor function will respond to son process direction to display text and to accept and verify keyboard inputs.

File Transfer Capability. Because each processor has access to only one of the three system disks, it will be necessary to transfer files from one disk to another. The executive shall perform the file transfers and shall notify the requestor when transfer is complete.

<u>Adaptive Training Control</u>. This process directs the student's training in a flexible manner and provides the following six functions.

Control. This function directs the activities a student will perform when he signs on.

Supplemental Activity. The student sign on procedure and system coordination for speech recognition tests, speech retraining, and speech playback are among the activities provided by this function.

Activity Selection. This function determines the next segment to be executed based on the last non-review and non-remediation segment. It also selects requested supplemental activity.

Segment Definition. This function controls the presentation of the selected training segments by processing the segment commands on a functional basis.

Manual Select. This function coordinates menu display and keyboard input generated by keyboard inputs such as retrain, override, and abort.

Diagnosis, Prescription, and Remediation. This function is responsible for diagnosing a student's performance and prescribing remedies for poor performance in the form of review, additional practice, interactive teaching, and instructor notification.

<u>Performance Measurement and Evaluation</u>. This process determines the correctness of student actions based on a predefined behavioral model. Performance Measurement detects and records errors in student performance so that the instructor may evaluate the student's performance. Performance Measurement functions include the three following functions.

Control. This function serves as the top level executive of the Performance Measurement process. It periodically executes a prescribed set of performance measurement routines so that they can sample the real time data, and asynchronously executes these routines when events are received.

Performance Measurement. This function is implemented as a series of small routines each of which is capable of measuring and scoring a single PMV. Only those PMVs so indicated by the course designer are measured, and, further, only designated PMVs will result in a freeze.

Performance Evaluation. At the end of a practice, evaluation is responsible for creating a Performance Measurement file in which Performance Measurement Variable (PMV) scores are stored for later evaluation by Adaptive Training Control's Diagnosis, Prescription, and Remediation Function.

System Summary. This process is responsible for providing text displays and hardcopy printouts of the student's performance for the instructor. It also provides system utilization reports and handles new student registration.

Student Summary. This process is responsible for providing text displays and hardcopy printouts of the student's performance for the student.

SIMULATION COMPUTER. The Simulation Computer is responsible for coordinating all simulation, display, and Training Enhancement Console (TEC) related activities. The Simulation Computer shall be considered subordinate to the Instructor Computer at all times.

In the on-line mode the Simulation Computer provides an executive process to schedule and synchronize all simulation activities, and to monitor communications with the Instructor and Speech Computers; an NTDS process to control and coordinate the various functions and interactions with the student via the TEC and to process TEC inputs and outputs; and a radar simulation process to simulate a radar display, to update the symbols displayed thereon; and a scenario control and models process, to set up and control specific problems and to control and simulate the actions of the various entities being emulated (modeled). Detailed descriptions of the processes contained in the Simulation Computer follow.

Simulation Executive Process. The Simulation Executive process will provide many of the same functions that the Instructor Executive provides, as follows.

Initialization. The Simulation Executive shall provide all of the initialization functions which the Instructor Executive provides with the exception that it will respond to, but not initiate, the sync message protocol. The Simulation Executive will also start a clock but it will only update the system clock at 100 msec intervals between shared page updates.

Shared Page. The Simulation Executive's function in the maintenance of the shared page is described in the correlative paragraphs above.

Message Routing. The Simulation Executive shall provide the same message routing functions as the Instructor Executive.

Keyboard Monitor. The Simulation Executive will not control a keyboard, however, it shall format text for output to the TEC CRT and provide a subset of the text display features of the Instructor Executive Keyboard Monitoring Facility.

File Transfer Capability. The Simulation Executive shall provide the same file transfer features as the Instructor Executive.

NTDS Simulation Process. This process provides the simulation of the NTDS operational programs and provides the following five functions.

Micronova Input Processing. This function processes TEC inputs (buttons, track ball, etc.) that have been detected by the Micronova. It interacts heavily with the extensive logic in NTDS simulation to determine if the proper actions have occurred in the proper sequence.

TEC Control Outputs. This function provides input to the Micronova to cause specific actions to occur in the TEC. Examples of this include activating the buzzer, lights, and light emitting diodes (LEDs). It is used extensively in the Interactive Teaching mode.

NTDS Simulation. This function provides, to the remainder of ACE, all the logic and interfaces needed to simulate the NTDS operational programs.

Radar Simulation. This function uses the track data section of the shared page to control and simulate a radar display on the display unit (Megatek 7000) by invoking Megatek supplied utilities. It provides four control subfunctions.

a. Sweep, Video and Range Mark Control. This subfunction controls the display of the sweep line and range marks. In addition, it determines if any videos are to be displayed at a given sweep angle and if so, displays it.

b. Scale Control. This subfunction is responsible for handling range settings associated with the display.

c. Offset Control. This subfunction is responsible for handling display offset.

d. Brightness Control. This subfunction is responsible for handling display brightness on an individual basis for the sweep line, range marks, videos, and symbols.

Tracker Simulation. This function maintains track symbology by updating positional data and the display.

Scenario Control and Models Process. This process has a twofold responsibility. The scenario control function processes the predefined scenario commands which define and guide the presentation of the scenario. The Models function provides the capability to simulate the external portions of the environment with which the student must interact.

Scenario Control. This function processes the scenario control commands based on time or the occurrence of an event. It is supported by three subfunctions.

a. Scenario Initialization. This subfunction processes all scenario initialization commands and establishes the initial conditions for the process.

b. Scenario Event Processing. This subfunction processes all scenario control commands whose implementations are dependent on the occurrence of events.

c. Scenario Time Processing. This subfunction processes all scenario control commands whose implementations are dependent on the time from the start of the scenario.

Models. This function is comprised of three subfunctions whose actions model a specific aspect of simulation.

a. Aircraft Model. This subfunction maintains the dynamic motion and characteristics of the aircraft on a track basis.

b. Pilot. This subfunction simulates the actions of the combat air patrol (CAP) and maneuvering aircraft (MAC) pilots. It provides response to student voice input via generated speech outputs and alteration of the CAP or MAC flight characteristics. Specific proximities to other aircraft will initiate pilot speech responses reporting on the "observations" within the environment.

c. Ship's Weapons Coordinator (SWC). This subfunction simulates the voice and console actions of the SWC as heard and seen by the student.

SPEECH COMPUTER. The Speech Computer is responsible for coordinating all speech recognition, speech understanding, and digitized speech recording and playback. It also coordinates and supports voice training and synthesized speech generation activities. While the Speech Computer gets inputs and outputs from both the Instructor and Simulation Computers, it shall be considered subordinate to the Instructor Computer at all times.

In the on-line mode the Speech Computer provides an executive process to schedule and synchronize all speech related activities and handle MCA communications with the Instructor and Simulation Computers, a speech collection and validation process to collect and validate voice reference patterns and maintain them for use during instructional segments, a speech recognition and understanding process to recognize student utterances and attach meaning to them, a digitized speech process to digitize and record student verbalizations and maintain them for playback during replay, and a synthesized speech generation process to simulate the vocalizations of the various entities being modeled.

The Speech Computer will use an RDOS operating system. This operating system is described in the Program Design section of this document. Although RDOS does not support processes in the literal sense of AOS, the groupings of tasks which are called Speech Computer processes are of the same functional level and have the same functional meaning as the processes described for the Instructor and Simulation Computers. RDOS is divided into two "grounds". The Speech Computer Executive, Speech Collection and Validation, and Digitized Speech processes will reside in the background. The Speech Recognition and Understanding and the Synthesized Speech processes will reside in the foreground. Descriptions follow of the processes contained in the Speech Computer.

Speech Executive Process. The Speech Executive Process shall provide many of the same functions that the Instructor Executive process provides, with minor modifications due to the different operating system environment, as described below.

Initialization. The Speech Executive shall perform the following initialization functions:

a. providing options selected by switch settings at runtime;

b. initializing the area in which the subset of shared page variables will be maintained;

c. initializing the intertask communications function to enable communication between background processes and tasks;

d. initiating executive and other background processes;

e. responding to the interprocessor synchronization message.

Message Routing. The Speech Executive will function as a postman in much the same way as the Instructor and Simulation Executives. The only difference is that the communication within the Speech Computer will be via the inter-task message facility for messages between background processes and tasks, and via the inter-ground communications facility for messages between foreground and background processes. Aside from this difference in the mechanisms used to actually send and receive messages internally, the techniques of determining message destinations and distributing messages shall be the same as in the other computers.

File Transfer Capability. The Speech Executive shall provide the same file transfer features that the Instructor Executive provides.

Speech Collection and Validation Process. This process provides the software interface with the remote speech recognition computer (NEC DP-100) and is responsible for maintaining the student voice reference data files on disk and in the NEC DP-100.

Speech Practice. This function allows the student to practice speaking to the system to insure that he is comfortable with the stylization required for collecting voice reference data.

Speech Collection. This function is used to obtain the actual voice reference patterns. The student is prompted to speak the phrases that will be collected.

The NEC DP-100's representation of the phrases is saved on disk for later use as reference data during commented practice or free practice.

Speech Validation. This function may be used after collection to test the quality of the reference data by prompting the student to speak a known sequence of phrases and verifying that the spoken phrases are properly recognized by the NEC DP-100.

Retrain. This function allows the student to re-collect voice reference data for poorly recognized phrases. Functionally, it is the same as Speech Collection.

Voice Test. This function provides feedback about the quality of recognition to the student by displaying the recognitions as they occur without the corrections, substitutions, and interpretations found during commented practice and free practice.

Miscellaneous Collection Functions. The speech collection process also has the responsibility of initializing the NEC DP-100 when the system is brought up, insuring that the proper set of voice reference patterns are loaded into the NEC DP-100 before any commented practice or free practice, and informing the Adaptive Training Control Process of changes in the footkey's state.

Speech Recognition and Understanding Process. This process receives and recognizes inputs consisting of student utterances. It also is responsible for attaching meaning to the utterances so that (1) functions in the Simulation Computer can provide appropriate pilot responses and (2) functions in the Instructor Computer can do performance measurement and evaluation. This process provides the following three functions.

Recognition. The recognition function is essentially an indication of what the hardware has recognized the input to be. This is considered a raw input and is subject to some transformation.

Syntactic Handler. This function is responsible for providing some preprocessing of the raw input based on its values. It helps reduce misrecognitions of sound-alike phrases.

Semantic Interpreter. This function determines the meaning of the preprocessed phrase. It is this data that the system acts upon most often. The function is also responsible for sending a message indicating what the understood and raw phrases were to the Instructor Computer, for use by Performance Measurement, and to the Simulation Computer, for use by the Model process.

Digitized Speech Process. This process is responsible for recording student utterances, replaying student utterances simulated voice messages, and synthesized speech messages, and playing back prerecorded instructional and simulated personnel (SWC and MAC flight crew) messages.

Recording. This function provides digitization and recording of student utterances. These data are time tagged and stored on disk in such a way that they may be easily retrieved for Replay.

Replay. This function accesses the time tagged indices from the Record function, Playback function, and Synthesized Speech process to control and retrieve phrases for playback in a timely fashion. It will, upon request, replay both recorded and generated speech that occurred in the last free practice segment.

Playback. This function provides the capability to access the prerecorded digitized speech files and playback the phrase whose number was presented as input. It maintains a time tagged index to the requested phrase for later use by Replay.

Synthesized Speech Generation Process. This process provides a speech generation capability for voice simulation of the CAP flight crew. The speech synthesizer is a VS-6 Votrax unit.

Votrax Requests. This function supplies phrase numbers of concatenated phonemes to the Votrax. The result is understandable speech.

Request Recording. This function time tags and records the speech requests for later use by Digitized Speech Replay.

SECTION II

PROGRAM DESIGN

INTRODUCTION

ACE software will provide the programs, processes and functions to fulfill the training requirements and operating system requirements within the system constraints. The software must be capable of the following:

a. interacting with instructors and students

b. simulating a complex tactical data base program (NTDS)

c. providing accurate, high resolution emulation of radar displays

d. simulating sophisticated aircraft actions and the human decision making process

e. generating understandable speech and recognizing speech accurately

f. monitoring and directing the activity of a system external to itself

g. providing off-line software to support the above functions.

As such, the primary objective of the software design can be stated as the implementation of the required functions with the available resources. In practice this involves tradeoffs and compromises in both delivered performance and use of resources. Tradeoffs made in the area of performance are not addressed here. This section is concerned only with functional and operational considerations of the software and their effect on resource allocation.

Conservation of basic resources is an important consideration. This includes:

a. main memory - use is normally minimized in order to accommodate all of the programs which must run at the same time

b. execution time - the goal is to have all programs which process critical events do so in the time allotted, with processing for non-critical time situations still completed in reasonable time

c. programming time - this is minimized so that the project is completed on time and within the resources allocated.

A balancing of the resources has been the guiding principle of the ACE software design. For this reason the design does represent a reasonable compromise of many conflicting goals. It is a design which can be implemented within the time and money allocated and can implement the capabilities described in the Functional Requirements document.

The various compromises and trade-offs made during the evolution of the design are not justified or even identified in this document. They continue to be subject to review and possible changes throughout the implementation of the system.

Before delving into the design itself, it is useful to examine some of the features of the application which influence the basic design.

OPERATIONAL CONSIDERATIONS. ACE is essentially a real-time, on-line process control system with certain information storage and retrieval elements imbedded within it. It is real-time in that data input is processed, and the results are output, in time to influence future activity in the system. It is on-line in that the data input is generated at the point of input. It is a process control system in that it directs the activity of a system external to itself.

One of the unusual features of ACE is that it is not only a multi-processor system, but it is also a multi-programming system. Multi-programming allows several programs to execute concurrently, thus giving the system designer a greater degree of design flexibility. Thus within each computer several programs are in execution performing several simpler functions that interrelate to form the complex system that is ACE. It is also significant that the system is driven by a series of external asynchronous events. This characteristic, perhaps more than any other, influences the basic design of the software.

The information storage and retrieval elements are best described as a series of disk files containing data about ACE users, scenarios and lessons that influence the basic actions of ACE, and training conducted for use in the future.

FUNCTIONAL CONSIDERATIONS. The application of ACE as a whole may be described as a series of functions. The definition and delineation of these functions was the first step of the design. After this was accomplished, some of the basic resources needed to execute these functions were established.

The first major effort in this area was the establishment of the need for three computer systems to handle the entire application. This was done very early in the project because it was obvious that a single processor system did not have the power to handle the task.

The next significant effort was the division of tasks between the three systems. It was decided that the most efficient way was along functional lines. The three functional areas are defined as "instructional processing", "simulation processing", and "speech processing". That is, one system is dedicated to support of the instructional/teaching function, another to the simulation/display function, and another to speech generation/understanding. In each computer, the related subfunctions were broken down further into a collection of separate and distinct but interrelated programs. The process will be repeated again at successively lower levels during the detailed design phase until all functions have been defined in sufficient detail for implementation. A more detailed definition of this division of functions between the systems appears later in this document.

Although each of the three systems will be responsible for different functions, the functions are very interdependent and must communicate with each other. This communication will be handled via the high speed data link, the multi-processor communication adapter (MCA), and associated software in the operating system. These data will consist of a series of formatted messages carrying commands, requests, and data. A more detailed explanation of this communication scheme appears later in this document.

PROGRAM CONTROL

ACE requires general supervisory programs for program execution and operation. Descriptions follow of the vendor supplied operating systems used, vendor supplied display software, and the basic system executives controlling each computer.

ACE will be implemented on a suite of three Data General Eclipse S/130 computers and a Micronova. The Eclipse computers and the Micronova share the same word length (16 bits), related internal architectures, and a high degree of instruction set overlap. The major differences between the two are concerned mainly with the size of addressible memory and which operating systems can be supported. Each computer has been given a functional assignment under one of the Data General operating systems as follows:

a. Instructor Computer, using AOS

b. Simulation Computer, using AOS

c. Speech Computer, using RDOS

d. Micronova (TEC functions), using RTOS.

The following paragraphs describe some of the characteristics of each operating system prior to discussing the software implementation. More detailed descriptions are contained in the vendor supplied documentation.

ADVANCED OPERATING SYSTEM (AOS). AOS is a large scale, general purpose, disk based operating system that controls and monitors applications programs on the Eclipse line of computers. AOS manages processor time and memory to provide multiprogramming and timesharing capabilities for up to 63 independent processes. A process is the basic unit to which AOS allocates its resources.

Each process's memory space is occupied by a user program which competes with other processes for use of the system resources such as memory, I/O devices, and central processor time. AOS schedules processes on the basis of priority, which may vary dynamically based on execution history or remain fixed.

Within a process, AOS allows for up to 32 asynchronous tasks, each with individual priority assignments. A task is a logically complete, independent, execution path through a process. AOS always gives control to the highest priority task ready to execute in a process.

On-line application functions will be implemented by a fixed set of processes in order to eliminate delays associated with process creation and termination. Processes within this set will be assigned differential priorities to allow control over order of execution. This is essential to ensure that time-critical activities receive memory and processor time upon request.

In each computer there will be at least one resident process containing the MCA monitor. All other processes will be preemptible or swappable. AOS will 'swap' (swapping is an AOS mechanism which writes the lowest priority suspended process to disk thus freeing its memory for use by another process) a preemptible or swappable process if a resident or another process with a higher priority requests memory and there is currently no free memory available. Preemption allows memory to be over-subscribed such that the total size of all applications programs that must run concurrently can exceed the size of available memory.

Figure 1 illustrates the basic process organization. Brief descriptions of each process follow for quick review. More detailed process descriptions appear elsewhere in this document.

REAL TIME DISK OPERATING SYSTEM (RDOS). RDOS is a medium size, general purpose, disk based operating system that controls and monitors applications programs on the Nova and Eclipse line of computers. RDOS manages processor time and memory to provide foreground/background capability, thereby allowing concurrent execution of two user programs (often referred to as "grounds"). RDOS also supports a variety of system utility programs, multitask programming, and a fairly sophisticated disk file management scheme. In many respects, RDOS provides a subset of AOS with the two "grounds" being analogous to two AOS processes. An important runtime difference between the two systems is in the area of communication between user programs. RDOS supplies only a rudimentary interground write capability that does not utilize the system task scheduler. Because of this, activities in the two RDOS grounds cannot be directly synchronized by interground communication.

REAL TIME OPERATING SYSTEM (RTOS). RTOS is another Data General operating system. RTOS is a small, fast, core resident, general purpose, multitask monitor designed to control a wide variety of real time environments. Since RTOS is core resident, it need not (but may) utilize the resources of a disk drive as a mass storage device. RTOS is a compatible subset of RDOS (and somewhat of AOS). Because of this, programs that execute under RTOS may be developed and tested in RDOS and AOS envirnoments. RTOS provides the following facilities:

a. multitask scheduling and priorities

b. intertask communication and synchronization

c. support for a wide variety of peripheral devices

d. modular, reentrant, and relocatable design.

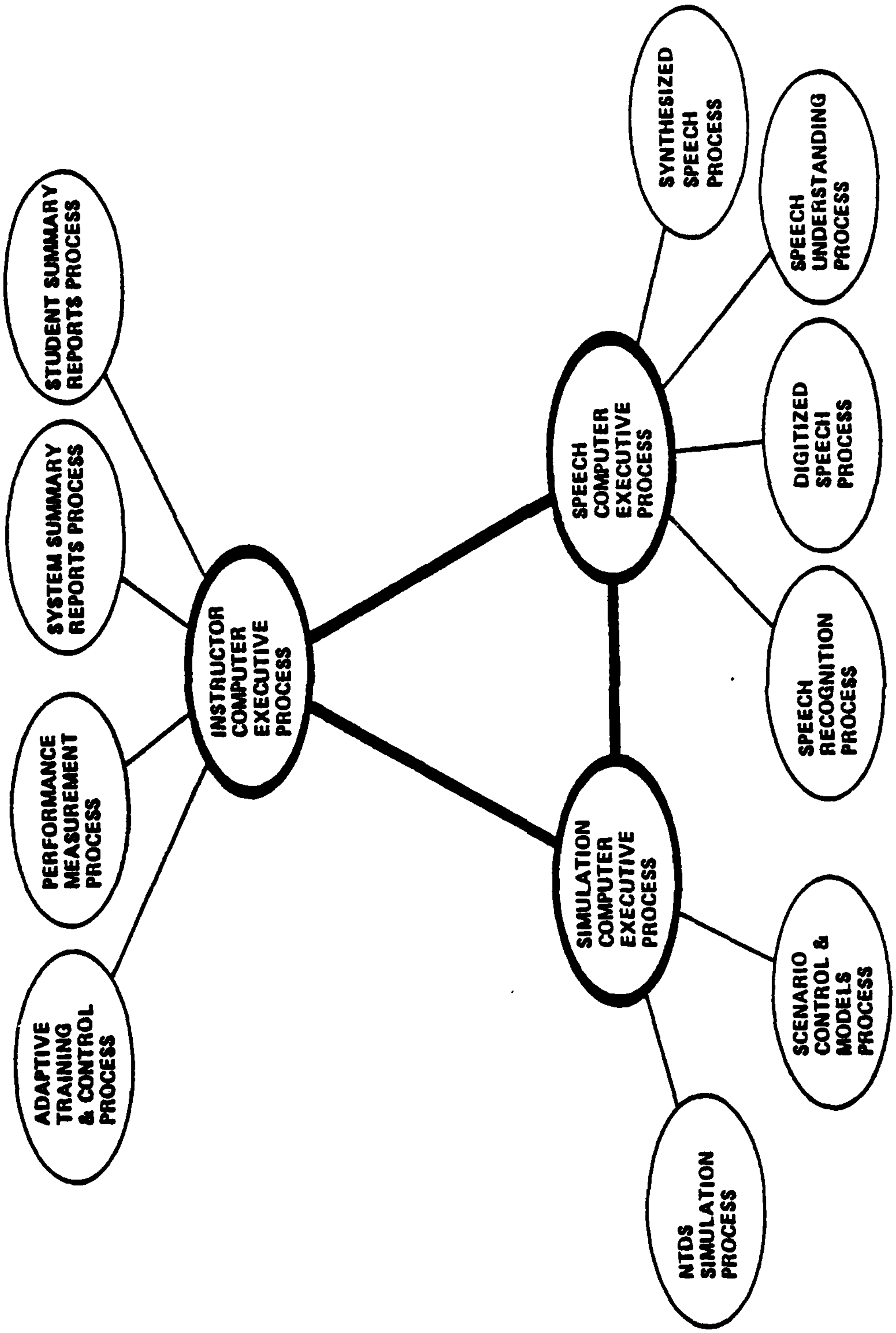


Figure 1. ACE Computer Hierarchy

In ACE, RTOS will be used as the environment for software executing in the Micronova that interfaces with the TEC.

MASTER CONTROL. The overall background running environment and user interface to ACE is provided by the combination of the basic executives for each of the three computers under the direction of the Instructor Executive process in the Instructor Computer and their interaction with the vendor supplied operating systems. The basic function of Master Control is servicing requests. These requests involve use of some resource in the system or the performance of some actions and can be categorized as external (from the user) or as internal (from the system software).

All external requests to ACE (made through either the instructor or the student keyboards, both of which are constantly being monitored by the Instructor Executive) are received, validated, and then processed by the Instructor Executive. In ACE, external requests shall be processed at the highest possible level of the program structure. The type of processing varies with the nature of the request in that the request may be serviced by the Instructor Executive directly, or it may be serviced by one of the other subordinate computers or processes therein.

Internal requests are those that are made by the various software modules within ACE while it is running. It shall be the policy in ACE that these internal requests will be sent to the executive in the computer in which the request originates. That executive will route the request either within the computer or to another computer executive for further routing, based on information in the special message routing table for the originating computer. If the request cannot be honored or is non-recognizable, a non-fatal error will be generated.

Data Flow and Communications. The ACE processors will communicate with each other via the MCA which provides full duplex inter-processor communications. The MCA is analogous, in its software operation, to a magnetic tape unit since the same I/O cycle is used. Normal MCA protocol performs line and data validation functions for each transmission.

A single process will be designated in each processor to receive and transmit MCA messages. The MCA maintains a message queue so that sequential transmission requests will be honored in a first-come first-served manner, automatically.

Within all processors, communications and synchronization between processes will be handled by the IPC mechanism and shared memory.

Processes send and receive IPC messages between ports. A port is a full duplex communications path to a process, and each process can have up to 128 ports. AOS assigns each port a unique 32 bit number; other processes can obtain the port number from the identifying port name.

Tasks within a process may have individual ports. This allows one task to suspend while waiting for an IPC message, while other tasks within the process remain active. A task may receive or send IPC messages to other tasks within the same process as well as to tasks within other processes.

The IPC facility also provides message spooling. Up to 32 messages (of arbitrary length) may be queued for receipt by a single process. Alternatively, message spooling may be disabled so that a message transmission will abort if there is no waiting receiver. Message spooling will normally be enabled for most communications chores, but for certain synchronization and control functions it will be disabled. In addition to the spooling option, a receiver may elect to continue processing if an IPC receive request is issued and no message is queued, or to suspend processing until a message is sent.

All IPC and MCA messages used in ACE will be assigned a unique numeric code. These codes will serve to identify message type and function. Each message type will have an established format that will accommodate variable length data. Receiver tasks will have a range of valid messages which they may receive. To prevent processing of erroneous input, received messages will be validated for proper numeric codes by all receivers before they are processed.

Shared memory is a facility whereby several processes may access the same page of physical memory. AOS allows the shared page to reside in different logical areas in each process that accesses it. When the shared page is referenced by a process, AOS maps the logical addresses into the proper physical addresses resulting in a use mechanism that is transparent to the user. This allows for a convenient global data area for information needed by more than one process.

Initialization Sequence. Initiating ACE operations will involve powering up, loading,and starting the Instructor, Simulation, and Speech Computers, the Micronova, TEC, and DP-100. The consoles and printer must also be powered up. The Simulation and Speech computers may be brought up in either order, but the Instructor computer should be brought up last. All must be running before normal ACE operations are begun.

The Eclipse processors will read a bootstrap loader from disk when the program load switch on the front panel is raised. This loader will bring in the operating system from disk and transfer control to it. Initializing the operating system is accomplished by a simple, interactive session at the maintenance console associated with each computer. A single command, entered after the operating system has initialized, will perform all remaining start-up functions necessary for each processor.

On-line applications software will be started from the maintenance console of each ACE processor by entering a macro command name. This command will create the basic system executive process and transfer control to it. This process will establish communications with the other ACE processors and the peripherals attached to it. The system executive process will create all other on-line processes subordinate to itself so that it is the root of the process hierarchy.

Off-line applications will be initiated by name from the maintenance console. ACE should be in the off-line mode before performance data analysis, scenario definition, or house-keeping chores are performed.

Error Handling. The error handling philosophy in ACE is to detect all possible errors, continue processing after recoverable errors, and bring the system down in an orderly fashion in case of major errors. Detectable errors may originate from file I/O operations, hardware malfunctions, or software consistency checks.

Whenever a hardware error is detected, a message describing the problem will be sent to the maintenance terminal. This will provide a maintenance log to help isolate and correct hardware errors before they become major problems. Software inconsistencies will be reported if they are non-fatal, but in the case of major errors, a termination message will be routed to the maintenance terminal before the system is brought down.

Each applications process will detect those errors peculiar to its function. Error handling routines will provide the error trap and recovery mechanisms. Error recovery will be attempted within the affected process whenever possible. If recovery attempts fail, the instructor control process will be notified of the problem and it will terminate applications processes after issuing an appropriate error message. In the on-line mode, this termination message will be sent to the remaining ACE processors via the MCA in order to notify users of system shutdown.

Reinitializing the processors after a major malfunction will entail analysis of the termination error, possible corrective actions, and execution of a program to detect and correct disk file integrity errors (FIXUP, provided by Data General Corporation). Reconstruction of transient data files that are open when a system failure occurs will not be provided since the data is of a non-critical nature. Normal system initializaton will follow error and disk file structure corrections.

A separate procedure will be designed to handle the major and minor errors. These procedures will be loaded along with each application program as external modules. Both error procedures will send a diagnostic message to the maintenance terminal with an error descriptor, task identification, and process identification. The major error procedure also will cause an orderly but immediate shutdown of the entire system. In order to prevent propagating or disguising errors, such a shutdown will not necessarily save student files. In this case, special file recovery programs will be used to recover any student file data which might be left in an unprotected state.

DATA STORAGE AND SERVICE ROUTINES

COMMON DATA. Common data is defined as data residing in a memory or disk area that is readily accessible by more than one process. ACE will use the AOS shared page mechanism to implement common data. Within each computer's address space, there will exist at least one page defined for access by multiple processes (shared references--see Appendix B for format and description). For purposes of implementation, the shared page shall be considered to be under the control of, and part of, the basic system executive for that computer. Any process requiring data in the shared area shall reference that data using the protocol described elsewhere in this document. In all cases, the amount of, and references to, shared data will be kept to a minimum to facilitate modular program design and integration.

The following represent the criteria for placing data in the shared area:

a. The amount of data required by the processes is larger than will practically fit in an IPC. The key point to consider here is the trade-off between modularity, storage utilization and intrinsic overhead. For messages of up to about 100 words, IPCs are the preferred method of communication. When message lengths increase much beyond this, storage and overhead costs may make IPC use impractical.

b. The frequency of data access is high enough to risk overflowing the receiving process's message queue. Since AOS queues messages at the receiver's port, if the receiver has many time-consuming tasks to perform, its messages may accumulate beyond the limit of 32. Although what constitutes a high access frequency must usually be empirically determined, ACE will consider 10 Hz or greater to be high.

c. The timing requirements of a process or task are critical enough so that IPC usage would seriously degrade system performance. This is also somewhat of a subjective decision, but certain items and areas are excellent indicators of system performance. These include graphics displays, keyboard responses, audiovisual responses, and the like. Data related to these items are likely candidates for placement in the shared page.

When a data structure required by more than one process satisfies one or more of the above conditions, it will be placed in the shared page for the CPU in which the process resides. An entry for each data item also shall be placed in the applicable section of the System Interface Notebook.

Intercomputer Common Data. Data shared between computers shall reside on disk files or in the shared page area and shall be transmitted as needed via the MCA. See Appendix B for MCA definitions and formats. Disk file formats shall be placed in the applicable section of the System Interface Notebook.

Interprocess or Intracomputer Common Data. Data required for sharing between processes shall be limited to that sent via IPC's and that placed in the shared page area (see Appendix B for details). If a process requires data local to itself, then process data blocks may be generated using the vendor supplied macro assembler and external entry points defined to resolve access rights. Examples of common data include:

a. track data table

b. radar display data

c. display specific data

d. system global data.

VENDOR SOFTWARE. ACE will make use of vendor software wherever practicable. This includes operating systems for the Data General Eclipse and Micronova computers, the DGL compiler, the FORTRAN 5 compiler, the macro assembler, and an extensive collection of utility routines to facilitate program development.

A set of routines supplied by Megatek Corporation will provide a high level software interface to the TEC display CRT. Another library of routines will provide an interface to the remainder of the TEC via the Micronova.

Utilities. Data General provides support for a full complement of utilities and high-level languages which allow the user to create, edit, assemble, execute, debug, compile, and manipulate files. Utilities are invoked via the Command Line Interpreter which may read commands interactively from the console or non-interactively from a disk file. While the vendor supplied manuals describe these utilities in great detail, two of them are of sufficient importance to mention here.

RDOS Utility. This utility is supplied by Data General to allow users to upgrade their RDOS software into an AOS environment. The RDOS utility provides the capability to access an RDOS structured disk and transfer files, load and dump files, and list RDOS directories. The utility also allows accessing other RDOS structured devices (i.e., magnetic tape, floppy disk) so that files may be transferred to and from these devices also. It is expected that the RDOS utility will be used extensively to facilitate development of RDOS and RTOS programs in an AOS development system.

RDOSBIND Utility. The RDOSBIND utility allows programs that have been compiled or assembled under AOS to be bound into program files that can be executed in an RDOS or RTOS operating environment. The option of RDOS or RTOS save file format is selectable by appropriate switch setting and the inclusion of a reference to the appropriate system library.

Compilers. The ACE applications programs will be coded primarily in DGL which is a systems implementation language based on Algol 60. DGL's block structured syntax and free form statements make it easy to read. A wide selection of data types, operations, and statements ensures efficient program execution.

DGL optimizes code as it compiles and consequently produces very efficient code. Extensive compile-time error checking is performed as well as optional runtime checking. Several compiler options are available which allow conditional compilation, procedure argument checking, string overflow checking, or full subscript checking.

Limited use of FORTRAN 5 will be made in generating displays since routines provided by Megatek which perform display processing are written in FORTRAN. FORTRAN 5 uses the same meta compiler as DGL with different semantic and syntactic elements specified. Consequently, it is also an optimizing compiler which generates efficient code. Code optimization is its primary enhancement over earlier Data General FORTRAN 5s.

Both DGL and FORTRAN 5 support multitasking by producing shareable, reentrant code. Both languages also support separate procedure compilation (external procedures) and the inclusion of source files (include files) which enhance the modular structure of a program. In addition, the versions of the compilers used on ACE both use the same external linkage protocol. This fact allows a program written in one language to call routines written in another language. Due to this commonality, ACE will be able to further maximize program development while minimizing resource utilization.

Editors. There are two text editors supported by Data General on the Eclipse computers. The first is a character oriented editor named SPEED and the second is a line oriented editor named LINEDIT (available under AOS).

SPEED offers an extensive set of commands which can search for a given string of text or perform powerful iterative operations. Editing is performed by opening one or more disk files containing alphanumeric characters, reading text from the files into one or more edit buffers, modifying the text, then writing text from the edit buffer(s) to the outputfile(s). SPEED uses short command sequences (usually one or two characters) which allow the experienced programmer to quickly and efficiently modify or create a text file.

LINEDIT provides editing capabilities in terms of lines and pages of text. Line numbers optionally precede each source line and are used as arguments to commands which modify, add, delete, move, duplicate, and substitute text. Command mnemonics are words which describe the function of the command such as insert or delete. A help command in LINEDIT will display all available command names or full information about a single command. This feature and the strength of the commands themselves make LINEDIT an easy to learn and easy to use text editor.

These editors will be used extensively in the preparation and modification of programs, documentation, and data files.

Assembler. For assembly language programming on the Eclipse computers, AOS supports a symbolic macro assembler (MASM). The input to MASM can be symbols representing machine instructions or instructions to the assembler called pseudo operators. MASM accepts symbolic addresses which free the programmer from the need to be concerned with exact memory locations.

The macro assembly facility permits the programmer to code only once a set of instructions that will be used many times, and substitute that set anywhere in the source file by merely using a symbolic name for the macro instruction set. One of the more sophisticated macro features allows a macro to have formal arguments which are used within the macro body. To invoke the macro, one uses the macro name with actual argument names and MASM will substitute the actual arguments for the formal names as it expands the macro body during assembly.

MASM will be used sparingly because it requires more programming resources than high level languages. A typical application is the TEC interface where speed and memory constraints dictate that the most efficient code possible be generated.

Debugger/Disk File Editor. The AOS debugger and its companion utility, the disk file editor are both interactive programs which provide valuable tools for program development. The debugger is used to detect, locate, and remove errors from a program. It can control the execution of a program through the use of breakpoints which can be set, deleted, or examined. Program execution may be restarted at arbitrary points within the program or at a breakpoint address. Memory locations, accumulators, and the carry bit may be monitored and altered in several numeric and ASCII formats. Any alterations made during a debugging session are valid only for that session; the changes are not recorded in the program file.

The disk file editor uses a subset of the debugger commands to make permanent changes to program or any other file types.

These two utilities will be used extensively for program debugging.

Binder. The AOS binder utility binds object modules together to form executable program files. Object files are created by the compilers and the macro assembler but are not executable by themselves. The binder resolves external and entry references in the course of constructing a program file and creates a symbol table of all such references and global symbols for use by the debugger and disk file editor. An overlay file may optionally be built by the binder.

A program file is an executable core image that resides on disk until it is brought into main memory for execution. An executing program may bring code modules (overlay files) from disk into an overlay area. Overlaying permits several code segments to occupy the same contiguous area in main memory at different times. The binder builds overlay files and program files.

Runtime Libraries. A library is a series of object modules which are grouped together by the library file editor (LFE). Collecting object modules into libraries provides a convenient way to group object modules for common reference by many programs. For example, mathematical routines are often grouped into a common library. By specifying library names with the list of object modules given the binder when generating a program file, a programmer can write code tailored for specific application problems without duplicating or incorporating into each compilation the source code for commonly used functions.

LFE can create, edit, and analyze library files. It can also merge libraries, extract object modules from the library, and add object modules to a library.

Megatek Library. The Megatek library is a collection of all the FORTRAN callable graphics routines. Some of the routines are coded in assembly language, the remainder in FORTRAN 5. They are broken into six categories by function; display list initialization, jumps and subroutine calls, move and draw, mapping rotation, translation and scaling, picture modifying, and special purpose.

DGL and FORTRAN 5 Runtime Libraries. There are several libraries associated with both of the language processors and separate libraries for interfacing to each of the operating systems used in ACE. During the bind operation, those object modules which are referenced by other routines are pulled from the runtime libraries specified in the binder command line and are collectively bound, with user-written routines to form an executable program file. FORTRAN 5 and DGL generate run-time library references automatically during the compilation process.

DGL has separate runtime libraries which contain: (1) all the DGL callable routines necessary to operate in a multi-tasking environment, (2) math routines, (3) operating system interface routines, (4) routines to establish the program's executions environment, and (5) all DGL program initialization routines.

FORTRAN 5 has a similar complement of runtime libraries plus a standardized set of operating system facilities as defined by the Instrument Society of America.

INSTRUCTOR/TRAINEE SUPPORT

ACE provides instructional support for both the instructor and the trainee. The instructor is kept up to date on student and class progress via summaries generated by the system summary function upon keyboard request. The student's performance is evaluated objectively via the performance measurement and simple testing functions. These functions alleviate the need for the mundane tasks associated with instructor supervision and allows the instructor to focus on student problem areas.

As a provider of computer based instruction, the ACE program design has included the following capabilities:

a. Adaptive training control

b. Performance measurement and simple testing

c. Presentation of instructional material and practice sessions.

The adaptive training control function takes charge of the trainee whenever he is on the ACE system. It requests the presentation of instructional material as well as directs practice sessions as defined in the syllabus. With the aid of the performance measurement function and simple testing function, training inadequacies are discovered and a diagnosis is made. Prescriptions made by the adaptive training control function consist of presentation of additional instructional material or practice or a directive for human instructor intervention. The latter is prescribed when a severe problem is detected.

In order to provide the trainee with an environment that approximates the operational environment, the model functions (NTDS, radar, aircraft, etc.), the speech recognition and understanding, and the speech generation function are employed. Whenever adaptive training control is directed by the syllabus to present instructional material in an operational context or when practice is provided, a request to scenario control is made to initate the creation of such an environment. The models are activated to simulate equipment (NTDS function, radar function), aircraft (aircraft, dynamic bogey,and stranger functions) and personnel (tracker, pilot, and SWC functions). In the last category, the speech recognition and understanding function acts as the "ears" of all personnel models and the speech generation function acts as a "voice" for all personnel models. Thus, a student is acclimatized to perform in the operational environment. Trainee support in terms of progress reports, reviews, and previews are also provided by the system summary and adaptive training control functions.

Although ACE is automated to provide syllabus defined material in a self-paced mode, some degree of trainee/instructor "self-determination" is allowed. The adaptive training control function makes allowances for some instructor and trainee activity selections.

Descriptions of the instructor and trainee functions follow. A clear distinction of functions as to instructor or trainee is not always possible. Instructor related functions are presented first followed by the trainee related functions. Descriptions are formatted to provide inputs, outputs, and processing descriptions. For some functions, a further breakdown into subfunctions was necessary for a clearer discussion.

ADAPTIVE TRAINING CONTROL FUNCTION. This function operates in the Instructor Computer to direct a student's training. The training is not a rigid, predefined entity; it is adaptive, providing the flexibility to meet the needs of individual trainees. Students may elect to review a segment, or they may skip training in areas in which they can demonstrate competency. Practice sessions are of variable length, providing longer sessions for students who are experiencing difficulties, and remediation is provided for problem areas.

The adaptive training control (ATC) function consists of six subfunctions:

a. Control subfunction

b. Supplemental activity subfunction

c. Activity selection subfunction

d. Segment definition control subfunction

e. Manual select subfunction

f. Diagnosis, prescription, and remediation subfunction.

Control Subfunction.

Inputs. The control subfunction receives and screens activity recommendations.

Outputs. Output from the control subfunction is the appropriate training activity.

Subfunction. ATC Control handles a variety of activity request and data report messages. ATC Control receives these messages via the ATC Control IPC port. These activity and data events are processed by ATC message modules. The message processing is highly dependent on message content and on the current ACE mode, that is, the present student activity mode. Data events are forwarded to the other ATC subfunctions. Activity request events are acted upon by ATC Control message modules. Each of these modules perform necessary control activity such as segment selection, supplemental activity selection, menu initiation, training termination, student records upkeep, training recesses and continuances, and student status update.

Supplemental Activity Subfunction.

Inputs. Inputs to the supplemental activity subfunction are data event messages which report key input or coordinate the termination of the supplemental activity.

Outputs. Output consists of text display and coordination messages which initiate or herald the completion of supplemental activity.

Subfunction. There are six supplemental activities: signon, warmup, retrain, validation, replay, and voice test.

a. Signon. Signon is the interface between ACE and the student which permits the student to sign on to the system. The student may sign on whenever the system is inactive. He may not sign on while student statistics are being accessed or when another student is signed on. To sign on, the student enters his last name at the student station. If he has been registered and his files are on the current disk, the files are transferred to the Instructor computer and the student is greeted by ACE. In all other cases, a warning is displayed and the student is given another opportunity to sign on.

b. Warmup. ATC warmup is activated by ATC Control after a student has signed on. ATC warmup determines whether the trainee is a new or continuing student. If the trainee is new (i.e., the voice test introduction segment has not been completed), ATC warmup sends a continue with lesson message to ATC Control. If a continuing student has signed on, ATC warmup generates a voice test request and then a continue with lesson request. As a result, the student is scheduled to enter the voice test activity as his first activity.

c. Speech Activities. The remaining supplemental activities serve as interfaces to the Speech computer (retrain, validation, replay, and voice test) activities. This module introduces the activity, sends activity initiation requests, waits for the activity to complete, and then records the time spent in the activity and what was done during the activity.

Activity Selection Subfunction.

Inputs. The inputs required for activity selection are files which contain the student's performance data, activity request flags, the training syllabus table, the PMV remediation table, and the identity of the last normal segment completed by the trainee. Normal means that the segment was not a review or remediation segment. In addition, the identity of the last segment completed and the performance result are necessary for activity selection.

Outputs. The activity selection subfunction will update the student's record files and identify and initiate the next training activity.

Subfunction. After the completion of any student activity, ATC Control calls the ATC activity select subfunction to record, update and select the next activity. The activity update and selection is based on the next segment in the training syllabus table or PMV remediation table, or activity requested via keyboard input.

The currently signed on student's records are updated on the Instructor Computer during any segment-to-segment transition. This occurs when a segment reaches its normal end, the segment is overridden by the instructor, the segment is aborted by the student, or the segment is exited to provide remediation (via another segment or instructor referral). Note that the detailed segment information will not be included in the update if the student aborts the segment.

A new activity is chosen by activity selection during any run, segment, or supplemental activity transition. Activities are chosen using the following priority:

1. stop after segment request via ^Stop

2. sign off requested via BYE

3. replay requested

4. voice test requested

5. retrain requested

6. instructor referral requested

7. review menu to be shown after normal termination of review segment

8. runs

9. resumption of segment after runs passed

10. review or challenge segments

11. instructor scheduled remediation segments

12. system scheduled remediation segments

13. resumption of segment after remediation for active PMVs completed after passing required commented practice runs

14. next syllabus segment (normal advance)

Activities 1, 2, 10, 11, 12, 13, and 14 are scheduled only if the student is at a segment end transition point.

Segment Definition Control Subfunction.

Inputs. Input to the segment definition control subfunction consists of a segment definition file accessible by the Instructor Computer; see Appendix B for format. Data events are also received as required due to the nature of the segment commands within the segment definition title.

Outputs. Outputs provided by the segment definition control subfunction consist of system control and coordination messages.

Subfunction. This subfunction controls the presentation of a training segment to a student. A training segment is defined by a file of segment definition commands. The syntax for the available commands is contained in Appendix C. The commands are categorized below, describing their associated processing.

a. Segment Descriptions. The DEFINE COMMENTED PRACTICE (CP/H) command provides parameter data for conducting commented practice segments. The DEFINE FREE PRACTICE (FP/H) command defines the number of performance measurement variables failures allowed in a single run as well as advancement criteria for the segment. DEFINE INTERACTIVE TEACHING (IAT/H) command identifies IAT type segments. The DEFINE TEST REQUIREMENTS (T/H) command defines overall test requirements. These requirements are used to evaluate and monitor the conduct of a test provided by a series of DEFINE TEST QUESTION (T/Q) commands and their associated condition commands. END TEST (T/E) terminates a test section of the command file. Similar to conducting test simple testing, check simple testing can be performed. The DEFINE CHECK (CK/H) command defines the requirements, CHECK TEC INPUTS - ANY ORDER (CK/A) specifies TEC inputs for any order, and CHECK TEC INPUTS - ORDERED (CK/O) specifies TEC inputs in particular order. CHECK END (CK/E) terminates a check section of the command file.

b. Sequence Control. ADVANCE TO STEP SENTINEL (ADV), CHALLENGE (CHAL) and REPEAT SEGMENT STEPS (R) cause the segment command sequence to be advanced to a sentinel position, a challenge request to be honored, or a set of commands between sentinels to be repeated before continuing. This sequencing action is directly performed. The END SEGMENT (END) command causes a segment termination status message to be issued.

c. TEC Interface Commands. The TEC interface commands request that information be output to the TEC or matched input be received from the TEC: TEC BUZZER (BUZZ), TEC CRT PAGE (CRT/D), SHOW TEC DROs (DRO), CLEAR DRO (DRO/CLR), LIGHT TEC LEDs (LED), CLEAR TEC LEDs (LED/CLR), DISPLAY NED (NED), RECEIVE TEC (REC/TEC), WAIT FOR (W/F), CLEAR TEC (SC/CLR), SIMULATE NTDS ALERT (TEC), SIMULATE VAB PRESSING (PRESS), SIMULATE NTDS DOWN (NTDS/D), TURN COMMUNICATIONS ON (COMM/ON), KILL TRACK SYMBOL (TRKSYM/KL), CORRECT TRACK SYMBOL POSITION (TRKSYM/PC), HOOK TRACK SYMBOL (TRKSYM/HK), POSITION BALL TAB (POSIT/BT), and SET LABEL STATE (LAB-STATE). For output, the data are formatted and transmitted to the Simulation Computer via the MCA for output to the TEC. The MCA receive will be monitored for the expected TEC input.

d. Student Console Commands. The student is alerted, and text data is presented at the student console, by STUDENT BELL (BEEP), STUDENT CRT MESSAGE (CRT/B) AND STUDENT CRT (CRT/T) commands. RECEIVE STUDENT KEYBOARD (REC/KD) commands anticipate inputs from the student keyboard. Since the student console is connected to the Instructor Computer, direct interfacing is accomplished by this subfunction to perform the commands.

e. Speech Related Commands. Commands COLLECT SPEECH (SP/C), PRACTICE SPEECH (SP/P), VALIDATE SPEECH (SP/V),and GENERATE SPEECH (GS) request phrases to be collected, validated, generated, and recorded. These commands are formatted and transmitted over the MCA to the Speech Computer for execution.

f. Audiovisual Commands. AUDIOVISUAL (AV) commands for audiovisual presentations are performed directly since the Instructor Computer has direct access to the equipment.

g. Scenario Control Commands. SCENARIO CONTROL (SC) commands cause MCA messages to be transmitted to the Simulation Computer to continue and stop scenarios, or stop the segment definition command sequencing until notification of a scenario freeze is received over the MCA from the Simulation Computer. The RUN commands start SC, CP and FP run related activity.

h. Function Commands. Commands which indicate that a message is to be generated to trigger a function include:

(1) CK - simple test, check function
(2) SC - scenario control function
(3) T - simple test, test function

i. Miscellaneous Commands. PERFORMANCE MEASUREMENT VARIABLE (PMV) commands provide performance measurement values for use by the performance measurement and evaluation function in the Instructor Computer. The WAIT command causes a pause to occur.

Manual Select Subfunction.

Inputs. Input to the manual select subfunction consists of requests supplied by the keyboard function and menu initiation requests.

Outputs. Output from the manual select subfunction consists of segment related user requests and processing status messages pertaining to aborts and overrides, and speech retraining.

Subfunction. The manual select subfunction processes the OVERRIDE, ABORT and RETRAIN keyboard functions. Review capabilites are accessible via the ABORT keyboard function. ABORT key also supports the sign off capability.

Most of the subfunction processing requirements are described in MAN MACHINE INTERACTION in Section III.

Diagnosis, Prescription, and Remediation (DPR) Subfunction.

Inputs. The inputs to the DPR subfunction consist of:

a. Test and check results from the simple testing function

b. Run errors and run evaluation data from the performance and evaluation function

c. Notification of segment execution status from the control subfunction (challenge, remediation, etc.)

Outputs. Outputs from the DPR subfunction consist of:

a. A performance report

b. Remediation segment choices

c. An intrasegment repeat choice

d. A test or check repeat choice

e. Advancement recommendations

f. Instructor referral recommendations

g. Performance data files

Subfunction. The DPR subfunction provides diagnosis, prescription, and remediation data for the three teaching areas: CP, FP, and Simple Test. The RUN command causes CP or FP processing to occur. The simple test processing occurs in response to the simple test function. Once activated, the DPR receives performance inputs, reports are generated and presented, and remediation or advancement pathways are recommended.

The student's performance determines much of the segment-to-segment path taken to complete the training course, although the instructor or the student may at times override the ACE system selected segments (the instructor via "overrides"; the student via "challenge" and "review"). A discussion follows of the manner in which the ACE system diagnoses and prescribes remedies for weak student performance and its effect on the segment-to-segment path.

a. Diagnosis.

A nominal path through the training syllabus is defined by the syllabus table. The student proceeds from segment to segment until training course completion. Within each segment, activities are performed sequentially in the order specified in the segment command file. CP and FP segments, in addition, present a minimal number of training exercises (runs) which the student is to perform. Disruption of the nominal path occurs when the student's performance is considered inadequate.

Diagnosis of weak performance areas is based on TEST and CHECK results in IAT segments and on run performance in CP and FP segments. TEST and CHECK definitions have associated pass/fail criteria. Thus, a TEST or CHECK failure indicates a weak performance area.

Run performance criteria may be more complex. Selected performance areas, referred to as performance measurement variables (PMVs), are monitored during each run. Each selected PMV has associated passing and failing criteria. Again, a failure pinpoints a weak performance area.

b. Prescription and Remediation.

Once a weak performance area is diagnosed, a prescription is made based on the student's performance history. The prescription may consist of: a repeat of the TEST, CHECK, or run; a remediative segment; or instructor referral. If a student fails a remediation segment, the student is immediately referred to the instructor.

Prescription and remediation in IAT segments will be described followed by a description of prescription and remediation in CP and FP segments. The PMV Remediation Table which is pertinent to both CP and FP remediation is discussed later.

c. Interactive Teaching (IAT) Model.

Figure 2 shows the IAT segment model. Note the instruction path is sequential unless a TEST or CHECK is encountered. A TEST or CHECK is an optional feature of an IAT segment. An IAT segment may contain any combination of TESTs and/or CHECKs in any order and be interspersed among other IAT segment commands. Figure 3 shows the simple testing DPR model.

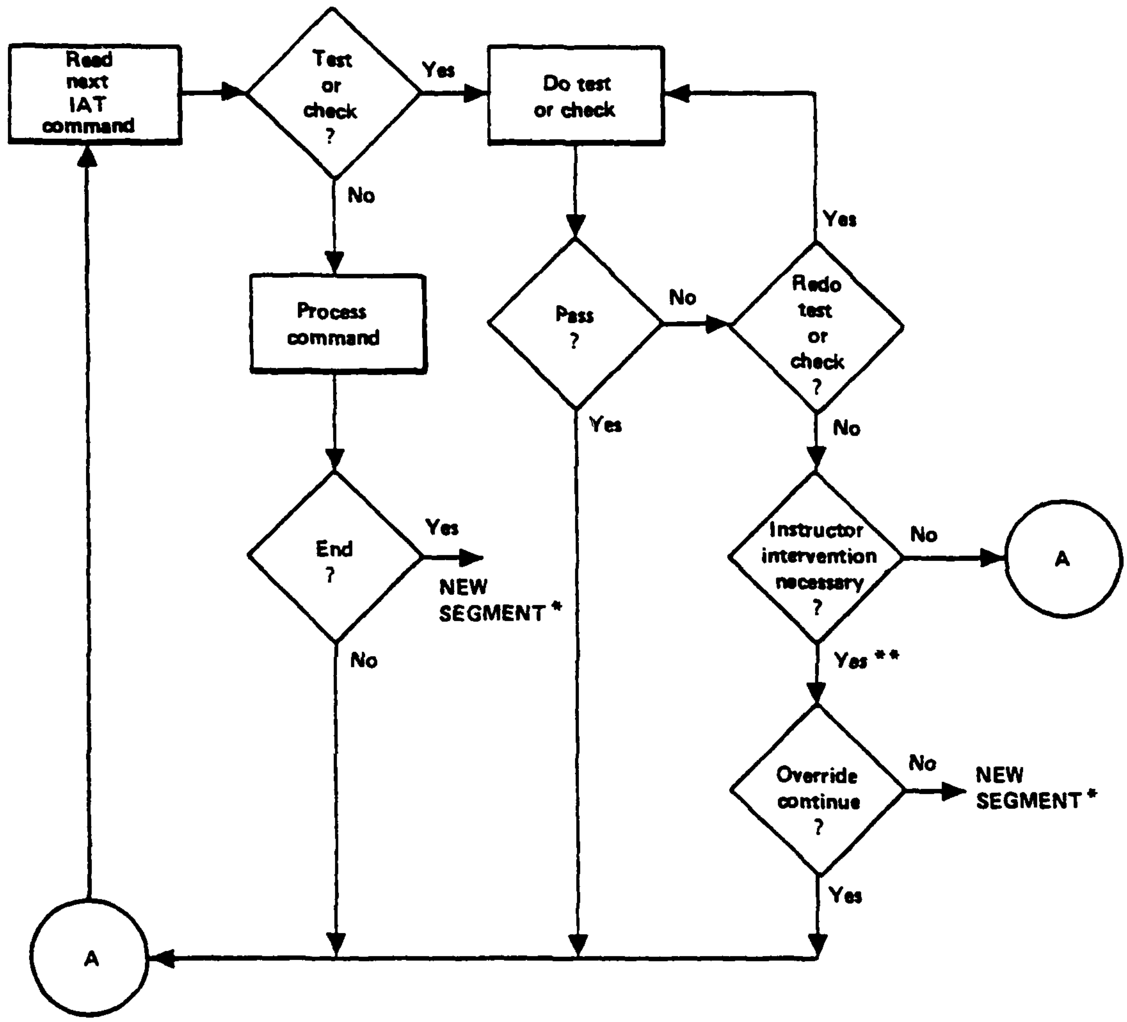
Test and check presentation in IAT segments and simple test remediation are dependent on two parameters:

(1) Second try option - This option allows the student to repeat a TEST or CHECK immediately after the first failure. This parameter is defined for each TEST and CHECK in each T/H and CK/H segment command.

(2) Instructor referral option - After a second failure of a TEST or CHECK, the student is immediately referred to the instructor if this option is selected. This parameter is defined for each T/H and CK/H for each TEST and CHECK.

d. Commented Practice (CP) Segment Model.

A CP segment may contain only one scenario. The scenario (SC) segment command which starts scenario execution may be prefaced by any segment command that has been defined as legal during CP segments. Commands will be executed sequentially until the SC command is encountered.



* Recommendation made to Control Subfunction

** At this point an automatic signoff can occur if the *instructor* has requested a system shutdown to occur after the end of the current segment.

Figure 2. Interactive Teaching Path

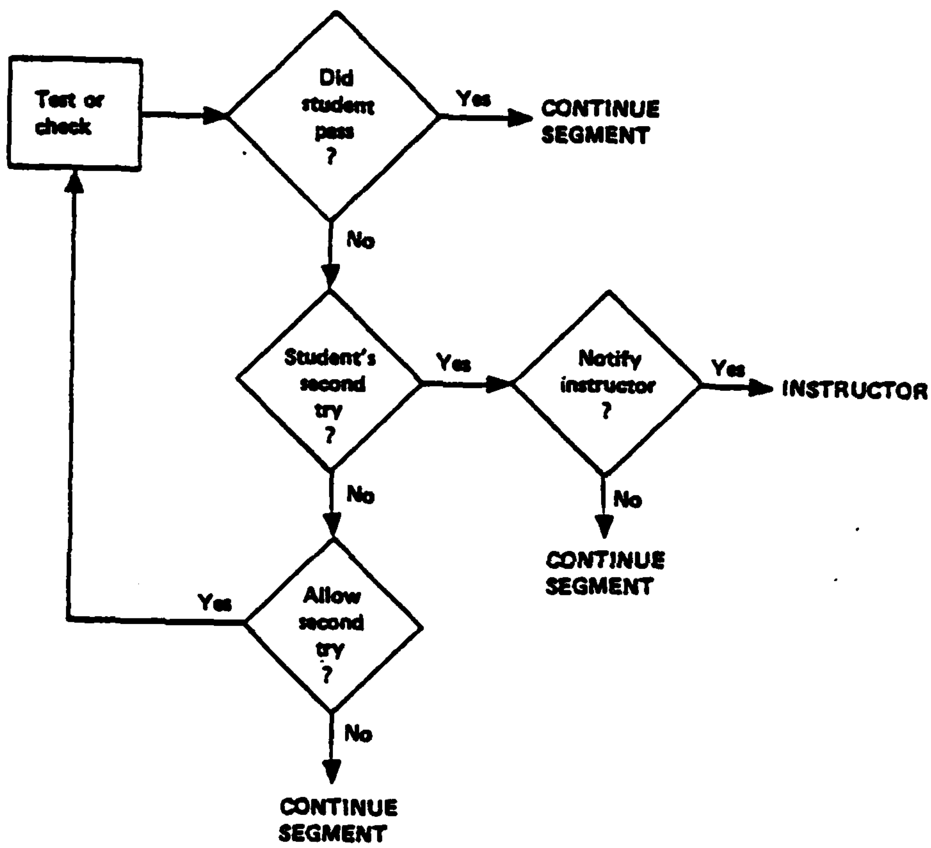


Figure 3. Simple Test Path

The SC command starts a sequence of runs based on the associated scenario file. Canned text will be presented to give the student run status, error feedback, etc., during these runs. Figure 4 illustrates the CP segment DPR model.

The sequence of runs is determined by student performance in relation to five CP parameters. Parameters (1) and (2) are defined with the PMV command; parameters (3), (4), and (5) are defined with the CP command. These commands must occur at the start of the segment's command file. Note that the total number of runs allowed per segment in Figure 4 is an ACE system variable. The parameters are defined as:

(1) "Freeze" PMVs - The student causes an error freeze to occur whenever an error belonging to the indicated PMV(s) lowers the PMV's point score such that the freeze error level is transcended. The run is frozen at that point while feedback is provided. The run is then immediately terminated. After a specified number of errors on the same PMV(s), remediation is provided (see item 3 below). The number of freezes due to a PMV is not stored cumulatively for the segment, i.e., the count is reset to zero at the start of each repeat of the segment. If the student passes the remediation segment, he is allowed to reenter the CP segment. If the student fails, he is referred to the instructor. If the student continues to cause freezes due to the remediated PMV, he is referred to the instructor after the same PMV has caused twice the number of freezes specified by item (3).

(2) "Active" PMVs - These PMVs are graded but do not generate an error freeze. Feedback is provided on these PMVs after the completion of a freeze-free run. If the number of failed PMVs exceeds the number specified in item 5, the PMVs are remediated in the order that they were taught. Note that the remediation for active PMVs occurs based on the passage or failure on that PMV for each run. No cumulative run-to-run information is used to determine remediation needs for active PMVs. If the student fails a remediation segment, he is referred to the instructor. If he passes, he is allowed to reenter the CP segment or to receive remediation for other active PMVs as required. If the student fails the same PMV twice in the same segment, the instructor is summoned. The remediation segment used for active PMV remediation is a CP type segment specified by the PMV Remediation Table.

(3) Number of error runs allowed before remediation - If the specified number of error freezes occur for any of the freeze PMVs, the student is given a remediative segment. This remediative segment is an IAT segment specified in the PMV Remediation Table or by the overriding segment specified in the Syllabus Table. If the student causes an additional equal number of error freezes after the remediative segment for any single "freeze" PMV, the student is referred to the instructor.

(4) Number of consecutive freeze-free runs required before advancement - If the student performs the specified number of runs freeze-free (no errors are made on the "freeze" PMVs), the student is allowed to advance. Any necessary active PMV remediation will be provided prior to advancement.

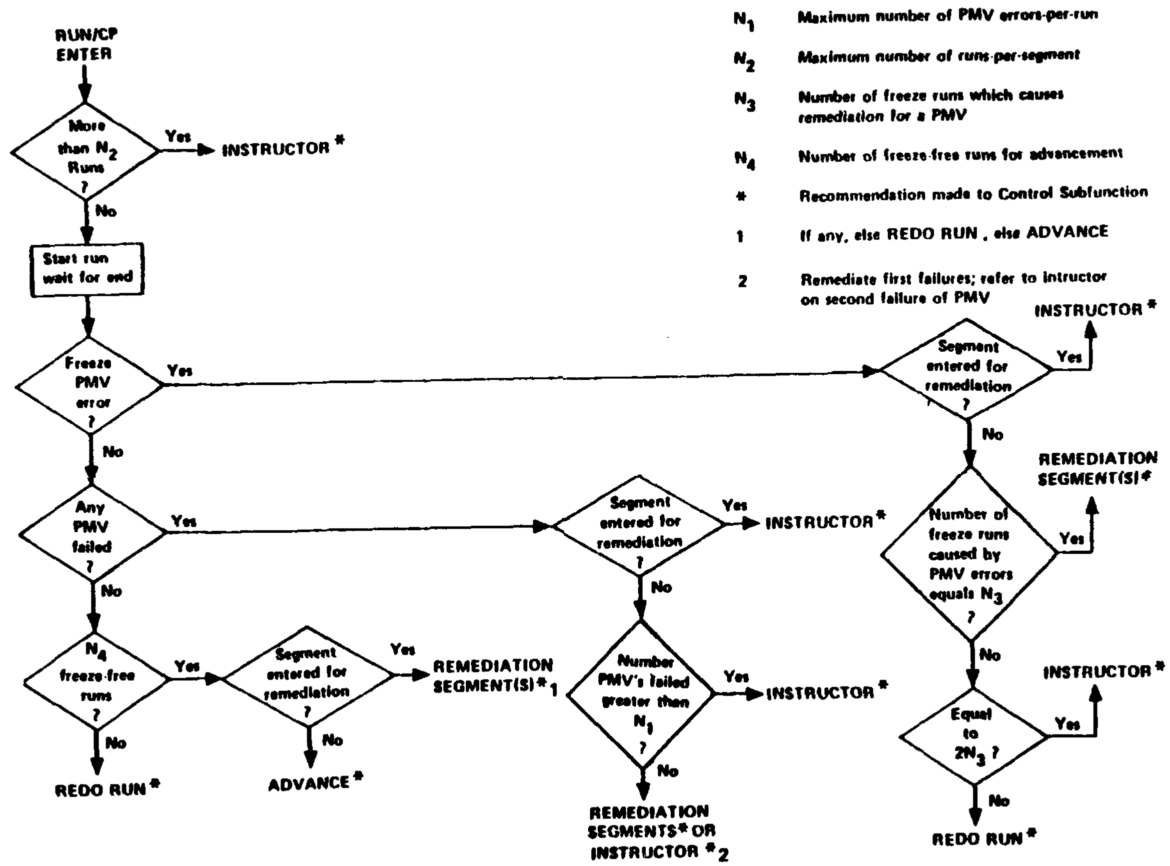


Figure 4. Commented Practice (CP) Segment, Diagnosis/Prescription/Remediation (DPR) Model

(5) Number of error PMVs for instructor intervention - If the student fails this number of PMVs in a single run, the instructor will be notified.

e. Free Practice (FP) Segment Model.

A FP segment may contain only one scenario (SC) command. This SC command may be prefaced by any segment command that is defined to be legal during FP segments. Commands will be executed sequentially until the SC command is encountered.

PMV and FP commands must occur at the start of FP segment command files.

The FP segment model is illustrated in Figure 5. Note that the total number of runs allowed per segment is an ACE system variable.

The SC command starts a sequence of runs based on the associated scenario file. Canned text will be presented to give the student run status, performance feedback, run replay capabilities, etc. during these runs.

The sequence of runs is determined by student performance in relation to three free practice parameters as well as entries in the PMV Remediation Table. The three free practice parameters are:

(1) Number of failed PMVs for instructor intervention - Maximum number of PMVs which the student may fail in any one run before instructor referral is prescribed. This maximum is defined via a FP/H command.

(2) "Active" PMVs - If the student fails more than the number of PMVs specified in (1) in a single run, he is immediately referred to the instructor. Otherwise, all failed PMVs are remediated in the order that they were taught. If the student fails a remediation segment, he is referred to the instructor. If he passes, he is allowed to continue with the FP segment or with additionally required remediation segments. If the student fails the same active PMV twice in the same segment, the instructor is summoned. The remediation provided for the active PMVs is specified as a CP type segment in the PMV Remediation Table.

(3) Advancement criteria - The advancement criteria for FP segments is a specified number of remediation-free consecutive runs. This criteria is specified via a FP/H command.

f. PMV Remediation Table.

Remediation for all PMVs is defined in the PMV Remediation Table. There is only one such table; all CP and FP segments will be subject to the entries in this table. Remediation for PMVs will be provided to the student in the order that they are to be taught. Each PMV is represented as an eighty column entry.

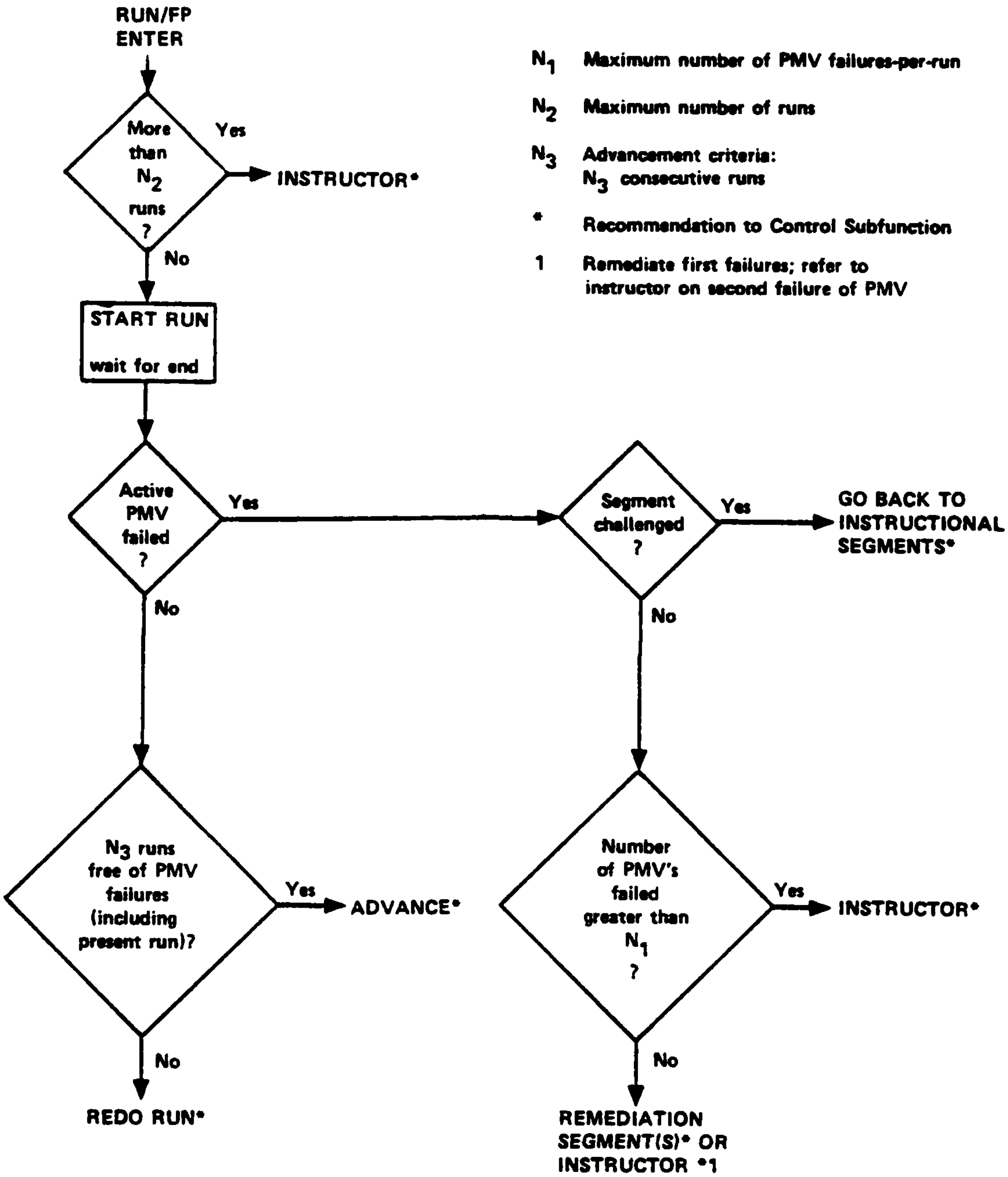


Figure 5. Free Practice (FP) Segment, Diagnosis/Prescription/Remediation (DPR) Model

The PMV Remediation Table is composed of five elements:

(1) Columns 1 - 2: Performance variable identifier - Identifier used to identify the particular performance measurement group.

(2) Columns 5 - 6: Remediation order - Order in which PMVs will be taught.

(3) Columns 9 - 11: Interactive Teaching Remediation Segment - Name of an IAT type segment which provides remediation for "freeze" PMV failures during CP segments. The criteria for "freeze" PMV remediation is defined individually for each CP segment within its command file. The IAT type remediation specified in this table is overridden by the remediation segment specified in the segment's adaptive description in the Syllabus Table.

(4) Columns 15 - 17: Commented Practice Remediation Segment - Name of a CP type segment which provided remediation for "active" PMV failures during FP segments or CP segments. If the student fails any remediative segment, the instructor is notified and any additional remediation is preempted.

(5) Columns 21 - 23: Freeze PMV Level - If the student makes enough errors such that the maximum achievable score falls below this level for the freeze PMV, a freeze will occur.

SYSTEM SUMMARY FUNCTION. This function processes all data necessary for the production and reporting of student performance data. It is activated by the Adaptive Training Control Function via the instructor console STATS key.

Input. The input to the system summary function consists of report requests and data IPCs and temporary binary data files.

Output. Output from the system summary function consists of disk text data files and report displays. The data files are structured for ease of manipulation in creating reports and summaries. These reports are generated and displayed via CRT and/or hard copy.

Function. This function provides generation of reports by taking data from the binary disk files, formatting it, and displaying the final report to the CRT or the lineprinter. These reports are designed to be viewed by the instructor. Other similar reports are generated by student summary to be viewed by the student.

Several types of reports may be requested: individual student run reports, student activity summary reports, speech recognition run reports, and system usage reports. The input data necessary for the production of these reports include student sign on time, time spent on a given segment, occurrences of challenges, overrides, summaries, number of runs per segment, segment identity, advancement criteria, failing criteria, and speech recognition and understanding data. These data are stored temporarily on disk in binary format until the system summary function can process the information.

Whenever the instructor console "STATS" key is depressed the system summary function is activated. It then proceeds to show a series of summary menus which the user can use to select which report he wishes to see. After the report has been determined, System Summary extracts data from the necessary temporary files and displays the report either on the CRT or the hard copy lineprinter.

The four categories of reports produced by the system summary function are:

a. Overall Pathway Summary Report

b. Detailed Segment Summary Report (IAT, CP, FP)

c. Speech Training Summary Report

d. Speech Recognition Summary Report

The Overall Pathway Summary Report gives a brief, one line, description of each segment completed by the student and is referenced by path number. A path number is assigned every time a student runs through an ACE segment. This includes remediated and challenged segments, as well as normal segments which are controlled through the ACE syllabus.

Since the information from this report is limited, the system provides the instructor with the Detailed Segment Summary Reports which contain more detailed data. There are three types of these reports corresponding to the three segment types (IAT, CP and FP). The IAT reports contain detailed test and check data, whereas the CP and FP reports show the results of the scenario runs.

The other two categories of reports are related to speech recognition data. The Speech Training Summary Report indicates whether or not the student is using the ACE system's speech retraining and testing capabilities. This can be useful if the student is getting poor recognition. The second speech report is for recognition only and shows the actual output from Speech Understanding, as well as the phrases recognized by the DP-100. The speech recognition report is only available for the student who is currently signed on.

STUDENT SUMMARY FUNCTION. This function is similar to the system summary function. It processes data necessary for the production and reporting of student performance data to be viewed by the student. It is activated by the Adaptive Training Control Function upon student request.

Input. The input to the student summary function consists of report requests, data IPC's and temporary binary data files.

Output. Output from the student summary function consists of disk text data files and report displays. These reports are generated and displayed via CRT.

Function. This function provides generation of student performance reports by taking data from the binary disk files, formatting it, and displaying a final report on the student station CRT.

The same input data is required to formulate these reports as is used by the system summary function. Several types of reports may be requested. They are:

a. Current Course Position

b. Last Practice Exercise Score

c. Overall Course Progress

The Current Course Position report displays the student's most recently completed non-remediation segment. If his last completed segment was a remediation segment, it will be noted on this report. The Last Practice Exercise Score report displays the results of his last free practice run. The format is identical to that of the Detailed Summary Report that is shown to the instructor. The Overall Course Progress Report displays the segments that the student has completed from the beginning of the course (in overall pathway order). Path numbers are assigned to each segment in chronological order. This includes remediated and challenged segments. This report is identical to the Overall Pathway Summary Report shown to the instructor.

SIMPLE TESTING FUNCTION. This function provides simple testing capabilities by asking questions and by requesting simple TEC operations to be performed. Responses are monitored, and the student's performance is recorded.

This function operates primarily in the Instructor Computer. The Speech Computer is used to provide generated speech and the Simulation Computer to provide interface with the TEC as required.

The simple testing function consists of two subfunctions: TEST (for asking questions) and CHECK (for observing TEC operations).

Test Simple Testing Subfunction.

Inputs. Input to the test simple testing subfunction consists of segment definition commands containing a set of test questions accompanied by test background information and student generated keyboard and TEC inputs.

Outputs. Outputs from the test simple testing subfunction consist of:

a. Correct score percentage

b. Pass or fail

Subfunction. Test subfunction consists of asking multiple choice, matching, or true/false questions which are answerable via single numerical keyboard input, true/false keyboard input, or single TEC button entry.

To conduct a test, the following processing operations are performed by the test simple testing subfunction:

a. Present test background/introductory stimuli

b. Obtain the question/request

c. Present the question/request to the student including any other necessary stimuli

d. Monitor the response time

e. Obtain response

f. Determine correctness or timeout

g. Provide feedback or timeout warning

h. Repeat steps b through g until test complete

i. At completion of test, provide test performance data for use by the adaptive training control function.

Test input data will be prepared using a preprocessor program. Test background data are required for each test specificationas well as data for each question/request presented. Test background data are comprised of:

a. Number of questions

b. Passing score percentage

c. Support stimuli needed to present the test

d. Feedback for no errors in the test

e. Feedback for passing the test

f. Feedback for failing the test a first time

g. Feedback for failing the test a second time

h. Test type (matching test or other)

i. Presentation page for a matching test

j. Whether a second try is allowable

k. Whether the instructor should be notified of a second failure.

Each question/request will consist of five components:

a. Response type (YES/NO, TEC input, or numerical keyboard input)

b. Support stimuli needed (videodisc sequence, TEC set-up, voice generation requests) to present question/request

c. Correct response

d. Feedback for correct response

e. Feedback for incorrect response

Appendix C contains the basic syntax, format, and procedures for preparing test input files.

Check Simple Testing Subfunction.

Input to the check simple testing subfunction consists of a disk file containing a list of expected TEC inputs accompanied by background information and student generated TEC inputs.

Outputs. Outputs from the check simple testing subfunction consist of:

a. Pass or fail

b. Type of error encountered

Subfunction. Check expects a series of TEC inputs to be entered. To conduct a check, the following processing operations are performed:

a. Present check background/introductory stimuli

Do steps b through e while the last expected response has not been logged or error not detected:

b. Monitor the response time

c. Obtain response

d. Log response or take timeout action if timeout

e. Evaluate input responses

f. Provide performance feedback

g. At completion of procedure, provide performance data for use by the adaptive training control function.

Check input data will be prepared using a preprocessor program. The data consists of:

a. Support stimuli needed to initiate the sequence of TEC inputs

b. Feedback for correct sequence of inputs

c. Feedback for errors of omission

d. Feedback for sequence errors

e. Feedback for errors other than c or d (or if c or d are not specified)

f. Feedback for failure on second try

g. Sequence of inputs expected in order

h. Inputs expected in any order

i. Last input expected (must be unique)

j. Timeout specification for completion of entire TEC input sequence

k. Whether a second try is allowable

l. Whether the instructor should be notified of a second failure

Appendix C contains the basic syntax, format, and procedures for preparing check input files.

PERFORMANCE MEASUREMENT AND EVALUATION (PME) FUNCTION. This function collects and evaluates data on student performance. Measurement and evaluation are determined using performance measurement variables (PMVs). Each PMV is a measure of a student's performance in executing a procedure, or an operation. The PMVs are described in Appendix D along with their associated scores and standards for scoring. PMVs are applicable for commented practice and free practice instructional segments. The desired subset of PMVs of interest to a segment are identified by appropriate segment definition commands. For free practice, each PMV of interest is associated with its passing score. For commented practice, the PMVs may be identified for scoring and/or for error freezing. When a freeze occurs a comment about the error that caused the freeze is provided.

Data for each PMV are collected and measured by a related PMV subroutine. Data for measurement are obtained from events of interest and from periodically analyzing the track data tables. When an error occurs, the PMV subroutine provides the error comment describing the error. An error comment is displayed immediately in commented practice. In free practice, the error comments are included with the evaluation data.

PME operates in the Instructor Computer and consists of the following subfunctions:

a. PME Control - provides the overall control of PME processing

b. Performance Measurement - consists of the PMV subroutines for scoring PMVs and detecting errors

c. Performance Evaluation - records an evaluation summary of the student's performance during commented and free practice segments

PME Control Subfunction.

Inputs. Inputs to the PME control subfunction consist of:

a. Desired PME mode of operation: commented practice or free practice

b. PMVs of interest

Outputs. None.

Subfunction. This subfunction provides the overall initialization and processing control of the other PME subfunctions.

Performance Measurement Subfunction.

Inputs. Inputs to the performance measurement subfunction consist of:

a. Event data

b. Track data

c. Identity of PMVs of interest

Outputs. Outputs from the performance measurement subfunction consist of:

a. The current measured scores for the PMVs of interest

b. Error comments

Subfunction. This subfunction performs the actual measurement for the PMVs of interest and reports error comments when errors are detected. Appendix D describes the various PMVs and the associated processing to be performed for each PMV's scoring and error detection. Each PMV is a measure of a student's performance in executing a procedure, or operation. Each PMV is measured by a related subroutine. This subfunction can be considered as consisting of a group of subroutines that are individually referenced by a controlling routine when data are available for them to measure.

Performance Evaluation Subfunction.

Inputs. Input to the performance evaluation subfunction consists of the current measured scores associated with the PMVs of interest.

Outputs. Output from the performance evaluation subfunction consists of an evaluation summary disk file of the student's performance during a free or commented practice segment.

Subfunction. This subfunction collects the current measured scores for the PMVs of interest and records them on a disk file.

BASIC SCENARIO CONTROL FUNCTION. During initialization, this function retrieves the time zero tagged data and event data from the prescribed scenario file. These data are used to initialize the track data files and the event table, and to condition the various models.

This function operates in the Simulation Computer and is composed of three subfunctions: Initialization, Scenario Event Processing, and Time Processing.

Initialization Subfunction.

Inputs. Input to the initialization subfunction consists of a disk file of scenario data accessible by the Simulation Computer (see Appendix B for format).

Outputs. Outputs of the initialization subfunction consist of established core tables containing formatted time driven data for time zero and an event driven commands.

Subfunction. Event commands and time commands are retrieved and tabled for rapid access. Time zero tagged data are processed and designated randomization factors applied. The data are converted, as necessary, and placed in the track data section of the shared page. Default conditions are supplied when initial conditions are not provided; e.g., proficiency models.

Scenario Event Processing Subfunction.

Inputs. Inputs to the scenario event processing subfunction consist of the numerically ordered table of events and the processing to be performed upon the occurrence of the respective events.

Outputs. Output of the scenario event processing subfunction consists of messages to trigger processing or changes to track data section of the shared page.

Subfunction. Upon notification that an event has occurred, the event table is accessed to determine matching event numbers and to cause the desired processing to be performed. Processing options consist of changing data in the track data section of the shared page to affect aircraft motion, turning tracks on/off, turning models on/off, freezing time, and/or sending messages to report situations (e.g., NTDS failure, emergency).

Time Processing Subfunction.

Inputs. Inputs to the time processing subfunction consists of the chronologically ordered table of scenario commands defining the processing to be performed upon the occurrence of the respective times.

Outputs. Outputs from the time processing subfunction consist of messages to trigger processing or changes to the track data section of the shared page.

Subfunction. The same type of processing is performed for time driven actions as for event driven actions. Time driven actions are activated when the amount of time from exercise start is reached; whereas, event driven actions are activated upon detection of an applicable event occurring.

NTDS FUNCTION. This function simulates the capabilities of NTDS that are necessary to support AIC training. This function works closely with the radar, pilot, aircraft, and tracking simulation functions.

The Simulation Computer and Micronova processors are used for this function. The NTDS simulation function is comprised of two subfunctions:

a. TEC Miconova Processing

b. TEC Simulation Computer Processing

TEC Micronova Processing Subfunction.

Inputs. Inputs to TEC Micronova processing subfunction consist of:

a. Activation of TEC buttons/switches

b. Requests received from the Simulation Computer (see Appendix B)

Outputs. Outputs from the TEC Micronova processing subfunction consist of:

a. Data transmitted to the Simulation Computer in response to button/switch actions or Simulation Computer requests (see Appendix B)

b. Control of TEC lights and LEDs (hardware)

c. Presentation of DRO and text messages (hardware)

Subfunction. This subfunction operates in the Micronova and interfaces with the Simulation Computer using an RS232, 9600 baud, full-duplex interface. The purpose of this subfunction is to monitor and maintain the status of the TEC buttons, switches, buzzer, trackball, lights, LEDs, and text presentations.

Initially, the Micronova will be bootstrap loaded by the Simulation Computer over the RS232 interface. The Micronova software and the TEC will be placed in an initial common state. Basically this will consist of setting the individual TEC components in an off or blank state, and then sequencing through the switches/controls to determine which ones are on and updating the Micronova data base accordingly. After initialization, the status of the TEC will be transmitted to the Simulation Computer. Subsequently, the Micronova will be in a button/switch interrogation loop waiting for an operator action, data available from the Simulation Computer, or a necessity to send trackball data to the Simulation Computer.

When a button/switch activation is detected, its identity and associated data will be formatted and transferred to the Simulation Computer. Only buttons/switches that are applicable are interrogated, others are considered to be non-existent.

When data are received from the Simulation Computer, the applicable processing will be performed to:

a. Control a light

b. Display a DRO or text message

c. Control the buzzer

d. Transmit status to the Simulation Computer

e. Control state of trackball updating

f. Perform designated testing

When trackball updating is active, every 50 milliseconds the trackball delta coordinates will be transmitted to the Simulation Computer unless both x and y components are zero.

TEC Simulation Computer Processing Subfunction.

Inputs. Inputs to the TEC Simulation Computer processing subfunction consist of:

a. Data transmitted from the TEC Micronova in response to button/switch actions at the TEC (see Appendix B)

b. Data transmitted in the form of IPC messages from other functions; e.g., SWC, tracker, replay

c. Data contained in the TEC section of the shared page

Outputs. Outputs from the TEC Simulation Computer processing subfunction consist of:

a. Requests to the TEC Micronova to alter the status of TEC lights, LEDs, buzzer, CRT displays, and requests for specific information (see Appendix B).

b. Updates to the TEC section of the shared page

c. Data transmitted in the form of IPC messages to other functions; e.g. SWC, tracker, replay, Instructor Computer processes

Subfunction. This subfunction simulates those input and output effects of the NTDS operational program that are described in Appendix F. These effects are a subset of the NTDS that are necessary for training AIC students. The following actions are performed by this subfunction:

a. Bootstraps the Micronova and causes initialization of the TEC.

b. Monitors asynchronous TEC button/switch action inputs from the Micronova as defined in Appendix F, and processes them in order to update the state of the NTDS simulation.

c. Requests TEC status information to determine the state of the TEC.

d. Provides the TEC Micronova processing subfunction with data for altering the external appearance of the TEC; e.g., to activate or deactivate lights, LEDs, and buzzer and to display numeric and text information on the TEC CRTs.

e. Monitors the periodic updating (every 50 milliseconds) of trackball x and y coordinate motion information from the Micronova and derives the instantaneous position of the x and y coordinate position of the ball tab.

f. Sends commands to the Megatek 7000 graphics display unit to control the display of: NTDS symbols representing aircraft tracks at their positions as known to the NTDS operational program simulation; the ball tab symbol, which reflects the motion of the track ball; fixed symbols representing geographic locations; special symbols to inidicate particular activities or events occurring at various locations; lines to indicate aircraft intercept geometry; text information concerning aircraft geometry; and text information concerning the activity state of the exercise.

g. Updates the TEC section of the shared page on a periodic basis with new aircraft positions as determined by a "dead reckoning" algorithm, and calculates new aircraft intercept geometry parameters according to them.

h. Transmits IPCs containing NTDS state information and updates the TEC state file.

i. Monitors IPCs from other functions and responds accordingly.

j. Detects a number of system events (identified in Appendix B) and button/switch action events and transmits them in the form of an IPC.

TRACKER FUNCTION. This function maintains track symbology positional data in the TEC section of the shared page and causes associated displays to be updated to reflect the changed positions. Symbol history data are also maintained. The tracker function operates in the Simulation Computer.

Inputs. Track number of symbol to be tracked.

Outputs. The tracker function will provide the following outputs:

a. Updated symbol coordinates for the track

b. Updated delta coordinates to match video coordinates when sweep passes, when appropriate

c. Updated track history data, when appropriate

d. Updated display list for the new symbol position and associated lines, if any

e. Detected system event data

Function. The Tracker function updates the TEC section of the shared page on an asynchronous basis with new aircraft positions, as determined by an algorithm simulating an external tracker (at another console) who manually repositions aircraft tracks to the positions of their radar video images; repositioning of each track symbol using this function takes place after the radar sweep passes over the radar video image of the aircraft.

History. The tracker function also updates the track history table each time that a position correction of a symbol is made. The history data is also updated if a manual TEC operation repositions a symbol; e.g., CAP tracking.

Display. As each symbol is repositioned, the display list is updated accordingly for the new position. Associated lines are also updated.

Event Data. System events, as identified in Appendix B, are detected and transmitted to the MCA router for distribution.

RADAR SIMULATION FUNCTION. This function maintains a simulated radar presentation on the TEC pertaining to a sweepline, range marks, and video. It operates in the Simulation Computer.

Inputs. Inputs to the radar simulation function consist of:

a. Sweep position (angle)

b. Video positions (range and bearing from ownship)

c. Video sizes

d. Range scale

e. Offset position

f. Intensity controls: sweep, range marks, videos, symbols

Outputs. Outputs from the radar simulation function consists of an updated display list reflecting the following possible adjustments:

a. Range scale updated

b. Offset position updated

c. Intensity of applicable elements updated

d. Sweep line displayed at its updated position

e. Applicable range marks displayed

f. Applicable videos display

Function. During initialization the display list will be established for the radar simulation. All elements of the display list will be at fixed positions for direct accessibility during updating. At the end of the radar simulation portion of the display list will be instructions for interrupt triggering the radar simultation function. That is, as soon as the current radar presentation has been displayed the radar simulation function will be scheduled to ready the list for the next presentation.

Range Scale. The new scale will be passed to the scaling hardware, and the change will be reflected in the next display list update.

Offset. The new offset x and y positions will be passed to the translation hardware and the change will be reflected in the next display list update.

Intensity changes. The new intensity will be applied to the appicable elements of the display list, and the change will be reflected in the next update.

Sweep Control. At each access, the sweep vector will be rotated one-half degree. The sweep will continue to advance whenever the environment is frozen.

Range marks. Range marks intensity will be controlled as a result of switch settings on the TEC. Range marks will be shown at two mile increments to ten miles and at ten mile increments to fifty miles.

Video Control. At each access, displayability of video will be determined. Each video's current position will be compared to the current sweep line position. If applicable, the video begins its display, which is generated by the sweep line. In addition, the position at which it is to be turned off is calculated, using the size of the video and the current position.

Videos which are presently being displayed are checked against their turn off position, and if appropriate their display is ended.

An array with an element for each video is maintained for use by the Tracker. In the event a video is turned on, its array element is turned on. The tracker will use this timing information for its symbol display and the transmission of Radar associated system events.

AIRCRAFT FUNCTION. This function maintains the dynamic motion and characteristic data related to aircraft video. It operates in the Simulation Computer.

Inputs. Track data section of the shared page.

Outputs. Updated track video position and characteristics in the track data section of the shared page.

Function. Each second, the aircraft function updates appropriate video delta x, delta y values, extrapolates the video positions to the next second, and controls CAP fuel count down.

Video Delta X, Delta Y Updates. Video delta x, delta y values are updated for active video when:

a. Speed changes

b. Turn rate changes

c. Turn initiates

Video Position Updates. Each active video position is extrapolated to the next second based on its delta x, delta y data and current position. If the video is involved in a turn, a determination of its completion is made. When it completes, the motion type is set to simple and the delta x, delta y values are computed for the next update.

Fuel Count Down. Fuel count down will be maintained for the CAP. Fuel will be depleted from 12,500 pounds at a rate of 100 pounds per 55 seconds for CAP reporting of fuel. Fuel will be depleted at 100 pounds per 60 seconds for NTDS readouts.

PILOT FUNCTION. This function simulates the actions of the CAP and MAC pilots pertaining to voice and maneuver responses to AIC instructions, voice calls for proximity and status conditions, and skill level of the pilot. It operates in the Simulation Computer.

Inputs. Inputs to the pilot function consist of:

a. Track data section of the shared page for obtaining status information

b. MCA messages containing AIC instructions

c. MCA messages containing AIC requests for data

d. MCA messages containing proximity relations with reference to the CAP

Outputs. Outputs of the pilot function consist of:

a. Updates to the track data section of the shared page reflecting actions associated with AIC instructions

b. Voice message requests due to proximity situations and responses to AIC instructions and requests

c. System event data as identified in Appendix B

Function. The pilot function has three levels of pilot proficiency:

a. Script pilot - follows the scenario only.

b. Dumb pilot - under scenario control until check in, subsequently AIC instructions are followed by interpreting heading orders by a delta of plus or minus 10 degrees and by varying the stated turn rate by plus or minus 1.5 degrees.

c. Good pilot - after check in, follows AIC orders perfectly.

The pilot function will operate as a one second periodic and in response to MCA messages (AIC instructions and requests for data) received from the Speech Computer. Appendix B contains the speech phrases that will be available for the pilot and the AIC and the digital formats of these speech phrases into MCA messages. System events, as defined in Appendix B, are detected and transmitted to the Simulation Computer EXEC for distribution.

Proximity (eyes). In reference to various CAP proximities to other aircraft (Bogey, MAC, and Stranger), a number of calls may be controlled by the scenario. These are: VISUALS, CONTACT, JUDY, TALLY HO, LOST CONTACT, and FAMISHED. A percentage of 0, 75, 90, or 100 may be selected for issuance of each call. The following processing occurs as a result of proximity situations:

a. Check in. When the CAP meets the appropriate check in condition, a check in voice message is requested to be generated "RUTH, THIS IS ...".

b. Visual. When a stranger is initially within three miles and is in front (defined as plus or minus 90 degrees of the CAP heading) of the CAP, a "VISUAL" voice message is requested to be generated.

c. Contact. When assigned bogey is initially within 15 miles and in front of the CAP, a "CONTACT" with bogey's heading and altitude voice message is requested for generation.

d. Judy. When the assigned bogey is within 10 miles and in front of the CAP, a "JUDY" voice message is requested for generation.

e. Lost Contact. After a Judy call, the voice message "LOST CONTACT" will be requested for generation when the bogey is within six miles, if the scenario generation has selected this action.

f. Tally Ho. When the assigned bogey is within three miles and in front of the CAP, the voice message "TALLY HO, FOX 1, BREAKAWAY" is requested for generation.

g. Famished. After an intercept has been initiated by the CAP (or Tally Ho or Breakaway condition) the voice message "FAMISHED" is requested for generation, if a transmission has not been received from the AIC for one minute. When under a lost contact condition, the FAMISHED time duration is reduced to 30 seconds.

h. Bogey Dope. If the CAP engages a missile (split track), and the distance beween the CAP and the missile is within 10 miles, the missile is destroyed and the voice message "BOGEY DOPE ON PLATFORM" is requested for generation.

i. Rendezvous. When a flag is set as a result of an event occurrence or time elapse from exercise start, the voice message "REQUEST RENDEZVOUS WITH c/s" is requested for generation.

Responsive (ears). AIC instructions and requests for data that are recognized are formatted into MCA messages by the Speech Computer and transmitted to the pilot simulation function for processing. The following processing is in this category:

a. Looking. When the AIC indicates that a stranger is in the area, the voice message "LOOKING" is requested for generation in response to the initial stranger concern.

b. Headings and Turns. When instructed to a new heading, the voice message "ROGER xxx" repeating the directed heading is requested for generation. The CAP's new directed heading, along with the turn rate and direction of turn when provided, are placed in the track data section of the shared page for the aircraft simulation function. Other related heading and turn data consist of:

(1) "STEADY xxx" response is requested for generation to a "STEADY" AIC verbalization, and the directed heading is set to the current heading in the track data section of the shared page.

(2) "BREAKAWAY xxx" and "CONTINUE xxx" heading responses are processed similarly to the directed heading and steady instructions. However, for a breakaway following a Tally Ho condition, an intercept outcome is determined. Determination is be based on the scenario and the voice message "SPLASH y BOGEY(s)" or "HEADS UP, y BOGEY(s)" is requested for generation.

(3) Responses for anchoring (port, starboard) and turning (ease, tighten) are also requested for generation, and the track data section of the shared page adjusted accordingly.

c. Data Requests. When status data pertaining to fuel, TACAN, or altitude are made by the AIC, the appropriate data are obtained from the track data section of the shared page, and the respective voice message requested for generation.

d. Roger. AIC verbalizations that require acknowledgement request the voice message "ROGER" be generated; e.g., "c/s ON STATION".

e. Lost Communication. When the AIC indicates lost communication intentions, the voice message "RENDEZVOUS POINT WHISKEY, ANGELS zz" is requested for generation.

BOGEY FUNCTION. This function maneuvers the bogey as defined by the scenario control commands. It operates in the Simulation Computer.

Inputs. The bogey function responds to scenario control command requests. These requests access the track data section of the shared page.

Outputs. The bogey function transmits system event data, as identified in Appendix B, and provides updates to the track data section of the shared page.

Function. The bogey function responds to the following scenario control command requests:

a. Turn on track. A one time computation of the heading toward ownship is performed and assigned to the bogey between one and ten seconds after the bogey is turned on. This randomizes the bogey's angle of attack on ownship and its reference to the CAP.

b. Jink. A bogey jink is implemented upon request, given the direction and duration. The balance of the system is notified with an MCA message. At the conclusion of the jink, the bogey is put back on a heading toward ownship.

c. Split. The bogey fires a missle directed toward ownship upon request. The track data section of the shared page is updated with a new track and other associated track characteristics.

STRANGER FUNCTION. This function flies the designated track to within three miles of the CAP to create a "stranger" condition. It operates in the Simulation Computer.

Inputs. Input to the stranger function consists of the track number of the stranger track and the track data section of the shared page.

Outputs. Outputs from the stranger function consist of:

a. The intercept heading to the CAP is set as directed heading in the stranger's entry of the track data section of the shared page

b. System event data, as described in Appendix B, is transmitted to the Simulation Systems EXEC for distribution

Function. The stranger function becomes active as the result of the execution of a scenario STRANGER command which must identify a track type that is a stranger. An intercept heading is calculated from the stranger to the CAP and placed in the stranger's entry in the track data section of the shared page. When the stranger track is within three miles of the CAP, the stranger function is set inactive, and the stranger track continues on its last intercept heading.

SWC FUNCTION. This function simulates the voice and operational actions of the SWC according to the SWC model. It operates in the Simulation Computer.

Inputs. System events and the track data section of the shared page.

Outputs. The SWC function causes voice messages to be issued from the Speech Computer and NTDS console actions (pointers, alerts, lights, etc.) to be implemented based on the SWC's operational activities.

Function. The SWC model has three levels of proficiency:

a. Passive. A passive SWC m kes no status requests or reminders to AIC to update CAP and bogey symbols, i.e., SWC function has nothing to do.

b. Fair Pesty. A fair pesty SWC tells the AIC to update CAP and bogey symbols whenever either is more than two miles off. Status reports are requested once during run out. In addition, monitoring for alert conditions and AIC-SWC voice communications is maintained.

c. Pesty. A pesty SWC tells the AIC to update CAP and bogey symbols whenever either is more than one mile off. Status reports are requested every five minutes. In addition, monitoring for alert conditions and AIC-SWC voice communications is maintained.

SWC processing is based on the following system events:

a. CAP symbol update - request symbol update when appropriate.

b. Bogey or split appeared - request appropriate alerts.

c. AIC depression of intercom button - respond with appropriate voice message.

d. AIC voice message - respond with appropriate voice message.

Generation of voice message "WHAT STATE" will be requested of the Speech Computer under the following conditions:

a. Fair Pesty SWC Model. Approximately one minute after the intercept has been initiated, and only once.

b. Pesty SWC Model. Approximately every five minutes after the scenario has been initiated.

SWC processing will not occur during rendezvous and during set ups. SWC processing detects system events, as defined in Appendix B, and transmits them to the MCA router for distribution.

SPEECH RECOGNITION AND UNDERSTANDING FUNCTION. The purpose of this function is to recognize and understand student AIC voice inputs in order to provide simulated pilot voice responses and associated flight maneuvers, and verbal responses from the SWC. Speech recognition and understanding are also necessary for performance measurement and evaluation of the student during commented practice and free practice instructional segments.

This function shall operate in the Speech Computer and shall employ an NEC DP-100 Continuous Speech Recognition System. The DP-100 system works by comparing speech input from a speaker with previously collected, or trained, samples of the speaker speaking the phrases in the vocabulary. The voice collection and validation function performs the initial voice collection operation.

The terms phrase, utterance, and message are used in this discussion. A phrase is defined as being composed of one or more English words. A phrase is a lexical item stored in the DP-100 and is the smallest unit used for speech recognition. Examples of phrases are: TWO, SPEED POINT, and MARK YOUR TACAN. An utterance is a collection of one or more phrases spoken between extended pauses; e.g., STATE 110. A message is one or more utterances comprising a complete transmission; e.g., SILVER HAWK <pause> VECTOR 180 <pause> FOR BOGEY <pause>.

Analysis has proven that by applying some syntactical knowledge to the output of the DP-100, better understanding of some recognized messages can be accomplished. The capability of the DP-100 is limited in its ability to recognize an utterance consisting of more than five phrases. Since many AIC messages contain more than five phrases (e.g., BOGEY TRACKING 207, SPEED POINT 4), concatenation of utterances into complete messages is necessary. Due to the five phrase limitation of the DP-100, the ACE AIC vocabulary includes extended pauses which requires some student training to aid in the DP-100's recognition. Some "extra" extended pauses provided by a hesitating student are removed by syntactical knowledge when concatenating a complete message.

The DP-100 does not always perfectly match speech input with its corresponding reference pattern, but instead, matches it with a "sound alike". For example, PORT and FOUR both sound very much alike to the DP-100. Utterances that are not syntactically correct are improved for "sound alike" conditions.

The speech recognition function consists of four subfunctions:

a. Speech recognition - which is the DP-100 hardware translation of utterances into a sequence of phrase numbers.

b. Syntactical handler - which syntactically improves the DP-100 outputs and concatenates utterances into messages.

c. Semantic Interpreter - which identifies and formats the messages for use.

d. Speech Recognition Recording - which time tags and records speech recognition subfunction and semantic interpreter subfunction outputs.

Speech Recognition Subfunction.

Inputs. Inputs to the speech recognition subfunction consist of:

a. Voice reference patters for all phrases in the vocabulary for the exercise

b. A list of phrase numbers to return for each phrase tht is recognized

c. Voice inputs

Outputs. Output of the speech recognition subfunction consists of a sequence of phrase numbers.

Subfunction. This subfunction assumes that the DP-100 hardware is loaded with the correct vocabulary set and that it is ready to accept voice data for translation into phrase numbers. When voice data are received, the DP-100 attempts to match the phrases with reference patterns contained in the loaded vocabulary set. When a phrase can not be recognized, a not recognized number is issued. The DP-100 outputs its data when an extended pause is encountered.

Syntactical Handler Subfunction.

Inputs. Input to the syntactical handler subfunction consists of a sequence of phrase numbers and confidence values from the DP-100.

Outputs. Outputs from the syntactical handler subfunction consist of:

a. A sequence of phrase numbers consisting of one or more utterances that has been syntactically improved, where possible, or an indication of non-recognition

b. A confidence factor of the recognition

Subfunction. The purpose of this subfunction is to find a syntactically correct explanation of the phrases recognized by the DP-100. This is accomplished by searching for phrases that can be substituted because they sound alike or can be ignored in cases where the raw phrases recognized do not fit the AIC grammar.

Along with the syntactically improved recognition, the syntactic handler subfunction will output a confidence factor which is a combination of the raw confidence values from the DP-100 and adjustments to the confidence for each improvement.

Semantic Interpreter Subfunction.

Inputs. The inputs to the semantic interpreter are:

a. A sequence of improved phrase numbers from the syntactic handler that constitute a possible message

b. A confidence factor which is used as an indication of how well the original sequence of phrases fit the explanation selected

c. State-of-the-world information from the shared page

Outputs. Outputs from the semantic interpreter consist of:

a. Requests to speech generation to speech the phrase "say again" for certain messages whose confidence factors are below a predefined threshold

b. Message formated for use by the rest of the system

Subfunction. This subfunction will use state-of-the-world information to fill in missing or unrecognized phrases and make desired improvements in recognition (e.g., correcting headings). It shall also decide whether the message should be output to the rest of the system. This decision is based on the message confidence factor and the settings of pertinent console switches (e.g., COMM2).

Speech Recognition Recording Subfunction.

Inputs. Inputs to the speech recognition recording subfunction are:

a. The sequence of phrase numbers recognized by the DP-100 for each utterance

b. The sequence of phrases that is selected by the syntactic handler for each message found

c. An indication from the semantic interpreter of the messages as they are understood and reported to the rest of the system

Outputs. The output of the speech recognition recording subfunction is a disk file of time-tagged recognitions.

Subfunction. The speech recognition recording subfunction builds and stores on disk an accumulated record of time-tagged recognitions. This information is transferred to the Instructor Computer after each exercise for use by System Summary.

SYNTHESIZED SPEECH FUNCTION. This function generates speech for the voice simulation of the CAP flight crew.

It shall operate in the Speech Computer. The implementation is centered around a voice synthesizer (a Votrax phoneme synthesizer) and is controlled by the Votrax subfunction.

Votrax Subfunction.

Inputs. Inputs to the Votrax subfunction consist of MCA messages containing one or more phrase identifiers. The MCA format and identification of the phrase phonemes are found in Appendix B.

Outputs. Outputs from the Votrax subfunction consist of the selected phrases being verbalized through the Votrax voice synthesizer.

Subfunction. The Votrax subfunction locates the indicated input phrase phonemes from a disk file, FRAZ.VO, and concatenates them into one phoneme set. This phoneme set is then provided to the Votrax system with the appropriate control data for generation of the respective voice data.

FRAZ.VO is prepared initially by using a text editor and a phrase composition program.

DIGITIZED SPEECH FUNCTION. This function generates speech for the purposes of providing:

a. Voice simulation of the MAC flight crew

b. Voice simulation of the SWC

c. Instructional messages to the student

d. Playback capability of recorded speech (REPLAY).

This function also provides the record facility for student utterances in preparation for Replay.

Playback Subfunction.

Inputs. Inputs to the digitized speech subfunction consist of MCA messages containing one or more phrase identifiers. The MCA message format and the established phrases are described in Appendix B.

Outputs. Outputs from the digitized speech subfunction consist of the selected phrases being verbalized through the digitized speech interface.

Subfunction. The digitized speech subfunction locates the indicated input phrase(s) from the disk file containing established digitized phrases for simulation or training purposes.

Each digitized speech disk file has an associated index file dictating the position and length of a given phrase in the speech file. The located digitized phrases are concatenated and supplied to the speaker system via an output channel to the digitized speech interface in order to verbalize the specified data.

Replay Subfunction.

Inputs. Inputs to the digitized speech subfunction consist of an MCA message described in Appendix B which requests the start of voice replay. The student speech files, indexing files, and replay timing files are all accessed during the operation of the subfunction.

Outputs. Outputs from digitized speech subfunction consist of the selected phrases being verbalized through the digitized speech interface and of buffered synthesized speech requests.

Subfunction. The replay subfunction steps through the time-tagged replay file using its own clock. When the clock matches the replay time, the speech request is checked for a destination of synthesized or digitized speech. For a synthesized speech destination, a message is placed in a buffer and sent to the synthesized speech function for generation. In the case of a digitized speech destination, the phrase request is forwarded to the playback subfunction.

Record Subfunction.

Inputs. Inputs are student utterances.

Outputs. Replay files consisting of the student's digitized speech file and replay timer/index file.

Subfunction. The record function is operational only in free practice. It is notified when the student is speaking. In this case, the digitized voice information is stored on disk. An entry is made in the replay timer/index file indicating time of entry, phrase position in file, and phrase length in blocks.

SYMBOLOGY

ACE will be using standard NTDS symbology for the symbols in the displays presented to the trainee. Table 1 illustrates these symbols and gives information about them.

The symbols will appear on the TEC radar display and in audiovisual presentations. They have been programmed into ROM memory in the Megatek computer as a special addition to the standard character set. Symbols will be displayed and moved on the radar display in response to requests from various simulation function programs. The initial type and position of the symbols will be determined by the scenario generation function and displayed by the NTDS model function. The NTDS model function will generate requests for symbology changes in repsonse to trainee actions (engagement, hook, etc.) and to satisfy the self-regulated aspects of the NTDS operational programs (engagment line updating, NTDS automatic tracker, etc.) The tracker model function will maintain aircraft symbology whose positioning requires changes in real time (i.e., CAP, bogey, and the air symbols).

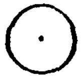
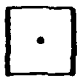
TABLE 1. ACE SYMBOLOGY

SYMBOL AND NAME	SIZE	DRO PRESENTATION WHEN HOOKED	ASSOCI-ATED LEADERS	ASSOCI-ATED VIDEO	TRACKED BY	NUMBER NEEDED	DISPLAY CONTROL	REMARKS
OWNSHIP	3/16	Participating unit number PIF number Range & bearing to ball tab	No	No	N/A	1	When NTDS is up	The point of sweep origin
CAP STATION	3/16	Station number Altitude in feet *(thousands)*	No	No	N/A	4	Scenario control	Shows a fixed geographic position
FRIENDLY AIR	3/16	Track number Heading	Yes	Yes	Tracker model	3	Scenario and/or pre-program control	Used as stranger, strike, tanker A/C and as an interim step in entering a CAP symbol
CAP ENGAGED CLEARED TO FIRE	3/16	Track number PIF number Aircraft type Data Link status Ordered heading Ordered speed Ordered altitude	Yes	Yes	Trainee's CAP, man All others tracker Auto Vector System	4	Scenario control	Responds to variable action buttons

TABLE 1. ACE SYMBOLOGY - continued

SYMBOL AND NAME	SIZE	DRO PRESENTATION WHEN HOOKED	ASSOCI-ATED LEADERS	ASSOCI-ATED VIDEO	TRACKED BY	NUMBER NEEDED	DISPLAY CONTROL	REMARKS
UNKNOWN AIR ENGAGED CLEARED TO FIRE	3/16	Track number Heading Speed Altitude	Yes	Yes	Tracker model	4	Scenario control and/or pre-program control	Represents "Bogey" aircraft
HOSTILE AIR ENGAGED CLEARED TO FIRE GIVEN	3/16	Track number Heading Speed Altitude	Yes	Yes	Tracker model	1	Scenario control and/or pre-program control	Represents "Bandit" aircraft
FRIENDLY SURFACE	3/16	Participating unit number PIP Number	No	No	N/A	1	Scenario control	Represents a Friendly Surface other than Ownship
UNKNOWN SURFACE	3/16	Track number	No	No	N/A	1	Scenario control	Represents an unknown surface unit

TABLE 1. ACE SYMBOLOGY - continued

SYMBOL AND NAME	SIZE	DRO PRESENTATION WHEN HOOKED	ASSOCI-ATED LEADERS	ASSOCI-ATED VIDEO	TRACKED BY	NUMBER NEEDED	DISPLAY CONTROL	REMARKS
 HOOK  HOOKED TRACK	3/8	N/A	N/A	N/A	N/A	1	Sequence function or ball tab	Used to place another symbol in close control, referred to as, "a symbol is hooked"
 BALL TAB	1/8	N/A	N/A	N/A	N/A	6	Ball tab center Ball tab enable Moved by track ball	Track history request uses the ball tab to display the last 5 position-correct positions of a hooked track
 POINTER	3/8	N/A	N/A	N/A	N/A	1	Scenario control or by pointer button	Used by the SWC or AIC to bring another operator's attention to a spec point
 REFERENCE POINT	3/16	N/A	N/A	N/A	N/A	4	Scenario control or operator input	Used to slave to a symbol or mark a geographical point

SYLLABUS

The Ordinal Syllabus[7] was developed to provide a curriculum outline defining instructional topics, sequences, standards, and conditions for subsequent courseware development. Implementation of the courseware development process has led to the identification and use of three types of instructional segments. The types of segments used to express the instruction are interactive teaching (IAT), commented practice (CP), and free practice (FP).

Three types of software tools are provided to aid training analysts to develop courseware: (1) segment definition translation, (2) scenario generation translation, and (3) voice collection and validation. Translation of segment definitions and scenario generations is accomplished using a preprocessing program. Voice collection and validation is integrated as a function with the ACE training system.

IAT SEGMENTS. Interactive teaching segments can be comprised of components using all the available instructional systems. Teaching and testing can be done using CRT interaction, voice generation, digitized speech, videodisc-based audiovisual presentations, and graphic display and practice on the training enhancement console (TEC).

Instruction in the IAT segments is based upon a step-by-step presentation and practice of preliminary procedures, knowledges, and psychomotor skills such as entering a CAP symbol or recognizing a jink. In IAT segments the learner approaches the individual instructional tasks with nothing else happening. Competing instructional tasks are minimized to promote concentration of the current instructional task(s).

IAT segments carefully build and integrate skill components and then test mastery of those individual skills or skill sets before the learner is allowed to proceed. Depending on the nature of the instructional task, the learner can be tested using a computer automated paper and pencil type test (i.e., multiple choice, matching, etc.) or using simple checks of student performance. In these the learner uses the TEC simulation to show mastery of a single procedure or procedure set within predefined criterion parameters such as "within 10 seconds."

CP SEGMENTS. Once the learner has mastered requisite skill sets in the task isolated IAT environment, he advances into appropriate commented practice (CP) segments. CP segments have two purposes:

a. to practice and integrate recently learned skills in a limited operational environment

b. to provide graduated practice (with an emphasis on new skill(s)) to assist the learner in more and more closely approximating the total operational job in either the tactical or the training environment.

---

7. _Ordinal Syllabus for Air Intercept Controller Prototype Training System_, Report NAVTRAEQUIPCEN 78-C-0182-3 (Logicon, Inc.). Naval Training Equipment Center, Orlando, Florida; in press.

In the commented practice segment the learner first applies skills acquired in isolation in the adjacent IAT segments. In the CP segments he will utilize the TEC as a work station. He will be given a brief simulated working task that emphasizes the newly learned skills in concert with other previously learned AIC tasks. The new skill emphasis is achieved by carefully limiting the environment in terms of scenario complexity; and stopping the scenario run, providing feedback or remediation as appropriate, and starting another scenario.

Testing for errors in commented practice segments is accomplished by a performance measurement subsystem which monitors learner performance versus both time-tagged and event-tagged standards. All measurement and scoring in CP segments is done in "real-time". Skills are required to be performed at end of course standards in order to pass a CP run.

FP SEGMENTS. Once the learner has mastered the skills up to end-of-course standards in the CP segment limited environment, he can advance to a free practice (FP) segment. The FP segments' primary purposes are to provide the learner with practice in an operating environment incorporating all the relevant skills the learner has mastered thus far in the instructional sequence; and to test the learner for course mastery on specific skills throughout the curriculum. FP segments use only the TEC for practice. Like CP segments, FP segments can measure learner performance on both time-tagged and event-tagged behaviors. Results of performance measurement and scoring are provided at the end of each run.

SEGMENT DEFINITION COURSEWARE. Segment definition commands are used to identify the sequences of displays and learner/system interactions possible within each curriculum segment. Appendix C describes the segment definition commands that are available. By utilizing these commands, the courseware designer has the capability to identify specific student inputs and alternate system responses to provide instruction, information, and practice for the learner. The set of definition commands provides a powerful software tool for defining courseware materials at a source language level. The preprocessor that translates the segment definition commands is described as the last paragraph under the syllabus topic.

A segment is the smallest unit of courseware. All segments fall into one of three categories: Interactive Teaching (IAT), Commented Practice (CP), and Free Practice (FP). Table 2 lists the functional characteristics of each segment with respect to student input, measurement, system output, and type of training.

Each segment is defined by a segment adaptive description and by a segment command file. Segment adaptive descriptions uniquely identify each segment and are used to establish the training syllabus. Segment command files contain commands which specify actions to be performed by the system during operation.

TABLE 2. SEGMENT TYPES

	INTERACTIVE TEACHING	COMMENTED PRACTICE	FREE PRACTICE
Student Input	Keyboard, TEC, Voice	TEC, Voice	TEC, Voice
Measurement	Automated pen & paper, Isolated measurement of TEC inputs	Real-time performance measurement via voice recognition and TEC input monitoring	Real-time performance measurement via voice recognition and TEC input monitoring
System Output	Audiovisual sequences, CRT text, TEC LED(s), Automated speech (Votrax, Digitized) Simulation displays (DROs and graphics)	Simulation displays (DROs and graphics) Automated Speech (Votrax, Digitized) CRT text	Simulation displays (DROs and grahics) Automated speech (Votrax, Digitized) CRT text
Type of Training	Isolated skills: Remembering (memory) and application in isolation; Overview, discussions, demonstrations	Isolated skills: Application in constricted environment and graduated practice of complex skills, Basic skill integration	Mastery level skills: Integra- of skills to end of course standards, Improvement of skill achievement levels

Segment adaptive descriptive information consists of:

a. Segment number

b. Segment type (IAT,CP,FP)

c. Segment number of next segment

d. Reviewability

e. Segment number of remediation segment

f. Segment number of challenge segment

Adaptive descriptive information for each segment will be contained as an eighty column entry in the syllabus table. The structure of the entry is given below:

a. Column 1 - 3: A unique segment number between 100 and 899

b. Column 5 - 6: 1 = IA (for IAT)
2 = CP
3 = FP

c. Column 9: F = Not reviewable, T = reviewable

d. Column 12 - 14: Segment number of segment which follows if the student successfully completes this segment or 0 if none

e. Column 17 - 19: Segment number of segment which follows if the student does not successfully complete this segment or 0 if none

f. Column 22 - 24: Segment number of segment which follows if this segment is challenged or 0 if the segment is not challenged

g. Column 28 - 29: Unit number to which the segment belongs

The sequence in which segment command files are executed is determined by the syllabus table and by student performance and choice. This is discussed further in Adaptive Training Control.

The segment command files are used to provide the learning environment described earlier. The commands used to build these files are given in Appendix C. The translated file format is given in Appendix B. The processing of the commands is described in Adaptive Training Control.

SCENARIO GENERATION COURSEWARE. Scenario generation commands are used to configure the environment simulation so as to present a broad spectrum of simulated CIC AAW team experience based on the AIC's role. Scenario commands are used to describe demonstrations or exercises. Appendix A defines the scenario generation commands that are available. Each command has a particular function which either alters the flight of an aircraft, changes the directives of the models, interrupts the running scenario, accesses specialized routines, or determines the initialization of the aircraft, TEC picture, or models. All commands are designated to trigger processing upon elapse of exercise operation time duration, or upon the occurrence of a recognized event; e.g., JUDY condition.

The preprocessor that translates scenario generation source commands into formatted binary file data is described as the last paragraph under the syllabus topic.

VOICE COLLECTION AND VALIDATION FUNCTION. This function is used during interactive teaching and as a result of activation of special function keys at the instructor and student stations. Its purpose is to collect the reference pattern for a designated phrase and to allow the trainee to practice a previously trained phrase. This function operates primarily in the Speech Computer with some control provided by the Instructor Computer.

There are four voice collection and validation subfunctions: the voice practice, the voice collection, the voice validation, and the voice test subfunctions. Normal usage in an interactive teaching segment consists of a practice, collect, and validate sequence. "Practice" allows the student to practice verbalization to gain familiarity with a vocabulary item. "Collect" allows a speech reference pattern to be stored for a vocabulary item. "Validate" allows the "collected" reference pattern for a vocabulary item to be verified. The "voice test" subfunction is used at some later time to test the system's recognition over a set of vocabulary items.

Voice Practice Subfunction.

Inputs. Inputs to the voice practice subfunction consists of:

a. The element number of the phrase or phrases to be practiced

b. The student's vocalization

Outputs. The outputs of the speech practice subfunction consist of:

a. Feedback to the student about the quality of his voicing

b. A message to Adaptive Training Control indicating student performance

Subfunction. Voice practice is used to reinforce proper speaking techniques. This is accomplished by prompting the student with a digitized voice prompt which demonstrates speech, pronunciation, and proper use of pauses. A CRT text prompt is also used to visually reinforce the digitized prompt. When the student speaks, his performance is evaluated. He will be informed if he has not met the criterion for footkey action, voice level, or proper number of utterances. The student will be reprompted until he has correctly spoken the messge twice or has erred three times in a row. Adaptive Training Control will be notified when processing is complete.

Voice Collection Subfunction.

Inputs. Inputs to the voice collection subfunction consist of:

a. The element number of the phrase or phrases to be collected

b. The student's vocalization

Outputs. The outputs of the speech collection subfunction consist of:

a. Feedback to the student about the quality of his voicing

b. A voice reference pattern for each of the phrases collected

c. A message to Adaptive Training Control indicating whether or not the phrases were collected successfully

Subfunction. Voice collection can be activated through a student generated request to retrain a phrase, or as the result of a segment definition command. The mode of invocation has no functional effect. The dialogue with the student is identical to that in voice collection. The student is prompted with the digitized voice prompt and a CRT text prompt. The voicing is evaluated for proper footkey action, voice level, and number of utterances collected. If the voicing meets the critera of acceptance, the voice reference patterns for the collectable phrases will be saved on disk for later use. If the voicing does not meet the criteria, the student will be reprompted and re-evaluated. Collection continues until two copies of each phrase have been collected or the student errs three times in a row. Adaptive Training Control is notified when processing is complete.

Voice Validation Subfunction.

Inputs. Inputs to the voice validation subfunction consist of:

a. The element number of the phrase or phrases to be validated

b. The student's vocalizations

c. The voice reference patterns of the phrases to be validated

Outputs. Output of the voice validation subfunction will consist of:

a. Feedback to the student about how well he is being recognized

b. A message to Adaptive Training Control indicating student's performance

Subfunction. The purpose of this subfunction is to verify that ACE can recognize the student's speech inputs using previously collected voice reference patterns. The student is prompted with a CRT text prompt. His vocalization is checked for proper voice level and footkey action. If he meets the criterion for each of those areas, the DP-100's interpretation of the vocalization is compared with the expected phrases. If the phrases match, the student is shown that he was properly recognized. If the phrases recognized do not match the expected phrases, the phrases that were recognized are displayed, and the student is reprompted. If the student cannot be recognized, he will be advised to retrain the unrecognizable phrases.

Voice Test Subfunction.

Inputs. Inputs to the voice test subfunction consist of:

a. The voice reference patterns for the present level of training

b. Student voice inputs

Outputs. Outputs of the voice test subfunction consist of a display of the phrases that are recognized or an indication that the incoming speech could not be recognized.

Subfunction. The purpose of the voice test subfunction is to provide the student with feedback about how well ACE is recognizing him, to help the student learn to speak in a way that optimizes recognition, and to identify poorly collected phrases so that they can be retrained.

This subfunction may be activated by the student via the student special function key INIT VOICE TEST. Processing continues until student requests the voice test to be ended.

During voice test the student may speak any phrase in the current vocabulary, without regard to normally applied rules of AIC grammar or ACE stylization constraints. Recognitions will not be improved as they are in a free or commented practice.

SEGMENT DEFINITION AND SCENARIO GENERATION PREPROCESSOR. A separate preprocessing program is used to translate segment definition source commands or scenario generation source commands into their prescribed binary formats. Appendixes A and C define the source commands for scenario generation and segment definition, respectively. The translated binary formats are presented in paragraphs of Appendix B. The preprocessor will normally operate in the Instructor Computer; however, one of the other processors may be used if a system console is available.

General Syntax Rules. The following basic rules apply to the syntax of the commands:

a. Both scenario generation and segment definition source command arguments are separated with spaces

b. Commands may be preceded by spaces and tabs

c. A "C" followed by a space as the first two entries of a line indicates a comment line

d. Each command is ended by a new line key

Translation Processes. The following kinds of ASCII source to formatted binary translation occurs:

a. Names to integer identifiers

b. Strings to strings (reformatting only)

c. Numbers to binary coded decimal numbers (4 bits each)

d. Numbers to integers or real numbers

Disk File Definitions.

Input Files. The following kinds of input disk files are used by the preprocessor:

a. Segment definition source commands

b. Scenario generation source commands

c. Scenario generation (package) source commands

d. Scenario generation (background) source commands

Input can consist of either a file of segment definition source commands or a file of scenario generation source commands. Multiple files of package and background source commands may be selected for inclusion as long as they do not cause conflicts; e.g., two tracks with the same number.

Output Files. The following kinds of output disk files are created by the preprocessor:

a. Segment definition translated ASCII File composed of variable word units

b. Two scenario generation translated ASCII files composed of 10 word units, one file containing time driven commands and the other even driven commands

c. Printable file of source versus translated data

d. Error file of detected syntax, sequence, and omission errors

Sorting. The translated scenario generation data are sorted in ascending numerical order based on the time and event word. Events are translated as negative numbers. Initialization commands are defined as those with a time of zero.

Error Checking. Each source command will be verified that it has the correct number of associated arguments and that applicable argument values are within their respective allowable ranges. If a syntax error is detected, an error message identifying the error is output to an error recording file.

A separate error pass is made on the translated data (after sorting of scenario generation data) to detect sequence and omission errors. Errors are recorded in the error file.

To aid in identifying the location of specific commands in error, command line numbers and the command line will be supplied with a description of the error. Sequence errors and omission errors will be considered as global errors whereby the error descriptions should adequately aid in locating the error. The following segment errors will be reported:

1. END COMMAND IS MISPLACED IN FILE
2. END COMMAND IS MISSING
3. TRANSLATION ERROR...INVALID COMMAND
4. HEADER COMMAND IS MISSING
5. COMMAND REQUIRES ARGUMENTS ...# OF ARGUMENTS SHOULD BE ->
6. COMMAND SHOULD NOT HAVE ARGUMENTS
7. ARGUMENT IS NOT AN INTEGER...ARGUMENT ->
8. ARGUMENT IS OUT OF RANGE...ARGUMENT ->
9. INCORRECT NUMBER OF ARGUMENTS...SHOULD BE ->

10. PMV/A ERROR -> ARGUMENTS ARE NOT IN PAIRS
11. TRANSLATION ERROR...INVALID ARGUMENT ->
12. INVALID USE OF 'CK/O CHECKED', DOES NOT FOLLOW PROPER COMMAND
13. WARNING <--- CRT/B ERROR...STRING IS TOO LONG
14. SPEECH EVENT NUMBER IS NOT VALID FOR PHRASE TYPE
15. NED ERROR -> TOO MANY DIGITS IN ARGUMENT (MAXIMUM OF 5 IS ALLOWED)
16. COMMAND IS OUT OF SEQUENCE FOR TEST/CHECK BLOCK
17. TEST/CHECK STEP WAS NOT DEFINED...STEP #
18. COMMAND PREVIOUS TO THE 'T/E' OR 'CK/E' WAS NOT A STEP
19. NUMBER OF TEST QUESTIONS IS INCORRECTLY STATED IN 'T/H'
20. COMMAND IS NOT VALID WITHIN A TEST/CHECK BLOCK
21. EITHER A 'T/E' OR 'CK/E' COMMAND IS MISSING
22. NO'CK/?' COMMANDS ALLOWED WITHIN A TEST BLOCK
23. NO 'T/?' COMMANDS ALLOWED WITHIN A CHECK BLOCK
24. COMMAND IS NOT VALID FOR THIS SEGMENT
25. COMMAND OUTSIDE OF TEST/CHECK BLOCK 'T' OR 'CK' IS MISSING
26. HEADER COMMAND IS MISPLACED IN THE FILE
27. WARNING <--- SAME COMMAND USED TWICE IN A ROW
28. STEP WAS DEFINED BEFORE THE ADV OR CHAL COMMAND
29. 'R' STEP NUMBERS ARE NOT VALID
30. PMV COMMAND IS OUT OF SEQUENCE
31. TOO MANY RUN/? COMMANDS
32. STEP NUMBER WAS DEFINED TWICE...STEP #
33. STEP WAS USED IN 'ADV' OR 'CHAL' AND NOT DEFINED...STEP #
34. TOO MANY 'SC' COMMANDS
35. 'SC' COMMAND HAS NOT YET BEEN DEFINED
36. INAPPROPRIATE USE OF RUN/S COMMAND, 'SC' COMMAND IS MIOSSING
37. INVALID SEGMENT TYPE USED
38. W/F ERROR -> NOT ALLOWED TO USE VABs OR FABs
39. MAJOR ERROR <--- INVALID CONTROL CHARACTER IN CRT/B STRING
40. CK/A COMMAND IS MISSING
41. THE LAST CK/A COMMAND IS NOT FOLLOWED BY A STEP COMMAND
42. THE LAST T/Q COMMAND IS NOT FOLLOWED BY A STEP COMMAND

The following scenario errors will be reported:

1. TRANSLATOR ERROR, INVALID COMMAND
2. COMMAND HAS NO ARGUMENTS
3. ARGUMENT IS NOT AN INTEGER...ARG:
4. INVALID NUMBER OF ARGUMENTS...SHOULD BE:
5. ARGUMENT IS OUT OF RANGE OR INVALID...ARG:
6. PACKAGE OR BACKGROUND NUMBER IS INVALID
7. PIF IS NOT ASSIGNED TO A 'CAP' TRACK TYPE (# 'S 2, 11 OR 12)
8. TRACK WAS USED WITH WRONG TRACK TYPE...TRACK:
9. TRACK TYPE OF 1 IS NOT VALID (OWNSHIP)
10. TRACK WAS REFERENCED BUT IS NOT INITIALIZED...TRACK:
11. -INITPOS- COMMAND IS MISSING FOR TRACK
12. -INITCOND- COMMAND IS MISSING FOR TRACK
13. TRACK WAS INITIALIZED MORE THAN ONCE...TRACK:
14. NO -ABORT- CONDITION IDENTIFIED
15. -ENGAGED- COMMAND WAS USED WITH UNINITIALIZED TRACK
16. TOO MANY -STRANGER- COMMANDS

17. -STRANGER- COMMAND IS MISSING
18. -END- COMMAND IS MISPLACED IN FILE
19. LAST COMMAND IS NOT AN -END-
20. NO ERRORS IN THIS FILE
21. BLANK LINE WAS FOUND
22. FIRST RECORD IN PACKAGE FILE MUST BE THE # OF COMMANDS
23. MOTION TYPE IS NOT COMPATIBLE WITH DIRECTED HEADING

SECTION III

SYSTEM DESIGN

As discussed in previous sections, ACE is basically a process control system that monitors and controls an environment external to itself, based on both internal and external inputs. As with any system of this type, ACE is a complex system in every sense. In following basic design principles, ACE has been designed from the broadest functional level down to the most detailed, then the interactions of these detailed functions have been carefully thought out and graphically depicted. Rigorously performing the above is what is termed "system design" and when properly done allows the designer to have a clear and simplified view of how the system works at its most fundamental level. This section describes how the basic components (including the users) relate at the levels of:

a. peripherals - devices attached to the system

b. man-machine interaction - a view of user orientation

c. intersystem interaction - how the various system parts work together

d. I/O requirements - timing and nature of input and output to-and-from the system.

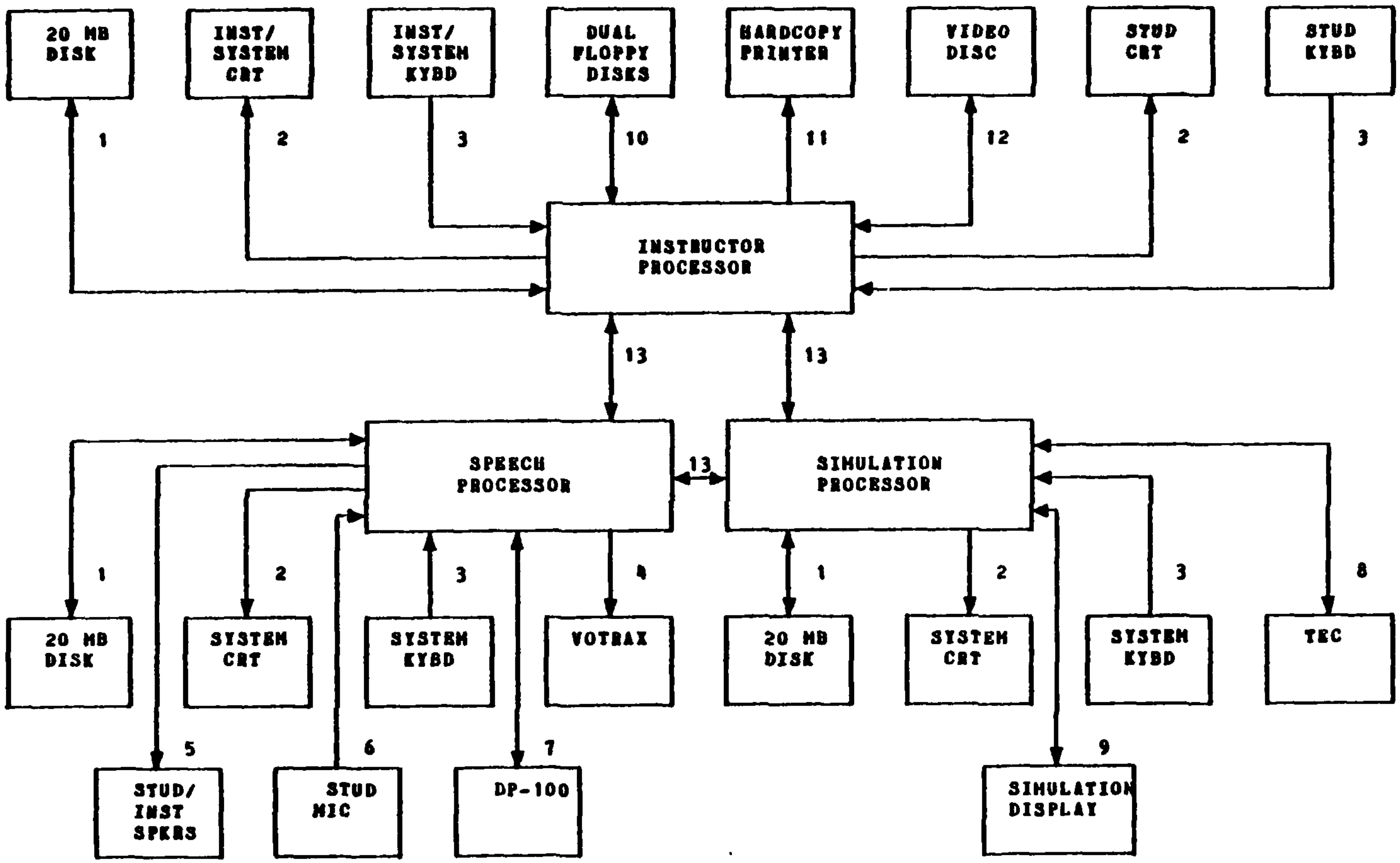
PERIPHERALS

A block diagram showing the relationship of the ACE equipment and processors is in Figure 6. The flow between the processors and equipment is identified by paths 1 through 14 in the diagram. Terminology and data paths are explained further in I/O requirements.

MAN-MACHINE INTERACTION

The student and the instructor will interact with ACE using a CRT display and a standard ASCII keyboard with a number of additional keys. These additional keys include a numeric keypad, cursor controls, and user defined, special function keys. The instructor and student keyboards will be similar, but not identical, in appearance. No keys will be in conflict between the two keyboards, but one keyboard may have a blank key where the other has a function key. Some keyboard functions will not be allowed to the student, and some will not be meaningful to the instructor.

The ACE keyboards will be used in two modes of operation. Mode 1 operates when a student has signed onto the system; an instructor may be present at the instructor station. In mode 1, only the student key subset will function on the student keyboard and the instructor key subset will function on the instructor keyboard. Mode 2 operates when a student has signed on and an instructor at the instructor keyboard has enabled the instructor key subset on the student keyboard, in addition to the student key subset. Figures 7 and 8 present the layout of the student and instuctor keyboards, respectively.



"STUD" indicates student station.
"INST" indicates instructor station.
"MIC" indicates voice input microphone.
"SPKRS" indicates an audio output speaker and earphone.
"KYBD" indicates keyboard attached to CRT.

Figure 6. Data Flow Among ACE Peripherals

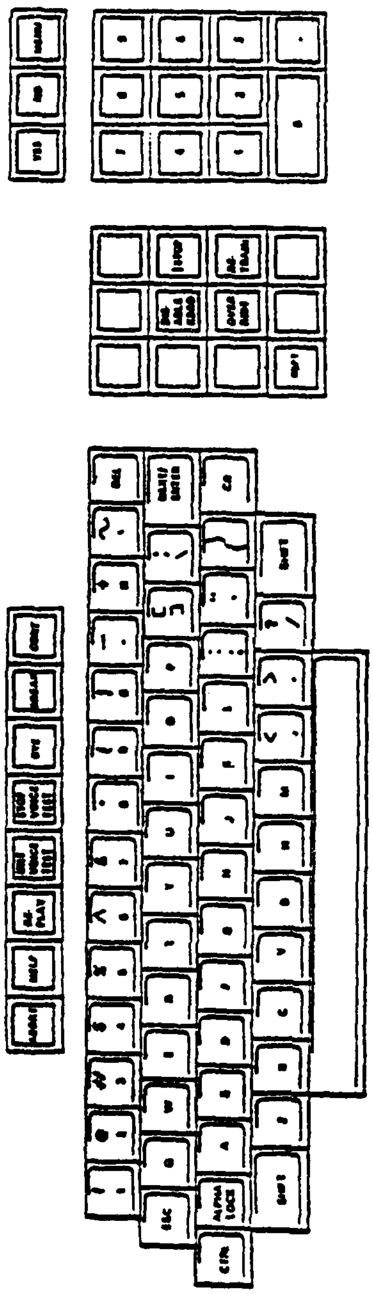


Figure 7. Student Station Keyboard

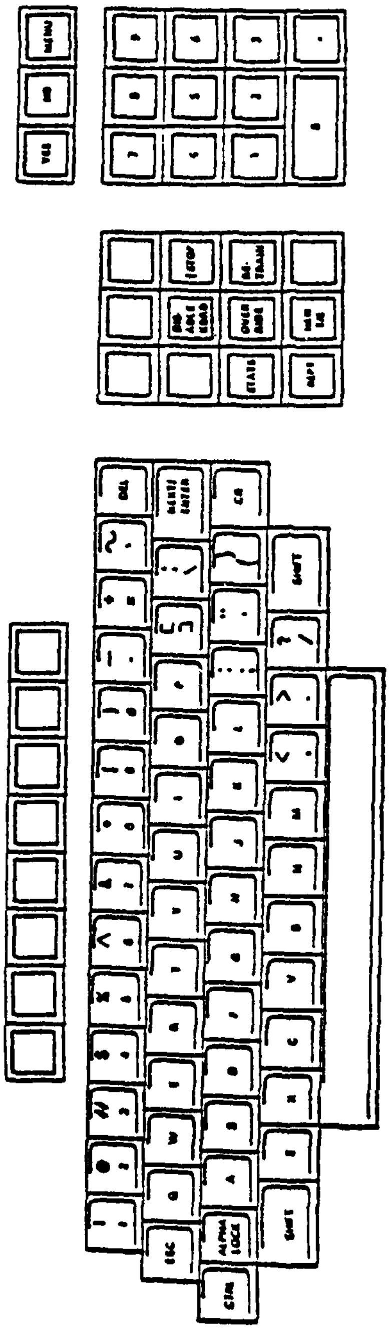


Figure 8. Instructor Station Keyboard

Unless otherwise noted, all special function keys act in the following manner. When a key is pressed, the title of the function that it calls appears in a single command line on the terminal's CRT. The contents of the command line are not acted upon by the ACE system until "entered" (that is, ENTER (NEXT/ENTER)) is pressed. Until that is done, the blinking cursor follows the last character in the command line to indicate that the command has not been accepted (and to indicate where the next character in the command line will be placed if another key is pressed). If an invalid special function is entered, "INVALID ACTION" will appear in the command line followed by a list of valid special function keys.

BOTH KEYBOARDS. Both the student and the instructor keyboards will have the following keys enabled:

a. The standard ASCII printed character keys (A-Z, a-z, 0-9, and punctuation). These are used to put single characters into the command line on the terminal's CRT.

b. The SHIFT key.

c. The DEL (delete) key, which will operate contrary to normal operation in that it will cause the entire command line to be deleted.

d. YES and NO special function keys will cause "YES" and "NO," respectively, to appear in the command line. They are used to input answers to questions posed by ACE.

e. MENU key will cause a list of the currently available special functions to be displayed on the CRT of the terminal at which it was pressed. The MENU display does not appear on the other terminal's CRT.

f. NEXT/ENTER key has two functions. The NEXT function serves to call up the next page of a sequence of text pages to be displayed on the CRT. The ENTER function causes the command input line to be transmitted to the ACE system. Commands in the command line do not take effect until the NEXT/ENTER key has been pressed. The action of the NEXT/ENTER key is immediate. This key replaces the NEW LINE key on the standard terminal.

g. RETRAIN function key will allow a special speech data collection mode of operation to be scheduled after the current segment or run is completed. This mode of operation will collect speech reference patterns for a previously trained phrase. The instructor selects the phrase to be sampled. When speech data collection is complete, the instructor is given the option to collect more data, enter the speech validation mode of operation, or continue in the syllabus. See Appendix E for menus.

INSTRUCTOR KEY FUNCTIONS. Six keys will function only for the instructor:

a. OVERRIDE function key will allow the instructor to overrule ACE's placement of the student within the syllabus. An override menu (see Appendix E) will appear on the CRT that will contain such options as to advance the trainee to the next scheduled segment, or to take him back to any previously encountered segment. If a previous segment is selected, the trainee will return to his former (pre-OVERRIDE) position in the syllabus after he completes the selected segment. The student's activity is frozen until after the selection of an override menu option.

b. ENABLE KBRD function key will allow the instructor to activate the instructor key subset on the student keyboard (to activate Mode 2). Additionally, any system initiated messages that would normally be sent only to the instructor station CRT will be sent to both CRTs when this function is operating. This key will be blank on the student keyboard.

c. Disable KBRD function key deactivates the instructor key subset on the student station keyboard.

d. (shift up) ^STOP function key will allow the ACE system to be terminated and returned to the Command Line Interpreter (CLI) of the operating system. If a student is using the other terminal, the system will notify the instructor and request resolution of the command for immediate execution or to occur after the student completes the current segment. This key will be blank on the student keyboard.

e. STATS function key will allow a menu to appear on the instuctor's CRT, from which he may select displays or printouts of statistical data pertaining to the individual student, the class, or the syllabus. The student is not disturbed when this function is accessed. The student has available a similar function which he may select from the ABORT menu pertaining to his own statistics. See Appendix E for menus. This key will be blank on the student keyboard.

f. NEW T/E function key will allow the instructor to register a new trained student. The instructor must answer a series of questions pertaining to the student before the student is fully registered. See Appendix E. This key will be blank on the student keyboard.

STUDENT KEYBOARD. Eight more special function keys are available on the student station keyboard that are not labelled on the instructor keyboard. They are:

a. ABORT function key will allow the execution of any segment or mode of operation to be aborted. After an abort, the student's position within the syllabus will be at the beginning of the current normally scheduled segment. After an ABORT has been entered, a special ABORT menu will appear from which the user may select the following options: review, continue instruction, sign off, or display statistics. See Appendix E.

b. HELP function key will allow the student to signal to the instructor that his assistance is requested. A message such as "Your student has asked for help. Please go to the student station." will appear on the instructor station CRT, heralded by a "beep" from the terminal. It will also cause the instruction or exercise to "freeze."

c. REPLAY function key will allow a special replay to be scheduled following the completion of the current segment or scenario. The replay will consist of an audio playback of the dialogue of the student's most recent practice exercise. It will be possible to "freeze" this replay.

d. BREAK function key will allow the action of the ACE system to be suspended (frozen). It may be used by the student or the instructor to force a pause during any segment.

e. CONT function key will allow action to continue that had been previously frozen using the BREAK key.

f. INIT VOICE TEST function key will allow a special speech validation mode of operation to be scheduled for execution after the current segment or run completes. During voice validation the accuracy of ACE's phrase recognition will be tested by the system echoing the student's spoken words. See Appendix E.

g. STOP VOICE TEST function key will allow termination of the speech validation mode of operation to be accomplished.

h. BYE function key will schedule a student sign off at the end of the current segment.

MISCELLANEOUS. Several points are noted pertaining to operation of the standard keyboard. The BREAK key will be hardware disabled on the ACE keyboards. Its position will be occupied with a blank key. The carriage return key will be disabled by software and will not function while the ACE program is operating. The AOS interrupt functions accessible through the CTRL-C CTRL-A, CTRL-C CTRL-B, and CTRL-C CTRL-E sequences normally will be disabled.

Menu selection will be accomplished by indicating in the command input line the number or letters corresponding to the desired selection, and then using the ENTER key to transmit this selection to ACE.

Experimenter functions may be enabled and disabled by entering SHIFT-YES and SHIFT-NO, respectively, from the instructor station keyboard.

Table 3 summarizes the ACE special function keys relating their applicability to instructor keyboard (I), student keyboard (S), and student keyboard with instructor functions enabled (I-S). The octal codes output as a result of pressing the keys are also included.

TABLE 3. ACE SPECIAL FUNCTION KEYS

KEY	FUNCTION	OUTPUT CODE	WHERE FUNCTIONAL
ABORT	Aborts the current segment. Displays abort menu. Instruction resumes at beginning of current normally selected segment.	036, 161	S,I-S
HELP	Sends request for assistance to instructor station terminal. Initiates BREAK action.	036, 162	S,I-S
REPLAY	Schedules a special replay mode of operation to follow current segment. Replay is of student's latest free practice exercise.	036, 163	S,I-S
BREAK	Suspends interactive teaching instruction, free and commented practice, and replay execution.	036, 167	S,I-S
CONT	Cancels action of BREAK key. ACE resumes instruction of exercise execution.	036, 170	S,I-S
YES	Places "YES" in command input line.	036, 171	I,S,I-S
SHIFT_YES	Enables instructor functions.	036, 151	I
NO	Places "NO" in command input line.	036, 172	I
SHIFT_NO	Disables instructor functions.	036, 152	I
MENU	Causes a list of valid special function operations to replace current CRT display.	036, 173	I,S,I-S
NEXT/ ENTER	Sends the contents of the command input line to the ACE system. Also causes the next page of a sequence to be displayed. Replaces NEW LINE key.	012	I,S,I-S
DEL	Deletes an entire command input line.	127	I,S,I-S
OVERRIDE	Overrules ACE's segment scheduling. Displays menu.	010	I,I-S
ENABLE KBRD	Enables instructor function keys on student keyboard.	036, 022	I
DISABLE KBRD	Disables instructor function keys on student keyboard.	027	I,I-S
^STOP	Prepares ACE for system power down. ACE asks for confirmation. Key responds only in shift-up mode.	036, 001	I

TABLE 3. ACE SPECIAL FUNCTION KEYS - continued

KEY	FUNCTION	OUTPUT CODE	WHERE FUNCTIONAL
INIT VOICE TEST	Causes ACE to schedule a special speech recognition validation mode of operation to follow current segment or run.	036, 164	S,I-S
STOP VOICE TEST	Terminates special speech recognition validation mode of operation.	036, 165	S,I-S
RETRAIN	Causes ACE to schedule a special speech mode of operation for collection of speech data to follow the current segment or run.	030	I,S,I-s
STATS	Displays a selection menu from which instructors can select statistics for display or printouts.	031	
BYE	Signs student off after current segment completion.	036, 166	s,I-S
NEW T/E	Registers a new student for ACE.	032	I

INTERSYSTEM INTERFACE

Figure 9 relates the flow and kind of data that is communicated among the three ACE processors over the Multiprocessor Communications Adapter (MCA) interfaces. Data are transferred over the MCAs between the AOS systems in packets consisting of a standard 10 byte header and a variable length data buffer. The header contains destination information, sender and receiver CPU number, length of the data buffer, and a time tag. MCA messages between the AOS and RDOS computers are of a standard 600 byte length and are constructed with the same header information preceding the data.

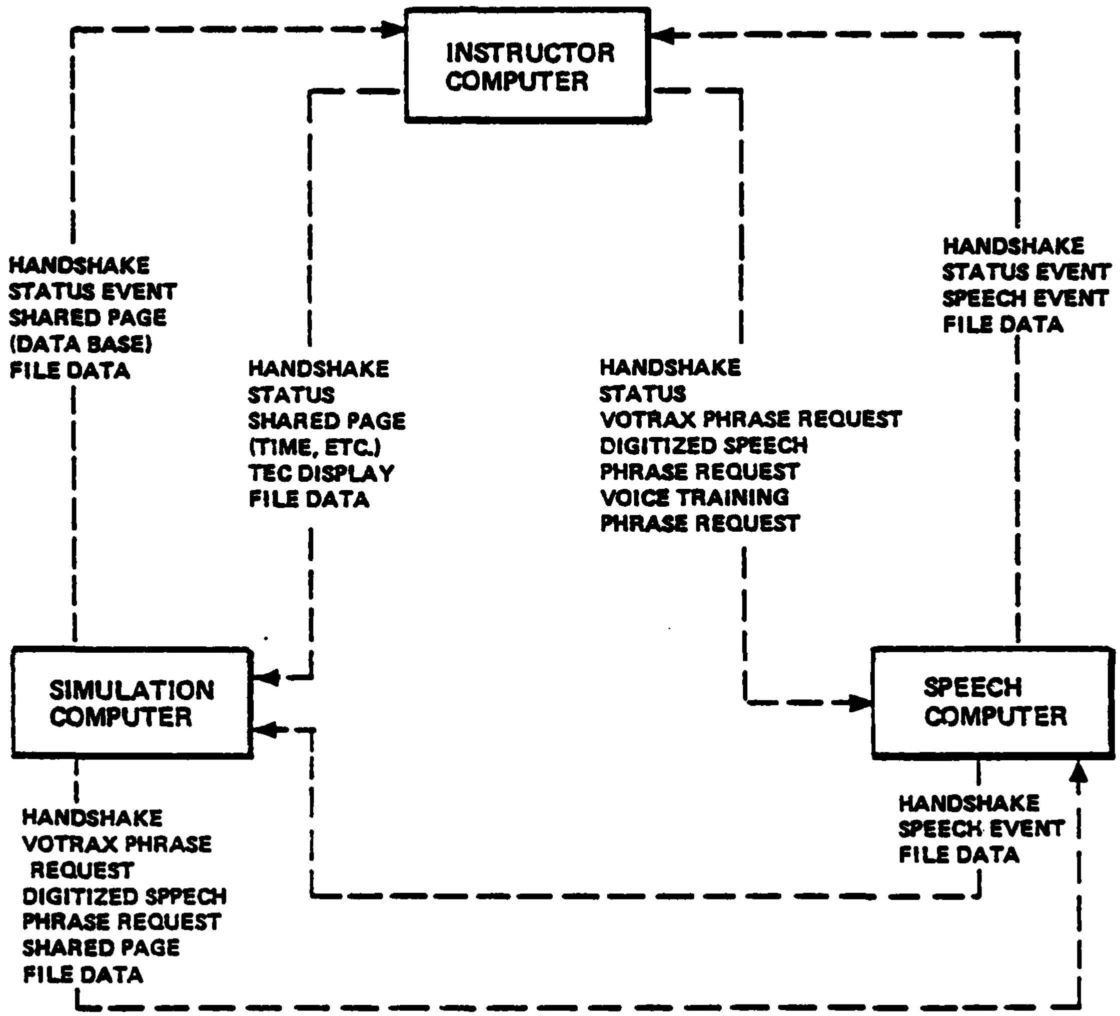


Figure 9. MCA Data Flow

I/O REQUIREMENTS

All of ACE functions are accomplished through input or output operations which are collectively termed "I/O." Due to its size and complexity, the timing and nature of these operations is a system resource that must be carefully defined and rigorously monitored. Table 4 summarizes the I/O operations for the major components of the ACE. The table indicates the processor with which the device exchanges data, the device type, data path (as identified in Figure 6), data path type, and data path speed.

TABLE 4. I/O REQUIREMENTS

PATH	PROCESSOR	DEVICE	DT	DATA PATH	DPT	SPEED
1	SP,SM,I	20 MB Disk	su	CPU buss	p	high
2	SP,SM,I	CRT	r	RS-232	s	slow
3	SP,SM,I	Keyboard	s	RS-232	s	slow
4	SP	Votrax	r	RS-232	s	slow
5	SP	STUD earphone	r	wire cable	a	--
		INST speaker	r	wire cable	a	--
6	SP	STUD microphone	s	wire cable	a	--
7	SP	DP-100	s/r	RS-232	s	slow
8	SM	SM display	s/r	CPU buss	p	high
9	SM	TEC	s/r	RS-232	s	slow
10	I	Dual floppy disks	su	CPU buss	p	high
11	I	Hardcopy printer	r	CPU buss	s	slow
12	I	Videodisc	s/r	RS-232	s	slow
13	SP,SM,I	SP,SM,I	s/r	MCA	p	high

DEVICE names used in Table 4 are for the most part self-explanatory or have been defined previously. The names which require further definition follow.

a. processor - ACE requires the services of three processors: the speech (SP), simulation (SM), and instructor (I) processors. Each processor consists of an Eclipse S/130 CPU, memory (96K in the speech processor and 256K in the SM and I processors), real-time clock, various peripheral controllers, memory address mapping and error checking hardware, floating point hardware, and battery backup.

b. TEC - The training enhancement console (TEC) represents the hardware console mockup of the UYA-4/V10 and the micro-Nova microprocessor which handles the transfer of data to-and-from the TEC. The simulation display is treated as a separate entity for the purposes of this discussion.

c. student station (STUD) earphone - The student station earphone serves as an audio output device for all recorded digitized voice and Votrax output. These modes are processor controlled, i.e., left channel, right channel, and channel select are manually selected via the communications panel but are implemented by the SP processor.

d. instructor station (INST) speaker - The instructor station speaker allows the instructor to hear all digitized voice, Votrax, and student speech output.

e. student station (STUD) microphone - The student station microphone provides voice input to the NEC DP-100 speech recognition device as well as input for the voice recording capability of the SP processor.

DEVICE TYPE (DT) is categorized as follows:

a. receiver (r) - ACE controls a receiver directly. A receiver is a passive entity and cannot directly influence the system.

b. sender (s) - ACE gathers information about the external parts of the system via senders. These devices supply inputs to the ACE and can directly influence the system.

c. sender/receiver (s/r) - A sender/receiver combines the characteristics of senders and receivers as a single entity.

d. storage units (su) - storage units are a separate class of passive devices. They are considered to be passive because they retain the information that they receive and can send previously received information only when they are properly queried. This makes them different from senders which are capable of initiating input based on the presence or absence of some external event.

DATA PATH indicates the hardware path used for transfer of information. The multiprocessor communications adapter (MCA) permits full duplex asynchronous communications to occur between two or more processors.

DATA PATH TYPE (DPT) categorizes the data path into one of three categories.

a. parallel (p) - The transfer of several binary information digits (bits) occurs at any one time.

b. serial (s) - Information is transferred as one sequential bit stream.

c. analog (a) - Directly measurable signals are transferred, e.g., voice is transferred as frequency and magnitude with respect to time.

SECTION IV

CONSTRAINTS

The purpose and intent of this section is to describe the constraints which will be limiting the capabilities of the ACE system. These constraints are divided between the areas of training and system. Training constraints are categorized under the subjects of general orientation, clarity, speech recognition, hardware, software, courseware, and miscellaneous. System constraints are the limitations imposed on the system due to the rules of implementation of the software. The rules concern documentation requirements, general programming standards, and the usage of ACS and DGL.

TRAINING

The ACE system was conceived as a research tool for investigating application of speech technology to training. With that as its major priority, the ACE system is attempting to use a real world training task as its context. Therefore, because ACE is a research tool rather than an operational training tool, the instructional system's development (ISD) process has been limited in scope. To develop a true training system, it is important to allocate adequate time up front for identifying the training problem. To do that, one first identifies the gap between the training goal and the present training situation. The next step in the process is to identify the training methodology. The third step is to plan a system that makes use of the methodology. This is followed by the design of the specific courseware, hardware, and software in order to make the system work. The final step is to test, tune, validate, and revise those components of the system until they work up to the previously established criterion standards.

For ACE, the specification of hardware components, training analysis, and software decisions are being simultaneously accomplished. Because the hardware decisions and software specifications must be based on limited information, revisions and/or accommodations have been made which have constrained the training approach. In addition, even when the system is developed, there is very little time allocated for tuning and testing and virtually none allotted for formal courseware validation and revision.

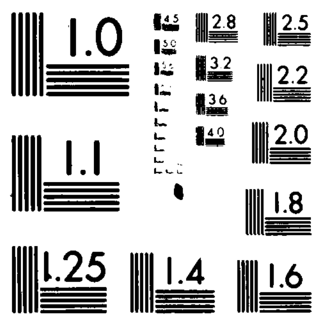
CLARITY. Another important constraint is the relative "fuzziness" of the AIC's job due to (1) sketchy job analyses and (2) the effects of changes in the job tasks presently being promulgated. There are at least six different areas which contribute to this lack of clarity of the AIC's role.

The first area of concern is that the AIC job has never undergone an extensive job task analysis. The job has been assessed in terms of the AIC's interface with the NTDS related equipment in the Refined AIC Job Task Inventory promulgated by the Commander, Training Force Pacific Fleet (COMTRAPAC),8

8. Refined AIC Job Task Inventory, Enclosure (1) to Commander Training Command, U.S. Pacific Fleet, letter serial 2834, December 19, 1975 (Appendix A to Behavioral Objectives report).

AD-A113 209 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
FUNCTIONAL DESIGN FOR AIR INTERCEPT CONTROLLER PROTOTYPE TRAINI--ETC(U)
DEC 81 E REGELSON, G SLEMON, R VERSTEEG N61339-78-C-0182
UNCLASSIFIED 8641-A009 NAVTRAEQUIPC-78-C-0182-8 NL

2 of 3



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

but no in depth analysis has been made of the various cognitive, affective, and psychomotor behaviors and skills involved. Early research for documentation on this project has filled part of this gap (see Behavioral Objectives, Objectives Hierarchy, and Ordinal Syllabus reports), but is still less than adequate for precisely detailing the skills involved.

A second area which affects clarity is the fact that the AIC's job changes with respect to the airborne weapons system with which he is working. The F-4 and F-14 aircraft have radar intercept officers (RIO) who perform many of the detecting and maneuvering functions formerly done by the AIC for one seat aircraft. The advent of the F-18, again with no "backseater," means that the AIC will again be responsible for detection, maneuvering, and helping to cover the fighter's "six o'clock." Since the job changes this way, it is very difficult to teach the AIC his exact responsibilities. Instead, it is easier to teach AIC candidates to understand the decisions involved in the job and let them adapt to the needs of the differing tactical situations.

A third problem for developing a training program is the lack of precisely identified AIC procedures. For any given task there is usually a difference in the way it is taught on the West Coast and on the East Coast. There may even be more than one way taught within a single school. The ACE training system requires rigorously defined procedures in order to be able to do accurate performance measurement. As a result, it has been necessary to define job task procedures just for ACE.

Another area which has never been defined in detail and which directly affects the development of ACE is the AIC vocabulary. Although established communications procedures do define many vocabulary items for the AIC, "real world" communications can be fairly loose. With the limited vocabulary available for use on ACE, it has been necessary to define a specific limited vocabulary. This may meet some resistance from instructors, but should help develop improved communications techniques in the learners.

A fifth area that impacts the development of the training system is an ongoing change in the NTDS program. The NTDS program is presently being changed from version 4.0 to version 4.0.1. The changes involved in the new program require parallel changes in the ACE procedures and, until the changes are promulgated, limit our abilities to test incoming learners on their pre-requisite skills.

The final impact area is the major changes which are being made in the FLECOMBATRACENPAC AIC School curriculum and approach. These changes amount to a positive revision of the curriculum, designed to enhance training effectiveness. However, these changes can make aspects of the current ACE curriculum, designed around the previous curriculum, contradictory or incorrect with respect to the FLECOMBATRACENPAC training package. It may be necessary to revise our curriculum to realign the two approaches.

SPEECH RECOGNITION. The NEC speech recognition system selected for this project will impose several constraints on the whole training system. Because of the importance of speech recognition, significant enhancements to the NEC system would cause these constraints to be modified.

There are a number of constraints with the present configuration of the NEC speech recognition system. The one second per phrase, five phrase concatenation limitations will require special, unrealistic vocabulary phrasing in order for the learners to get recognition. For example, present system recognition results indicate a requirement for special phrasing such as pauses between digit strings and words.

The vocabulary which has been listed in Appendix B is a compromise between training and speech recognition needs as they are presently understood. Speech recognition research is continuing. If this research shows that a more limited vocabulary will be significantly helpful in achieving adequate recognition, there are several phrases which have been identified as expendable. These phrases will be eliminated as necessary without severe impact on training effectiveness.

The 120 phrase maximum memory for the NEC places severe limitations on the vocabulary. The fact that recognition may require the collection of up to three voice reference patterns (VRPs) per phrase additionally limits the usable vocabulary. Furthermore, swapping reference memory into and out of the computer can require over two minutes. It will require special planning in the instructional system to make these delays acceptable to the students.

Since recognition degradation may occur as a student progresses within a level, the student may be periodically subjected to the tedium of VRP recollection. Multiple collections of VRPs may introduce learner morale problems with which to contend.

All vocabulary for a challengeable sequence (one for which the learner can try to show mastery without having to go through the intervening instruction) will have to be trained prior to implementing the challenge. Training all the phrases at the outset or prior to a challenge requires voice training out of context. Out of context training can present additional problems to the quality of speech recognition. These constraints all put additional pressure on the development of an adequate introduction to speech recognition and VRP collection so as to enhance speech recognition success.

The NEC system is very speaker dependent, to the point where a change in speaker voice stress or voice quality due to situational pressure, a cold, or fatigue can significantly degrade recognition. This characteristic requires the development of special procedures for collecting revised VRPs from time to time or a procedure for collecting VRPs which span the entire range of speaker voice quality. Either procedure results in special procedures which take away from the training emphasis and can erode learner confidence in the system.

HARDWARE. The hardware constraints involve the videodisc, the TEC, and the Votrax. The inclusion of the MCA videodisc into the ACE configuration has eliminated some constraints, but others still remain. Two sides of one disc can provide ACE with up to 70-80,000 single frames of information but limits it to a maximum of only 54 minutes of audio and/or motion video for the three week course. Because the MCA player can only read one side of the disc at a time, the information is further limited to 27 minutes of audio/visual data unless the disc is turned over by hand. The final videodisc limitation is that presently videodisc is a "read only" medium. This means that once the disc is mastered, the information on it cannot be changed. Making any changes requires mastering an entire new disc.

Constraints associated with the TEC derive primarily from the fact that there is no one standard console in use in the fleet. The decision to emulate the UYA-4/V-10 console will mean that different learners will have differing levels of familiarity with the console. Special time must be set aside at the beginning of the instruction to get each of the learners acquainted with the differences between the console simulation and their own familiar console. Cost considerations have also led to the decision to make parts of the simulated console non-operative. These parts of the console would not normally be used in the instruction, but the learner must be told that they do not work.

Constraints associated with the Votrax are a function of the state-of-the-art of computer voice synthesis. Difficulties with pronunciation of certain sounds and difficulties in providing correct stress and inflection to each phrase to be spoken lead to constraints in intelligibility and learner acceptance of computer synthesized speech.

SOFTWARE. There are only two general types of software constraints: simulation and automated instructor. The constraints on the simulation are that identification friend or foe (IFF) and data link will not be simulated and this, in turn, limits ACE's ability to teach the whole AIC job.

Within the automated instructor there are two types of constraints. The first is that there is little learner control over the instructional sequence. Although the system does allow the learner to challenge entire segments of instruction if he thinks he already possesses the skills being taught, the learner alternatives within Interactive Teaching (IAT), Commented Practice (CP), and Free Practice (FP) sequences are very limited. The major different paths provided for different rate or type of learners are through remediation. A good learner can speed his passage through CP and FP runs by showing early mastery using the minimum number of trials. Within the interactive teaching sequences, an alternate [illegible]ructional path can be provided for the learner who already knows almost a[illegible] content. The "step" command provides entry and exit points within instr[illegible]l sequences. Courseware developers' dexterous manipulation of the "step" [illegible]mmands can allow the learner to skip or repeat certain aspects of the total instructional package through learner control. These pathways are still extremely limited.

The second constraint within the automated instructor is, as noted earlier, that all the vocabulary for the instructional material covered spanned by a learner challenge must be trained before the student is given the choice to "challenge" the sequence. This means that most of the vocabulary must be trained outside the instructional context to which it applies.

COURSEWARE. Courseware limitations derive from the state-of-the-art of computer based instruction (CBI) and from the finite resources available. Although much research is currently being undertaken in the area of learning theory and CBI, there are still large gaps in the knowledge spectrum, especially in the areas of remediative instructional sets (diagnostic, prescriptive, and remediative) and of adaptive instruction. Since it is not within the scope of this project to perform in depth research and development in those areas, the courseware control design is based upon presently existing, somewhat limited models, and those areas are somewhat restricted in scope in ACE.

Other limitations which are affecting the development of the training program are both external and internal to the project. Externally, there is very limited information concerning the learner's entry knowledge level. A common error analysis and the validation process for the pretest should provide some data inputs in this area, but there will still be no comprehensive data base for decision making concerning courseware design. Internally, ACE is a very complex and extensive training system in regards to the AIC's job. Supplying the "nitty-gritty" day-to-day decisions for specifying and developing the eight levels of syllabus identified courseware which would utilize the full capacity of the ACE training system would require a larger courseware staff than is available. Instead, limitations are being identified which will both best use the ACE capabilities and still be within the resources available to the courseware staff. This process includes designing instruction which will require as little revision as possible and will limit the amount and scope of remediative instruction being developed.

MISCELLANEOUS. Miscellaneous constraints have to do with the constraints on the system which result from its placement and use at FCTCP. Historically, instructors have grave suspicions about new teaching systems, especially ones that they feel might eventually replace them. At this point we have no control over the attitudes or the abilities of the instructors that are to be chosen to use ACE at FCTCP. We must rely on a one week instructor training course to overcome any biases and to train them into a whole new role of instructional manager and facilitator. Additionally, the system is being required to show criterion learner achievement of 90 percent of the learners passing 95 percent of the instructional sequences. Currently there is a drop out rate of 35 percent at the school and no criterion standard (outside of subjective instructor evaluation) being applied to those students who remain within the school. Moreover, there has been no strict control over the precise experience the learner must have before being admitted to the school. Unless there is some selection process utilized in the identification of candidates for enrollment into the ACE curriculum (such as passing the pretest), the likelihood of ACE performing up to the identified criterion standard is severely diminished.

SYSTEM

This major segment establishes both standards and policy for the various software efforts of this project. The primary objective is that the programs be developed in an orderly and efficient manner. A secondary benefit anticipated is flexibilty and ease of maintenance in the future.

DOCUMENTATION. Much has been written and many efforts have historically been made to provide adequate documentation for software projects. This particular project will emphasize documentation in areas indicated below. Two of the items, the Program Design Language and the System Interface Notebook, are programming tools only and are not contract deliverable items.

Functional Design Report. This document is an important part of the ACE system. It is a detailed formal specification of the design of the entire ACE system. It is organized in terms of the functions that the system must perform in order to fulfill the requirements imposed by the training objectives, and specifies the inputs, outputs, and method of implementation of these functions. It is intended to represent the system design as of the date of publication of the document, and to fix it in that state; any changes to the overall design after publication will be the results of necessity and not mere changes in policy or attitude.

Program Design Language. The Program Design Language (PDL) is a tool to aid in designing and documenting a program or a system of programs. A design in PDL is written in structured English which is then input to the PDL processor. The output is a working design document consisting of a table of contents, a listing of the segments automatically formatted, a display of the procedure calling tree, and a cross-reference of the procedure calls.

PDL is a product of Caine, Farber, and Gordon, Inc., which has been obtained by Logicon for use on several projects. The detailed design phase of this project will make heavy use of this product and the detailed design documents will be outputs of the PDL processor.

System Interface Notebook. The sole function of this notebook shall be to clearly and accurately record the interfaces between all software entities in ACE. This includes interfaces between computers and between modules within a computer. The System Interface Notebook is organized as follows:

The first section shall consist of the definition of all interfaces between the three computers detailing the MCA messages sent and received. Messages will be numerically ordered. This section should ensure that each computer's inputs and outputs are exactly defined.

The second section shall contain instructor computer IPC definitions. This section will define all the IPCs used for communications between the processes in the instructor computer. It will list all IPCs (ordered by function code), their format, the sender, destination, and a description of the function of the message.

The third and fourth sections shall contain the same information as the second section, but for the simulation and speech configuration.

The fifth section shall contain shared page layouts arranged by the process which uses them.

The sixth section shall contain file definitions. For each file in ACE it will contain information about which processes create and use the file, where the file resides, the type, structure, and organization of the file, and the explicit contents of the file.

It shall be the responsibility of the module designer to define the messages required to support the function of the module and to make sure that the System Interface Notebook is accurate.

Program Source Listing. The lowest level of software documentation is the program listing itself. It will serve as the final reference for any changes or modifications to the software. Listings of all delivered Logicon written programs will be provided with the following characteristics:

a. The programs will be separated by language, that is, all DGL together and all assembly language together.

b. They will be arranged in alphabetical order by the name of the module (program, subroutine, procedure, etc.).

c. The program listings will, of course, reflect the programming practices and conventions.

d. Each program module listing will contain a header with the following information:

(1) The module name if it does not already appear as part of the program.

(2) The name of the principal author, the date of the most recent revision, and the name of the project.

(3) The module calling sequence if not clear from the code.

(4) A broad description of the module's function. This includes inputs and outputs not obvious in the code, files referenced, external references, or any other information which is likely to be needed by another programmer to maintain or modify the program. If this module is the main module of a task or process, this section will contain a thorough explanation of how the entire task or process functions. This includes the circumstances under which it runs, how it communicates with other tasks or process, significant resources that it uses, etc. It is the program author's responsibility to ensure that this section contains all pertinent information. This section will correspond with the "text" section of the PDL for this module.

GENERAL PROGRAMMING STANDARDS. The objective of the standards and conventions provided in this section is to ensure the writing of "good" programs and thereby ensure the success of the project. The following standards and conventions are presented as firm guidelines, not unbreakable rules.

Modularity. Programs will be constructed of independent modules following the single function module concept. To the greatest extent possible, these modules will be designed so that they can be replaced or modified without affecting other modules.

Source Files. Each program module will exist as a separate file. This also applies to "include" files. Comments will be maintained within "include" files to enumerate all using modules.

Comments. All comments will convey the larger functional role of a statement or instruction, or group of statements or instructions. A comment will not be the mere translation of the instruction into English. Any section of particularly obscure code will be preceded by a paragraph of comments explaining the intention of that code. In any case there will be sufficient comments in a module to enable a following programmer to finish it, debug it, or modify it.

Self-modifying Code. Self-modifying code will not be permitted.

Shared Temporary Storage. Modules will not share temporary storage among themselves. Sharing temporary storage requires the assurance that modules will not conflict with each other, which needlessly complicates system design.

Local Data Elements. Local data elements will be defined in a separate section of code preceding any executable code.

Entry Points. Each module will have a single entry point. This entry point will be the first executable instruction or statement.

Program Flow. Modules will be coded such that they flow down the page, even at the cost of extra branches or jumps. This organization enhances the readability of the listings. This convention is intended primarily for non-structured assembly language programs. The structural organization of DGL programs is treated elsewhere.

Exit Points. All exits from a sub-module will occur through a single normal or one alternate error exit point. These exit points will be the last executable statements or instructions in a module.

Module Length. A module will be long enough to perform a single function. This normally should not require more than 100 executable statements.

Variable Names. Variable names will be chosen which reflect or indicate the contents of the variable.

Reentrant Code. All routines written in assembly language will be reentrant and will conform to Data General practice in this area. All high level language compilers generate reentrant code for routines which do not use "common," "own" or "equivalence" data storage.

Debugging Measures. To the greatest extent possible, programs will be written to prevent or automatically catch bugs. That is, they will include features to:

a. Check the validity of arguments passed to a module.

b. Make range and reasonableness checks on all data input from outside the program.

c. Check the range of control variables used in computed GOTO statements.

d. Make array subscript range checks.

In some cases these checks will require extra code, and in some cases they can be accomplished with the use of compile option switches. If the checks require additional code, this code will be marked for conditional compilation so that it can be easily removed if necessary after testing has determined that the program functions correctly.

AOS Usage Guidelines. This information is intended to provide guidance to the designer and programmer so that the system will use AOS efficiently and avoid its pitfalls.

Grouping of Functions. All interrelated functions will be grouped together and handled by a single process. For example, all functions related to updating the aircraft displays will be in the same process.

Assignment of Functions to Tasks. Each clearly defined function in a process will be handled by a separate task assigned to it. For example, update of aircraft tracks and the maintenance of the aircraft's geographic plot will be assigned to different tasks within the same process.

Assignment of Process Types. Processes in which the main task runs frequently (twice a second or more often) or for another reason must remain in main memory will be declared "resident." Other processes concerned with real-time online operation of the system will be "preemptible." All ancillary and off-line support processes will be "swappable."

Use of Overlays. Use of overlays solely to reduce the size of a process, that is, placing tasks or routines which run frequently in overlays, will be reserved and resorted to only if memory limitations become a problem. The extra complexity of the design, the execution time used by the overlay loading facility and the increased difficulty in testing may not outweigh the savings in memory. Exceptions to this are large one-time initialization programs.

Use of IPC. AOS provides a facility for communication between processes in a machine that consists of the ability to send variable length data messages between ports associated with processes. This facility will be used extensively in ACE to control and coordinate the various activities. Detailed data concerning use of IPCs for this purpose is contained in the introductory section of this document. Use of IPCs requires significant system resources, especially central processor time. This dictates that they not be used casually and not outside the framework established by this design.

Resource Deadlock Caution. AOS allocates resources dynamically in accordance with some very sophisticated algorithms. This makes the system very fast and flexible. However, in any operating system of this complexity there is a possiblity of resource deadlocks where one task or program cannot proceed because it is waiting for a resource held by another, and it in turn (perhaps indirectly) holds a resource needed by another. This is particularly true of memory when a task or process takes some action (e.g., some system calls) which cause the process or the operating system itself to grow. In all cases such as this the system may not be able to swap out enough code to accommodate the expanded program. For this reason applications processes will be designed so that they do not dynamically change size, and those activities which cause the operating system to grow will be carefully monitored.

File Handling. The file structure will be designed so that only one task in one process will be required to write into that file. All files will be open for "reading" to all tasks that have a need for the data. This restriction is to prevent deadlocks and the occurrence of inconsistent data.

Use of Common Area. There will be one page of shared memory reserved to hold data of interest to more than one process. This will contain data concerning the present state of the simulator and various system wide information from ACE. This data will be accessed via the literal pointer facility of DGL. Files containing the pointer definitions will be centrally maintained and can be "included" in the program wishing to access data in that shared page. The system will be designed so that only one task will write into a particular location in the shared page. A mechanism such as this carries with it some potential dangers from erroneous or unintended updates of data. For this reason access to it will be restricted to the literal pointer mechanism mentioned above.

Creating Processes. AOS permits processes to be created and killed dynamically. However, each creation requires substantial system resources and takes several seconds to complete. Therefore, for processes that run repeatedly, it is better to use the block/unblock mechanism to control their execution.

Number of Tasks Per Process. Although AOS will permit up to 32 tasks in a process, past experience indicates that four or five is a more practical limit. Use of more than that should be approached with extreme caution.

Task Scheduling. AOS and RDOS provide the programmer with some powerful and flexible (and dangerous) ways of altering the algorithm used by the system to schedule task execution. A user program may dynamically alter a task's priority or suspension conditions, and may communicate between tasks via an intertask system of what are commonly referred to as "mailboxes." While these utilities may be needed in some cases to resolve extreme or delicate timing problems, past experience has shown that dynamic priority changes are difficult to use effectively and often result in a solution that is worse than the problem. Because of this, in ACE dynamic task priority changes will be avoided and intertask communication via mailboxes carefully planned.

Common Sense. AOS is a very powerful and versatile operating system. But it also has some limitations, particularly in the area of real-time process control (such as ACE). A little thought and a lot of common sense ought to be exercised when using the system. It is not possible to list all pitfalls and danger areas, but practices such as these are examples:

a. Using busy loops in high priority processes.

b. Doing reads and writes from processes running on a periodic time schedule.

DGL Usage Guidelines. The following guidelines are intended to enable the efficient use of those features unique to DGL, the principal language used on the ACE project.

Structured Flow. DGL is a block structured, procedure oriented language that is very suitable for structured programming. Over the last several years this method of writing programs has received much favorable publicity because of the ease of maintenance and testing and higher reliability of such programs. However, the application of this technique has been slow in coming to real-time situations. This is because the processing of asynchronous events does not lend itself to the technique and because structured programs are not generally as efficient in the use of memory and execution time as are non-structured. However, DGL uses an optimizing compiler which should minimize the inefficiencies accompanying structured programs. The techniques of structured programming will be followed where possible on the ACE project at the task level and where possible at the process level.

Argument Passing. DGL has a capability not found in many other languages in that it is possible to access any variable in a superior block or procedure in the program. Although this capability is often useful, it is also the source of many bugs and other difficulties (such as readability of the code) and for this reason is discouraged. That is, all variables needed by a subordinate procedure will be passed as arguments. Also, when passing arguments to procedures, they will be "by value" except if a value is to be actually returned to the calling procedure.

Arrangement of Procedures. In DGL, procedures can be bodies of code within the program or can be external to it if so declared. Generally all procedures should be handled as external. This makes the listings shorter and more readable and makes the source files more manageable. If a procedure is not worthy of being handled as an external, it will be defined within the body of code after the main procedure.

Use of "Own" Variables. "Own" variables constitute a form of shared temporary storage mentioned in the section on general standards and conventions. Their use is discouraged.

Error handling. DGL has an extensive facility for detecting and handling errors. Included in this is the user's ability to intercept an error and handle it in any way he sees fit. The ACE system will use this facility to screen all errors, determine their severity, and act on the error according to the following:

a. If the error is minor and has no effect on the real-time operation, ignore it.

b. If the error can be corrected by the program, do so.

c. If the error cannot be corrected and does effect continued operation, invoke the error shutdown procedure.

APPENDIX A

SCENARIO GENERATION SYNTAX

INTRODUCTION

This appendix describes the commands that are available to create scenarios for ACE training.

GENERAL SYNTAX RULES. The following basic rules apply to the syntax:

a. Multiple arguments are separated by spaces
b. Argument data are integers
c. All commands are terminated with the new line key
d. Comments may be included by placing a "C" and a space in the first two columns
e. All bearings are degrees magnetic from ownship
f. Command names must be all capitalized.

SCENARIO GENERATION COMMANDS SUMMARY. ACE scenario generation commands consist of initialization and control commands summarized as follows:

a. Initialization Commands

INITPOS (track #, bearing, range, heading, speed, altitude)
RANDPOS (track #, bearing, range, heading)
INITCOND (track #, motion model, turn rate, directed heading, video size, NTDS #, track type, PIF
PACKAGE (#)
BACKGROUND (#)
ROTSCRN (degrees)
ENGAGED (track #, track #)
OFFSET (bearing, range)
MODPROF (pilot, bogey, tracker, SWC)
FUEL (track #, # of pounds)
CALLS (visuals, contact, judy, tally ho, lost contact, famished)
CAPSTN (bearing, range)
TACSTN (bearing, range)
REFPT (track #, bearing, range)
OUTCOME (outcome #, # bogeys)
POINT (track #, bearing, range)

b. Scenario Control Commands

TRKON (track #, time or event)
TRKOFF (track #, time or event)
CHGHDG (track #, time or event, motion model, turn rate, directed heading)
JINK (track #, time or event, degrees, time duration)
FADE (track #, time or event, # of sweeps)
SPLIT (track #, time or event, split heading)
NTDSFAIL (time or event)
EMERGENCY (time or event)

STRANGER (track 3, time or event)
FREEZE (time or event)
END (event, time)
ABORT (time, event)
RENDEZVOUS (time or event)

INITIALIZATION COMMANDS

Descriptions follow of the commands that are used to initialize ACE scenarios. Up to 12 tracks may be initially positioned in relation to ownship by bearings and ranges (range in data miles). For simplicity and consistency, track numbers are allocated as:

0 - ownship
1 - CAP
2 - bogey and pseudobogey
3 - stranger
4-12 - background tracks
13 - not used in initialization (reserved for bogey split, see SPLIT below).
14 - CAP station
15 - TACAN
16-18 - reference points
19-20 - points in space

INITIAL POSITION. This command describes the initial position of a track when it is first turned on.

INITPOS (track #, bearing, range, heading, speed, altitude)

Valid argument values are:

a. track # - numbered 1-12
b. bearing - degrees, 1 to 360
c. range - miles, 0 to 64
d. heading - degrees, 1 to 360
e. speed - mach *10, 1-25
f. altitude - thousand feet, 0 to 50

RANDOM POSITION. A capability exists for the scenario writer to indicate that a random increment is to be applied to the initial position arguments. The random increment limits are provided by this command and the real time program determines the actual increment to be applied. If the Random Position command is not used, the initial position data are used.

RANDPOS (track #, bearing, range, heading)

Valid argument limits:

a. track # - numbered 1-12 (For bearing, range or heading, a number entry offers a range of application as a plus-or-minus increment from the initial INITPOS value entered for bearing, range or heading.)

b. bearing - degrees, 0-10
c. range - miles, 0-10
d. heading - degrees, 0-10

INITIAL CONDITION. This command describes the prevailing conditions of a track when it is turned on.

INITCOND (track #, motion model, turn rate, directed heading, video size, NTDS #, track type, [PIF or PU])

Valid argument values are:

a. track # - numbered 1-12
b. motion model

1 - simple (straight)
±2 - turning (+2 for starboard, -2 for port)
±3 - orbit (+3 for starboard, -3 for port)
4 - stationary (surface vessels)

c. turn rate
1 - standard (3.0 degrees per second)
2 - hard (4.5 degrees per second)

d. directed heading - degrees, 1 to 360

e. video size
1 - small
2 - medium
3 - large

f. NTDS # - 0 to 7777 octal (an NTDS track number)

g. track type
0 - no symbol displayed
1 - ownship
2 - cap
3 - bogey/unknown
4 - friendly/stranger
5 - hostile/bandit
11 - engaged cap
12 - CTF cap
13 - engaged friendly
14 - CTF friendly
15 - surface friendly
16 - engaged bogey/unknown
17 - CTF bogey/unknown
18 - surface unknown
19 - engaged hostile
20 - CTF hostile
21 - surface hostile

22 - ball tab
23 - video
24 - CAP station
25 - hook
26 - pointer
27 - arrow
28 - large zero
29 - large R
30 - large B
31 - large F

h. PIF # (only if track type is CAP)
PIF (CAP) XXXXX (octal), Personal Identification Function, 0-29000

PACKAGE. This command allows the user to select a predefined syntax description with computer randomization of the CAP, bogey, and ownship for inclusion in the exercise.

PACKAGE (#)

The package number will be used to identify a file containing the syntax descriptions; e.g., 4 yields PACK4.SN. Defined package numbers consist of:

1 - standard (bogey, CAP)
2 - stranger (bogey, CAP, stranger)
3 - joinup
4 - setups

If the Package command is not used, the CAP, bogey, and stranger positions must be specified by the INITPOS & INITCOND commands. At least one track must be defined to operate a scenario. There will be no more than ten packages.

BACKGROUND. This command allows the user to select a predefined syntax description with computer randomization of background tracks for inclusion in the exercise.

BACKGROUND (#)

The background number will be used to identify a file containing the syntax descriptions; e.g., 3 yields BKGN3.SN. If this command is not used, desired background tracks must be scripted and included using the INITPOS and INITCOND commands. Otherwise, there will be no background tracks. There may be no more than twelve background tracks.

RANDOM SCREEN ROTATION. This command may be used to rotate all tracks of a scenario to create a different look.

ROTSCRN (degrees)

Display may be rotated 1 to 360 degrees. Zero (0) entry indicates that the rotation amount is to be randomly selected by the computer. If this command is not used there will be no rotation.

OFFSET. This command may be used to initially create a radar display representation with the center of the scope designated by the input bearing and range coordinates. Ownship, from which the sweepline originates, will reflect this offset by appearing in a non-centered location. If omitted, ownship is in the center of the screen.

OFFSET (bearing, range)

Valid argument values are:

bearing - degrees, 1-360
range - miles, 0-64

ENGAGEMENT REQUEST. If desired, the initial picture can be defined with CAP or pseudobogey engaged to a CAP station, another initiated track, or a bogey.

ENGAGED (track #, track #)

Valid argument values are:

track # - 0, 2-12 or 14

Note: 14 indicates engagement to the CAP station

MODEL PROFICIENCY. This command allows nonstandard proficiencies to be established for the pilot, bogey, tracker, and SWC models. If omitted then the model proficiencies will be defined by the following default conditions: 2 - smart pilot, 0 - script bogey, 0 - perfect tracker, 0 - passive SWC

MODPROF (pilot, bogey, tracker, SWC)

Valid values for each argument have the following definitions (refer to applicable model functions for definition descriptions):

value -	pilot	bogey	tracker	SWC
0 -	script	script	perfect	passive
1 -	directed	-	-	fair pesty
2 -	smart	-	-	pesty
3 -	dumb	-	-	-

FUEL. The CAP or pseudobogey can be initialized with a set amount of fuel.

FUEL (track #, # pounds)

Valid argument values are:

track # - 1-2
pounds - 1-185 (in hundreds of pounds)

PILOT CALLS. This command allows the user to request a percentage of time that the pilot is to make the correct calls when the conditions exist. Conditions only apply to the platform; i.e., no calls are made against the missile (split track). All percentages default to 100.

CALLS (Visuals, Contact, JUDY, TALLY HO, LOST CONTACT FAMISHED)

Valid percentages that may be selected for each argument are: 0, -100. For contact calls a value of -1 is also valid and means that all calls will be incorrect.

STATIONS. When desired, one each CAP station and TACAN station maybe designated.

CAPSTN (bearing, range, altitude)

TACSTN (bearing, range)

Valid argument values are:

bearing - degrees, 1 to 360
range - miles, 0 to 64
altitude - angels, 0-24

REFERENCE POINTS. When desired up to three reference points maybe designated.

REFPT (track #, bearing, range)

Valid argument values are:

track # - ±19, ±20 (a negative track # will define a reference point, but no symbol will appear on the PPI)
bearing - degrees, 1 to 360
range - miles, 0 to 64

INTERCEPT OUTCOME. Desired outcome of an intercept can be selected, assuming the proper conditions are satisfied. If no outcome is defined, default will be zero. Outcome is processed by the pilot function after a tally ho condition and after receipt of a breakaway command from AIC.

OUTCOME (#,y)

The argument "y" is used to specify the number of bogeys to be reported (1 to 3). Default will be one. Valid argument values for # are:

0 - computer determines whether splash or heads up
1 - splash
2 - heads up

SCENARIO CONTROL COMMANDS

These commands may be used to alter the initial conditions at designated times or upon the occurrence of a predefined event.

Time is expressed in seconds from exercise begin and events are expressed as negative integers. The definition of events will be contained in the Instructor Handbook. All track numbers are in the range of 0 to 12 (13 for split reference).

TRACK ON/OFF. These commands allow the scriptor to control the display of individual track video. Video is initially turned off.

TRKON (track #, time or event)

TRKOFF (track #, time or event)

CHANGE TRACK HEADING. Individual track headings may be changed by using this command.

CHGHDG (track #, time or event, motion model, turn rate, directed heading)

Valid argument entries for motion model, turn rate, and directed heading are defined under the INITCOND command.

BOGEY JINK. This command allows the scriptor some relative turning control. Normally this command is used for "jinking" the bogey.

JINK (track #, time or event, degrees, seconds)

Valid arguments are:

track # - 1 to 12

degrees - Relative heading amount (1 to 360), negative degrees for port direction, positive degrees for starboard direction. Turn rate will be at current rate. If track two is selected, the degree argument will be interpreted as a relative heading change in either direction with respect to the current heading. Track two example: If track's present heading is 180, and 30 is the degree argument, the computer would select between headings 150 and 210.

seconds - The total time for the jink and the duration of time that the Bogey Model is to be turned off. Note: This argument is ignored for all tracks other than track # 2 or 13 and need not correlate with the time to reach the new heading.

FADE A TRACK. This command allows an individual piece of video to be explicitly faded.

FADE (track #, time or event, # of sweeps)

Valid argument for # of sweeps is 1 to 84

SPLIT A TRACK. This command allows the scriptor to split one track. The new track is identified as track number 13 and simulates a missile.

SPLIT (track #, time or event, split heading)

Split heading is in degrees, 1 to 360

NTDS FAILURE. NTDS functions may be inhibited for the remainder of the exercise with this command. Scenario will continue with all other elements.

NTDSFAIL (time or event)

Time must be greater than zero.

EMERGENCY. This command may be used to indicate that the CAP has declared an emergency. A beeper on the guard channel of the TEC is activated.

EMERGENCY (time or event)

STRANGER. This command allows the scriptor some control over associating tracks with the stranger model. Control of the indicated track (limited to track 3) will be turned over to the stranger model at the designated time or event. Track's video will be ascertained to be on. If the track has not been initialized, an error will occur.

STRANGER (time or event, track 3)

FREEZE. This command may be used to cause time to cease being updated and consequently tracks from moving. Radar sweep continues.

FREEZE (time or event)

END. This command allows scenarios to be gracefully ended.

END (time or event)

ABORT. This command allows the scriptor to identify a time at which a particular event must have been completed, otherwise cause the exercise to abort.

ABORT (event, time)

RENDEZVOUS. This command allows the scriptor to have the pilot function request a rendezvous. Normally, the rendezvous command will be based on an event, e.g., activation of a button.

RENDEZVOUS (time or event)

APPENDIX B

SYSTEM INTERFACE DEFINITIONS

INTRODUCTION

This appendix defines the system interface data. These data are described in an appendix because: (1) they are used by most of the programming personnel, and (2) they are the most susceptible to initial change when coding begins. The system interface data are described at the computer language level and include the following kinds of data:

a. Common data definitions

b. Data transferred among the ACE processors using the multiprocessor communications adapter (MCA) inter-computer high speed DMA interface and data transferred among the ACE AOS processes within a processor using the IPC capability

c. Micronova interface definitions

d. Scenario generation output definitions

e. Segment definition output definitions

Exhibits B1 through B16 at the end of this appendix contain computer printouts of files used to define common data and the MCA and IPC message formats.

COMMON DATA DEFINITIONS

These data are considered to be of interest to more than one process and/or processors.

SHARED PAGE. All common data or data that are required for use by multiple processes shall reside in the shared page area.

The shared page is divided into two major areas: the system one area and the system two area. They are so named because the instructor computer updates and maintains the values contained in the system one area; likewise, the simulation computer updates and maintains the system two area. Each of these computers transfer their portion of the shared page to the other computer each second so that all current data values can be referenced by processes on both computers. The speech computer receives only a select subset of the data on shared page and does not provide any data update to the rest of ACE.

Shared page variables access is achieved via the literal and pointer constructs of DGL. Each of the data areas consists of one or more buffer areas. A buffer area is defined to be a logical aggregate of related data that are physically contiguous. The system one area is composed of one buffer area which holds ACE system and trainer state variables. The system two area contains two buffer areas: an NTDS simulation area and a track data table area. At the start of each buffer area an entry is defined. Data within the buffer area is referenced as an offset from the entry.

The following convention is used to define a datum in the shared page:

LITERAL name ((pointer_expression) -> data_type);

a. Name - This is the literal by which the data is referenced and is defined by the originator.

b. Pointer_expression - The pointer expression contains the buffer area entry plus the relative word offset from the beginning of the current buffer area to the named datum.

c. Data_type - This represents the amount of storage associated with the data. These types are:

(1) I2 or BI - a single precision integer occupying two bytes of storage

(2) I4 - a double precision integer occupying four bytes of storage

(3) R4 or BR - a single precision real occupying four bytes of storage

(4) R8 - a double precision real occupying eight bytes of storage

(5) STRn or BSn - a string having n characters (one per byte) and occupying n bytes of storage

(6) B2 or BB - a single precision Boolean occupying two bytes of storage

(7) no data type - if no data type is specified the named datum is a pointer to the defined location in the shared page. This is done when the datum is the start of an array. An array is defined as a collection of related data items which is referenced via an index.

Exhibit B1 is a computer listing of the shared page definition literal file.

SYSTEM ONE AREA. This area is reserved for those data that are global to ACE in general. Of primary concern is information related to the state of ACE, the mode of operation, the identities of the current instructor and student, and the elapsed time on the master exercise clock.

NTDS SIMULATION DATA. This data area contains NTDS symbol data and TEC switch data as described in Exhibit B1. The track-related symbol data is referenced via data arrays indexed by the track number. Other data items stored include offset position, ball tab position, PPIRO, and intercept information. The

TEC switch data table contains the current switch settings for all operational TEC switches.

TRACK DATA TABLE. Track data for all videos are maintained in this data area. Each video parameter is stored as an array indexed by track number. In addition to this track data, the number of half degree counts and the student disk identifier are stored as described in Exhibit B1.

MULTIPROCESSOR COMMUNICATIONS ADAPTOR (MCA) AND INTERPROCESSOR COMMUNICATIONS (IPC) FORMATS.

The same format is used for data to be transferred across the MCA and via IPCs. The general format is defined as follows:

a. word zero contains intercomputer routing information

b. word one contains the message ID

c. word three contains the data length and sender's ID

d. word four and five contain a time tag

e. any additional words contain message data.

This general format and some commonly used data item offsets are defined in the computer file, IPC_FORMATS.LT. Exhibit B2 is a listing of IPC_FORMATS.LT.

All MCA and IPC formats are referenced in IPC.FORMATS.LT. However, the actual definition of many of the IPC messages are contained in smaller, message specific computer files. Exhibits B3 through B15 are printouts of these files.

STATUS DATA. Status data will describe the state that the processors are to assume. Exhibit B3 defines all possible states.

EVENT DATA. Event data indicates the occurrence of events in SYS2 and SYS3.

Micronova Reported Switch/Button Action Events. All Micronova reported switch/button action events are detected by the NTDS function. These events are of interest primarily to the simple testing function when the system is monitoring a check or test. Exhibit B4 contains the basic message format. Data contained within the message are defined in Table B1.

TABLE B1. NTDS SWITCH/BUTTON EVENTS

EVENT_ID	EVENT_DATA	DESCRIPTION
0	---	Not used
1	LABELS	"UP/LOW LABELS" VAB pressed. EVENT_DATA contains LABELS, the label state at the completion of a VAB entry. LABELS = 0 = upper labels in effect LABELS = 1 = lower labels in effect
2	LABELS	"ACCPT/DEL" VAB pressed.
3	LABELS	"BRK/CANTO" VAB pressed.
4	---	Not used.
5	---	Not used.
6	---	Not used.
7	LABELS	"TRACK HIST" VAB pressed.
8	LABELS	"POSIT DATA" VAB pressed.
9	1 - 6	"HDG/SPD/ALT | FUEL/MSLS" VAB pressed. EVENT_DATA = 1 = Heading = 2 = Speed = 3 = Altitude = 4 = Fuel = 5 = Rear Missiles = 6 = All-Aspect Missiles
10	N/A	"DN/LFT" VAB pressed.
11	N/A	"UP/RT" VAB pressed.
12	N/A	"POSCOR/STDY" VAB pressed.
13	LABELS	"GEOM" VAB pressed.
14	LABELS	"ORD SEND" VAB pressed.
15	---	Not used.
16	---	Not used.
17	LABELS	"NEW TRK" VAB pressed.
18	---	Not used.
19	0	"BALL TAB ENABLED" FAB pressed.
20	0	"BALL TAB CENTER" FAB pressed.
21	0 - 18	"HOOK" FAB pressed. EVENT_DATA contains number of hooked track.
22	0 - 26	"SEQ" FAB pressed. EVENT_DATA contains number of hooked track (0 - 18) or alert code +20 (21 - 26).
23	0	"ENTER OFFSET" FAB pressed.
24	0 - 18	"DROP TRK" FAB pressed.
25	0	"ENTER MODE AND RADAR" FAB pressed.
26	0	"FUNCTION CODE" FAB pressed. Followed by three separate messages of EVENT_ID 30, 31, 32.
27	0	"TRACK NUMBER" FAB pressed. Followed by three messages of EVENT_ID 30, 31, 32.

TABLE B1. NTDS SWITCH/BUTTON EVENTS - continued

EVENT_ID	EVENT_DATA	DESCRIPTION
28	0	"HEIGHT" FAB pressed. Followed by three messages of EVENT_ID 30, 31, 32.
29	0	"SIF CODE" FAB pressed. Followed by three messages of EVENT_ID 30, 31, 32.
30	NED 1	NED 1 value; BCD.
31	NED 2/3	Values of NEDs 2 and 3; packed BCD.
32	NED 4/5	Values of NEDs 4 and 5; packed BCD.
33	---	Not used.
34	---	Not used.
35	DATA	"RANGE" switch setting changed.
36	DATA	"CRT CENTER" switch setting changed.
37	DATA	"STANDARD LEADERS" switch changed.
38	DATA	"RADAR SELECT" switch changed.
39	DATA	"RADAR VIDEO SELECT" switch changed.
40	0	Not used.
41	0	Not used.
42	0	Not used.
43	0	Not used.
44	DATA	"TALK" switch setting changed.
45	DATA	"LEFT PHONE" switch setting changed.
46	DATA	"RIGHT PHONE" switch setting changed.
47	1 - 2	"FOOTSWITCH" state change. EVENT_DATA = 1 for "ON", 2 for "OFF".
48	1 - 2	"COMM1 (POINTER)" state change. 1 = ON, 2 = OFF.
49	1 - 2	"COMM2 (SWC)" state change. 1 = ON, 2 = OFF.
50	---	Not used.
⋮	⋮	⋮
112	---	Not used.
113	0	"ILLEGAL ACTION" alert issued.
114	---	Not used.
⋮	⋮	⋮
159	---	Not used.
160	0	"NTDS RESET".
161	---	Not used.
162	0	"NTDS RUNNING".
163	---	Not used.
164	0	"NTDS CONTINUING".
165	---	Not used.
⋮	⋮	⋮

Scenario Events. Scenario events are detected by various functions. The general message format is described in Exhibit B6. Events are listed in Exhibit B16 in a literal file format:

LITERAL event (event_number); % function_name

a. "event" is the name and decription of the event

b. "event_number" is the number which represents the event

c. "function_name" is the name of the function which detects the named event.

Abbreviations for the nonobvious functions are: ATC for adaptive training control, SC for scenario control, and ROFM for radar output formatter.

SUS Expression Events. SUS expression events are detected by the speech recognition function. These events are of interest to the appropriate pilot or SWC functions and to the performance measurement function. Exhibit B2 defines the general message format.

The SUS expressions are listed and identified by number in Table B2. Arguments to the expressions consist of applicable call sign (C/S) code identifiers, and values of variable data; e.g., heading. Applicable call signs are Silver Hawk and Crackerjack.

The following conventions will be used:

xxx - is heading or bearing spoken as single digits using 001 to 360 degrees.

yy - is the range in miles from 0 to 39.

fff - is the fuel from 0 to 125 pounds.

zz - is the altitude from 0 to 50 (thousand) feet.

m - is the mach speed as a single digit.

n - is a single digit number.

TABLE B2. SUS EXPRESSION EVENTS

EXPRESSION	EVENT #	EVENT LIST
ROGER	201	201
SAY AGAIN	202	202
CORRECTION	203	203
DISREGARD THIS TRANSMISSION	204	204
C/S; PORT XXX	205	205, C/S, XXX
C/S; STARBOARD XXX	206	206, C/S, XXX
C/S; VECTOR XXX	207	207, C/S, XXX
C/S; PORT XXX; FOR BOGEY	208	208, C/S, XXX
C/S, STARBOARD XXX, FOR BOGEY	209	209, C/S, XXX
C/S; VECTOR XXX; FOR BOGEY	210	210, C/S, XXX
STATION XXX; YY	211	211, XX, YY
BOGEY XXX; YY	212	212, XXX, YY
BOGEY TRACKING XXX; SPEED POINT M	213	213, XXX, M
C/S(;) MARK YOUR TACAN	214	214, C/S
C/S(;) WHAT STATE?	215	215, C/S
ROGER, STATE FFF	216	216, FFF
C/S, STATE FFF (TO SWC)	217	217, C/S, FFF
I HAVE CONTROL OF C/S (TO SWC)	218	218, C/S
C/S(;) ON STATION	219	219, C/S
C/S(;) BREAKING AWAY (TO SWC)	220	220, C/S
SPLASH; ONE/TWO BOGEY(S) (TO SWC)	221	221, 1/2
HEADS-UP; ONE/TWO BOGEY(S) (TO SWC)	222	222, 1/2
BOGEY SINGLE ALTITUDE ZZ THOUSAND	223	223, ZZ
BOGEYS MULTIPLE ALTITUDE ZZ THOUSAND	224	224, ZZ
BOGEY JINKING LEFT	225	225
BOGEY JINKING RIGHT	226	226
BOGEY SPLITTING	227	227
ROGER; BOGEY TRACKING XXX	228	228, XXX
NEGATIVE; BOGEY XXX; YY	229	229, XXX, YY
STRANGER XXX; YY	230	230, XXX, YY
STRANGER OPENING	231	231, XXX, ZZ
STRANGER OPENING	232	232
C/S(;) TIGHTEN TURN	233	233, C/S
C/S(;) EASE TURN	234	234, C/S
C/S(;) RADIO CHECK OVER	235	235, C/S
BOGEY IN THE DARK	236	236
FIGHTER IN THE DARK	237	237
C/S(;) MY OCTOPUS IS BENT	238	238, C/S
C/S(;) EMERGENCY (TO SWC)	239	239, C/S
C/S; PORT XXX; FOR RENDEZVOUS	240	240, C/S, XXX
C/S; STARBOARD XXX; FOR RENDEZVOUS	241	241, C/S, XXX
C/S; VECTOR XXX; FOR RENDEZVOUS	242	242, C/S, XXX
C/S DETACH PORT XXX; FOR SEPARATION	243	243, C/S, XXX
C/S DETACH STARBOARD XXX; FOR SEPARATION	244	244, C/S, XXX
C/S; CONTINUE XXX	245	245, C/S, XXX
C/S; BREAKAWAY XXX	246	246, C/S, XXX
C/S; ANGELS ZZ	247	247, C/S, ZZ
C/S; C/S XXX; YY	248	248, C/S, XXX, YY

TABLE B2. SUS EXPRESSION EVENTS - continued

EXPRESSION	EVENT #	EVENT LIST
C/S(;) STEADY	249	249, C/S
C/S(;) LOST COMMUNICATIONS INTENTIONS	250	250, C/S
ROGER LOST COMM	251	251
C/S; PORT XXX; AS BOGEY	252	252, C/S, XXX
C/S; STARBOARD XXX; AS BOGEY	253	253, C/S, XXX
C/S; VECTOR XXX; AS BOGEY	254	254, C/S, XXX
AIC1 (TO SWC)	255	255
BOGEY TRACKING XXX	256	256, XXX
SAY AGAIN (TO SWC)	257	257
EMERGENCY XXX YY	258	258, XXX, YY

Call signs are represented as follows:

SILVER HAWK	25
CRACKERJACK	26

Votrax Events. Votrax events are detected by the pilot function and are of interest to the speech generation and performance measurement functions. Exhibit B6 contains the general format for the message. Votrax events are listed in Table B3. The conventions used are identical to those used for the SUS Expression Events.

TABLE B3. VOTRAX EVENTS

EVENT	EVENT #	EVENT LIST
RUTH THIS IS SILVER HAWK ON MIRAMAR'S XXX YY, ANGELS ZZ, HEADING XXX, OVER	301	301, XXX, YY, ZZ, XXX
RUTH THIS IS SILVERHAWK ON MIRAMAR'S XXX, YY, CRACKERJACK, ANGELS, ZZ, HEADING XXX, OVER	302	302, XXX, YY, ZZ, XXX
LOOKING	303	303
VISUAL	304	304
CONTACT XXX YY	305	305, XXX, YY
JUDY	306	306
LOST CONTACT	307	307
TALLY HO, FOX1, BREAKAWAY	308	308
HEADS-UP N BOGEY(S)	309	309, 1/2
SPLASH N BOGEY(S)	310	310, 1/2
RENDEZVOUS POINT WHISKEY, ANGELS ZZ	311	311, ZZ
ROGER XXX	312	312, XXX
STATE FFF	313	313, FFF
ROGER EASE TURN	314	314
ROGER TIGHTEN TURN	315	315
XXX, YY (REPONSE TO TACAN REQUEST)	316	316, XXX, YY
REQUEST RENDEZVOUS WITH CRACKERJACK	317	317
FAMISHED	318	318
REQUEST BOGEY DOPE ON PLATFORM	319	319
ROGER	320	320
SAY AGAIN	321	321
JUDY, FOX1, BREAKAWAY HEADING	322	322
REQUEST BREAKAWAY HEADING	323	323
STRANGER OPENING	324	324
(REPEAT LAST PHRASE)	325	325

Digitized Speech Events. Digitized speech events are detected by the SWC or the pseudo bogey functions and are of interest to the performance measurement function. Exhibit B15 lists these messages. The specific events requested by the SWC and the pseudo bogey appear in Table B4.

TABLE B4. SWC AND PSEUDO BOGEY EVENTS

EVENT	EVENT #	EVENT LIST
SWC:		
WHAT STATE	401	401
UPDATE SYMBOLS	402	402
RESULTS OF INTERCEPT	403	403
BREAKAWAY REPORT	404	404
SILVERHAWK AIRBORNE FOR CONTROL	405	405
SWC AYE	406	406
THANK YOU	407	407
VERY WELL	408	408
SAY AGAIN	409	409
(REPEAT LAST PHRASE)	410	410
STATE BEARING AND RANGE FROM OWNSHIP TO EMERGENCY	411	411
PSEUDO BOGEY:		
ROGER	501	501
ROGER XXX	502	502, XXX
STATE FFF	503	503, FFF
ROGER EASE TURN	504	504
ROGER TIGHTEN TURN	505	505
XXX, YY (RESPONSE TO TACAN REQUEST)	506	506, XXX, YY
SAY AGAIN	507	507

SHARED PAGE DATA. Shared page data updates are passed from SYS1 to SYS2, from SYS2 to SYS3, and from SYS2 to SYS1 on a once per second basis for the purposes of performance measurement and general system parameter update. Exhibit B7 contains the message and lists the data sent from SYS2 to SYS3. As mentioned previously in the discussion of the shared page, SYS1 sends only the system one data area to SYS2 and SYS2 sends only the system two area to system one.

SIMULATION COMMANDS. NTDS Simulation commands may be issued by the instructor computer during interactive teaching. Table B5 contains the command data definition for the message defined in Exhibit B8.

TABLE B5. NTDS SIMULATION COMMAND DATA

INPUT_ID	INPUT_DATA	N	DESCRIPTION
0	0	0	Illegal - do not use.
1-29	0	0	TEC QAB simulated inputs. See Table B6 for identifications
30-122	---	0	TEC simulated buttons - USE WITH CAUTION!
123	see text	0	Alert input. INPUT_DATA values: 1 = SWC Break Engage Order 2 = SWC Engage Order 3 = Poor Sit/Sol Impos 4 = A/C Arrived 5 = Beeper-on-guard 6 = SWC Enage Split Order
124	see text	0	Kill track symbol. INPUT_DATA contains the number of the track to be dropped from PPI display.
125	0	5	Input POS COR via IPC. TEC_DATA[1] is track number, TEC_DATA[2] + [3] is new and TEC_DATA [4] + [5] is new Y (both real)
126	0		Flash SWC com button - SWC is calling.
127-132	0	0	Reserved - do not use.
133	0-1	0	Specify upper or lower labels without simulating button press. 0 = upper 1 = lower
134	0	1	Single-word direct Micronova commands. Contents of TEC_DATA[1] are defined as in Table B7, with this exception: instructions having Byte 1 equal to 135, 136, 137, 190, 192, 193, 198 and 199 shall not be valid inputs.
135	0	5	Load NEP readout. TEC_DATA [i] contains the decimal value of the i-th digit to be displayed.
136	0	18	Load DRO display. TEC_DATA [1] through TEC_DATA [18] contain 36 bytes indicating the number of the display to go in each cell. See Table B8.
137	0-4	40	ID_15_TEC_DATA contains ASCII characters, two per word. The only control characters allowed are <form feed>, which clears the CRT and homes the fictitious cursor, and <CR> which places the fictitious cursor at the beginning of the

TABLE B5. NTDS SIMULATION COMMAND DATA - continued

INPUT_ID	INPUT_DATA	N	DESCRIPTION
			next line. After a <CR> or <FF> a maximum of 80 text characters will be displayed; all others will be ignored until a <CR> or a <FF> is received. Each new string will begin at the point where the previous string stopped. The last byte of the message MUST be 254 (FE), the message terminator for the Micronova; ID_15_TEC_DATA MUST be 42 words in length.
140	0	0	"NTDS DOWN" command input.
141	see text	0	"HOOK SPECIFIED TRACK" command input. INPUT_DATA contains the number of the track to be hooked. If INPUT_DATA = -1, a currently-hooked track will be unhooked.
142	see text	4	"POSITION BALL TAB" command input. TEC_DATA[1] contains the X coordinate (in miles) of the ball tab; TEC_DATA[2] contains the y coordinate. INPUT_DATA determines the display state of the ball tab: 0 = disabled 1 = displayed but disabled 2 = dislayed and enabled
143-159	---	0	Reserved - do not use.
160	0	0	"RESET NTDS" command input.
161	0	0	"SHOEHORN NTDS" command input (valid only after "RESET NTDS").
162	0	0	"RUN NTDS" command input. (Valid only in the IDLE State.)
163	0	0	"FREEZE NTDS" command inputs.
164	0	0	"CONTINUE NTDS" command input. (Valid only after "FREEZE NTDS".)
165	0	0	"ABORT NTDS" command input.

KEYBOARD AND CRT DISPLAY COMMANDS. Keyboard control and page and message commands may be issued by the instructor or speech computer at any time to facilitate student instruction. Among these are commands used by adaptive training control (ATC) to enable and disable special function keys to control their availability; commands used by ATC, speech training, and summary functions to enable page display for any of the teaching segments, menus, or speech prompts; and key input commands used by ATC and summary functions to sequence through page displays, to enable menu selection, and to facilitate simple tests. Exhibit B9 describes the message format used for both the instructor and student consoles and the TEC CRT.

FILE TRANSFER COMMANDS. File transfer commands, as defined in Exhibit B10, will allow files from a disk associated with one computer to be transferred to a disk associated with a different computer. Exhibit B10 contains the MCA synchronization message which allows MCA communication and the error report commands. The error report commands enable SYS1 to log system errors in a common error file and to coordinate a system shutdown if necessary.

ADAPTIVE TRAINING COMMANDS. Adaptive training commands are used to direct and advise adaptive training control. Keyboard inputs, menu requests, performance errors, speech activity status, NTDS status and a variety of other training related activity are reported by the IPCs described in Exhibits B2, B11, and B12.

SUMMARY FUNCTION COMMANDS. Summary function commands forward both student and instructor instigated requests for student statistics from ATC to one of the two summary functions. Keyboard inputs to facilitate menu selection and to satisfy summary function queries are also defined in Exhibit B13 in addition to the summary initiation IPCs.

SPEECH COMMANDS. ATC requests speech collection, validation, practice, voice test, replay, speech recognition, and all other speech activity via the IPCs in Exhibit B14. These commands are issued to initiate speech activity in training segments and supplemental activity.

VOICE GENERATION COMMANDS. ATC issues the voice generation commands in exhibits B6 and B15 during interactive teaching segments using the events defined in Table B3, and Tables B2 and B4, respectively. Speech training utilizes the voice generation commands to elicit verbal trainee responses.

MICRONOVA INTERFACE DEFINITIONS

The Micronova and the simulation processor, SYS2, will be interfaced using an RS232, 9600 baud, full duplex communication line. The Micronova monitors and maintains the status of the TEC buttons, switches, buzzer, lights, LEDs, and alphanumeric presentations. Status information is sent from the Micronova to SYS2 as one word per change, where the upper byte is a code identifier and the lower byte is the related data. Table B6 describes the Micronova to Simulation Computer data.

TABLE B6. MICRONOVA TO SIMULATION COMPUTER DATA

NAME	BYTE 1 DEC	BYTE 1 OCT	BYTE 2
Illegal	0	0	-
VAB 1	1	1	0
VAB 2	2	2	0
VAB 3	3	3	0
VAB 4	4	4	0
VAB 5	5	5	0
VAB 6	6	6	0
VAB 7	7	7	0
VAB 8	8	10	0
VAB 9	9	11	0
VAB 10	10	12	0
VAB 11	11	13	0
VAB 12	12	14	0
VAB 13	13	15	0
VAB 14	14	16	0
VAB 15	15	17	0
VAB 16	16	20	0
VAB 17	17	21	0
VAB 18	18	22	0
Ball Tab Enable	19	23	0
Ball Tab Center	20	24	0
Hook	21	25	0
Sequence	22	26	0
Enter Offset	23	27	0
Drop Track	24	30	0
Enter Mode and Radar	25	31	0
Function Code	26	32	0
Track Number	27	33	0
Height	28	34	0
SIF (reserved)	29	35	0
NED 1	30	36	NED 1 (BCD)
NED 2/3	31	37	NED 2/3 (BCD)
NED 4/5	32	40	NED 4/5 (BCD)
X Ball Tab Update	33	41	Delta X
Y Ball Tab Update	34	42	Delta Y
Range	35	43	Note 1
CRT Center	36	44	Note 2
Standard Leaders (reserved)	37	45	-
Radar Select (reserved)	38	46	-
Radar Video Select (reserved)	39	47	-
Video Brightness	40	50	Note 3
Sweep Brightness	41	51	Note 3
Symbol Brightness	42	52	Note 3
Range Mark Brightness	43	53	Note 3
Talk	44	54	Note 4
Left Phone	45	55	Note 5
Right Phone	46	56	Note 5

TABLE B6. MICRONOVA TO SIMULATION COMPUTER DATA - continued

NAME	BYTE 1 DEC	BYTE 1 OCT	BYTE 2
Footswitch	47	57	Note 6
COMM 1 (Pointer)	48	50	Note 6
COMM 2 (SWC)	49	51	Note 6
COMM 3 (end of switch status updates)	50	52	Note 6
:			
(reserved)			
:			
Illegal Switch Request	112	160	0

Micronova to Simulation Computer Notes:

Note 1: 5 = 16 mile range
6 = 32 mile range
7 = 64 mile range

Note 2: 1 = offset point
2 = ownship

Note 3: 16 levels of brightness (1 - 16), 1 = dimmest

Note 4: 1 = talk left
2 = talk both
3 = talk right

Note 5: 1 = radio
2 = sound powered (reserved)
3 = interconsole

Note 6: 1 = on
2 = off

SYS2 transmits data to the Micronova to control the lights and buzzer, to request status of the TEC, and to provide textual data for display. Except for alphanumeric data to be displayed, the format will consist of one word transmissions where the upper byte contains a code identifier and the lower byte contains the data. Additional alphanumeric data follows, when necessary. The amount is either determined by the code identifier or is detected by an end of data sentinel. Table B7 describes the Simulation Computer to Micronova data.

TABLE B7. SIMULATION COMPUTER TO MICRONOVA DATA

	BYTE 1		
NAME	DEC	OCT	BYTE 2
Illegal	0	0	-
VAB 1 Light	1	1	Note 1
VAB 2 Light	2	2	Note 1
VAB 3 Light	3	3	Note 1
VAB 4 Light	4	4	Note 1
VAB 5 Light	5	5	Note 1
VAB 6 Light	6	6	Note 1
VAB 7 Light	7	7	Note 1
VAB 8 Light	8	10	Note 1
VAB 9 Light	9	11	Note 1
VAB 10 Light	10	12	Note 1
VAB 11 Light	11	13	Note 1
VAB 12 Light	12	14	Note 1
VAB 13 Light	13	15	Note 1
VAB 14 Light	14	16	Note 1
VAB 15 Light	15	17	Note 1
VAB 16 Light	16	20	Note 1
VAB 17 Light	17	21	Note 1
VAB 18 Light	18	22	Note 1
Ball Tab Enable Light	19	23	Note 1
Ball Tab Center Light	20	24	Note 1
Hook Light	21	25	Note 1
Sequence Light	22	26	Note 1
Enter Offset Light	23	27	Note 1
(reserved)	24	30	Note 1
Buzzer	25	31	Note 1
Function Code Light	26	32	Note 1
Track Number Light	27	33	Note 1
Height Light	28	34	Note 1
SIF Light	29	35	Note 1
:			
(Reserved)			
:			
Range LEDs	35	43	Note 2
CRT Center LEDs	36	44	Note 3
Standard Leader LEDs	37	45	Note 4

TABLE B7. SIMULATION COMPUTER TO MICRONOVA DATA - continued

NAME	BYTE 1 DEC	BYTE 1 OCT	BYTE 2
Radar Select LEDs	38	46	Note 5
Radar Video Select LEDs	39	47	Note 6
:			
Talk LEDs	44	54	Note 7
Left Phone LEDs	45	55	Note 8
Right Phone LEDs	46	56	Note 8
.			
COMM 1 Light	48	60	Note 1
COMM 2 Light	49	61	Note 1
COMM 3 Light	50	62	Note 1
:			
(reserved)			
:			
Alert 1 Light	120	170	Note 1
Alert 2 Light	121	171	Note 1
Alert 3 Light	122	172	Note 1
Alert 4 Light	123	173	Note 1
Alert 5 Light	124	174	Note 1
Alert 6 Light	125	175	Note 1
Beeper-on-Guard	126	176	Note 1
Radio In Use Light	127	177	Note 1
True Bearing Light	128	200	Note 1
NEP Clear Light	129	201	Note 1
:			
Load NEP Readout	135	207	Note 17
Load DRO	136	210	Note 18
Load Text CRT	137	211	Note 19
:			
(reserved)			
:			
Load DRO Cell 1	151	227	Note 20
Load DRO Cell 2	152	230	Note 20
Load DRO Cell 3	153	231	Note 20
Load DRO Cell 4	154	232	Note 20
Load DRO Cell 5	155	233	Note 20
Load DRO Cell 6	156	234	Note 20
Load DRO Cell 7	157	235	Note 20
Load DRO Cell 8	158	236	Note 20
Load DRO Cell 9	159	237	Note 20
Load DRO Cell 10	160	240	Note 20
Load DRO Cell 11	161	241	Note 20
Load DRO Cell 12	162	242	Note 20

TABLE B7. SIMULATION COMPUTER TO MICRONOVA DATA - continued

NAME	BYTE 1 DEC	BYTE 1 OCT	BYTE 2
Load DRO Cell 13	163	243	Note 20
Load DRO Cell 14	164	244	Note 20
Load DRO Cell 15	165	245	Note 20
Load DRO Cell 16	166	246	Note 20
Load DRO Cell 17	167	247	Note 20
Load DRO Cell 18	168	250	Note 20
Load DRO Cell 19	169	251	Note 20
Load DRO Cell 20	170	252	Note 20
Load DRO Cell 21	171	253	Note 20
Load DRO Cell 22	172	254	Note 20
Load DRO Cell 23	173	255	Note 20
Load DRO Cell 24	174	256	Note 20
Load DRO Cell 25	175	257	Note 20
Load DRO Cell 26	176	260	Note 20
Load DRO Cell 27	177	261	Note 20
Load DRO Cell 28	178	262	Note 20
Load DRO Cell 29	179	263	Note 20
Load DRO Cell 30	180	264	Note 20
Load DRO Cell 31	181	265	Note 20
Load DRO Cell 32	182	266	Note 20
Load DRO Cell 33	182	267	Note 20
Load DRO Cell 34	184	270	Note 20
Load DRO Cell 35	185	271	Note 20
Load DRO Cell 36	186	272	Note 20
.			
Global VAB Control	188	274	Note 21
Light Test	189	275	Note 22
Request Switch Position	190	276	Note 23
Request Complete Status	191	277	Note 24
Request Display Control Status	192	300	Note 25
Request Communications Status	192	301	Note 26
:			
DRO Test	196	304	Note 27
Text CRT Test	197	305	Note 27
Ball Tab Update Control	198	306	Note 28
Replay Control	199	307	Note 29
⋮			
End of Transmission	254	376	----

Simulation Computer to Micronova Notes: Add 128 to Byte 2 to blink lights.

Note 1: 0 = light/LED off
1 = light/LED on

TABLE B7. SIMULATION COMPUTER TO MICRONOVA DATA - continued

Note 2: 4 = 8 mile range LED on
5 = 16 mile range LED on
6 = 32 mile range LED on
7 = 64 mile range LED on
8 = 128 mile range LED on

Note 3: 0 = all off
1 = offset point LED on
2 = ownship LED on

Note 4: 0 = all off
1 = off LED on
2 = air LED on

Note 5: 0 = all off
1 = off LED on
2 = 1 LED on
3 = 2 LED on
4 = 3 LED on
5 = 4 LED on
6 = 5 LED on
7 = 6 LED on
8 = 7 LED on

Note 6: 0 = all off
1 = off LED on
2 = 1 LED on
3 = 2 LED on
4 = 3 LED on
5 = 4 LED on

Note 7: 0 = all off
1 = talk left LED on
2 = talk both LED on
3 = talk right LED on

Note 8: 0 = all off
1 = sound powered LED on
2 = interconsole LED on
3 = radio LED on

Notes 9 - 16: Reserved

Note 17: Follow the 135 code with 4 bytes containing:

Byte 1 = NED 1 (BCD)
Byte 2 = NED 2/3 (BCD)
Byte 3 = NED 4/5 (BCD)
Byte 4 = 254 (EOT)

TABLE B7. SIMULATION COMPUTER TO MICRONOVA DATA - continued

Note 18: Follow the 136 code with the 36 bytes containing the number of the display to go in each DRO cell. The 38th byte should be 254 for EOT. See Table B8 for cell numbers.

Note 19: Follow the 137 code with ASCII data to display. The CRT is capable of 24 lines by 80 characters. Terminate each line with a carriage return. The first line sent will be displayed on line 1. When reaching the bottom line, each subsequent line will cause the display to scroll up. End the text with a 254 for EOT. A form feed (FF) will erase the screen.

Note 20: Refer to Table B8 for DRO display numbers.

Note 21: 0 = turn all VABs off
1 = turn all VABs on

Note 22: 0 = turn all lights/LEDs off
1 = turn all lights/LEDs on

Note 23: no. = switch/data group number corresponding to byte 1 of change data sent to Simulation Computer. See Table B6 for codes. Data sent back to Simulation Computer is in same format as change data:

30 = NED data
1 = last QAB pushed

Note 24: Switch position data for switches 35 - 50 are sent back one switch at a time in the same format as change data.

Note 25: Send back switch position data on switches 35 - 43.

Note 26: Send back switch position data on switches 44 - 50 (44 - 63 if enabled).

Note 27: 1 = clear DRO or text CRT
2 = display test pattern on DRO or text CRT

Note 28: 0 = cease ball tab updates
1 = start ball tb updates at 50 msec intervals

Note 29: 0 = delete replay mode
1 = replay mode. Implies that Micronova will not send any data to Simulation Computer unless specifically requested to do so.

Table B8 identifies the cell layout of the DRO displays and the readouts that are available for each cell.

TABLE B8. CELL NUMBERING OF DRO DISPLAYS

1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36

a. Universal Displays

000 = Number '0'
001 = Number '1'
002 = Number '2'
003 = Number '3'
004 = Number '4'
005 = Number '5'
006 = Number '6'
007 = Number '7'
010 = Number '8'
011 = Number '9'
377 = Blank Cell

b. Cell 1

012 = ASMD
013 = PAIRED
014 = ERROR
015 = WILCO
016 = CANTCO
017 = CANTPRO
020 = LINK-11
DOWN
021 = LINK-11
NO RESP

c. Cell 2

022 = TN
023 = CT[illegible]
024 = TRACK
QUAL
025 = TTG
MIN/SEC

d. Cell 4

026 = BANK
ANGLE

TABLE B8. CELL NUMBERING OF DRO DISPLAYS - continued

e. Cell 8

027 = SPEED
030 = SPEED
MACH
031 = DESIG
UNIT
032 = ALL ASP
MISSILE
033 = MAG
BRG
034 = CONTROL
UNIT
035 = TARGET
RANGE
036 = CONTRLD
AIRCRAFT

f. Cell 11

037 = V.3 MSG
DISCRETE

g. Cell 13

040 = ENGAGE
041 = BREAK
ENGAGE
042 = CEASE
FIRE
043 = HOLD
FIRE
044 = COVER
045 = SALVO
046 = PRO-TO-
POINT
047 = CS PRO-
TO-POINT
050 = INVEST/
ASSIGN
051 = ENGAGED
052 = LINK-4A
DOWN
053 = TUMA

h. Cell 14

054 = MODE-2
055 = ORD
HEAD

TABLE B8. CELL NUMBERING OF DRO DISPLAYS - continued

056 = A/G WPN
UNGUIDED
057 = RANGE
060 = ANGLE
OFF

i. Cell 16

061 = A/G WPN
062 = QR
NUMBER

j. Cell 18

063 = RIGHT
064 = LEFT

k. Cell 19

065 = PRIORITY
KILL
066 = EMERG
067 = CHECK
TRACK

l. Cell 20

070 = ALT
100 FT
071 = FUEL
072 = REAR
MISSILE
073 = TARGET
BEARING
074 = TIME
075 = LAT/LONG
POSITION

m. Cell 25

076 = HAND-
OVER
077 = AIR
CRAFT
100 = ORD
CHNG
101 = ANY/
UNSPEC
102 = ARRIVED

TABLE B8. CELL NUMBERING OF DRO DISPLAYS - continued

n. Cell 26

103 = COURSE
104 = ORIGIN
UNIT
105 = MLT INT
MISSILE
106 = REPORT
RESPON
107 = RANGE
MILES
110 = ORD
SPEED

o. Cell 27

111 = MAG
NETIC
112 = TACAN

p. Cell 31

112 = POOR
SIT
114 = SOL
IMPOS
115 = REPLY
DATA USED
116 = LINK-4A
117 - LINK-4A
TWO WAY

q. Cell 32

120 = V.3, V.9
121 = V.1, V.2,
V.3

SCENARIO CONTROL OUTPUT FORMATS

A language preprocessor will be used to translate scenario control source commands to produce two binary output files to be used by ACE as input to simulation and performance measurement functional areas.

The translated files are composed of 256 word records which are comprised of 10 word units describing the translated commands. The first word of each word unit contains the time or event upon which the command is to be processed; the second word identifies the command; and the third through tenth words contain data related to the command modifiers.

The files will be separated into a time file and event file. Both of the files will be sorted in numerical order based on the first word of each 10 word unit. Accordingly, the files will be ordered as:

a. Event driven data denoted by negative numbers.

b. Initialization data denoted by zeros.

c. Time driven data denoted by positive numbers corresponding to seconds from exercise start.

Table B9 maps the word locations with their respective data for each scenario command.

TABLE B9. SCENARIO CONTROL TRANSLATION FORMAT

COMMAND	WORD1	WORD2	WORD3	WORD4	WORD5	WORD6	WORD7	WORD8	WORD9	WORD10
INITPOS	0	1	trk #	bearing	range	heading	speed	alt	0	0
RANDPOS	0	2	trk #	bearing	range	heading	0	0	0	0
INITCOND	0	3	trk #	motion model	turn rate	directed heading	video size	NTDS #	trk type	PIF or PU
PACKAGE	0	4	pkg #	0	0	0	0	0	0	0
BACKGROUND	0	5	bkg #	0	0	0	0	0	0	0
ROTSCRN	0	6	degrees	0	0	0	0	0	0	0
ENGAGED	0	7	trk #	trk #	0	0	0	0	0	0
OFFSET	0	8	bearing	range	0	0	0	0	0	0
MODPROF	0	9	pilot	bogey	tracker	SWC	0	0	0	0
FUEL	0	10	trk #	# lbs	0	0	0	0	0	0
CALLS	0	11	vis $	cont $	judy $	tally $	lost $	famish $	0	0
CAPSTN	0	12	bearing	range	alt	0	0	0	0	0
TACSTN	0	13	bearing	range	0	0	0	0	0	0
REFPT	0	14	trk #	bearing	range	0	0	0	0	0
OUTCOME	0	15	#	bogey #s	0	0	0	0	0	0
POINT	0	16	trk #	bearing	range	0	0	0	0	0
TRKON	T or E	17	trk #	0	0	0	0	0	0	0
TRKOFF	T or E	18	trk #	0	0	0	0	0	0	0
CHGHDG	T or E	19	trk #	motion model	turn rate	directed heading	0	0	0	0
JINK	T or E	20	trk #	degrees	duration	0	0	0	0	0
FADE	T or E	21	trk #	# sweeps	0	0	0	0	0	0
SPLIT	T or E	22	trk #	heading	0	0	0	0	0	0
NTDSFAIL	T or E	23	0	0	0	0	0	0	0	0
EMERGENCY	T or E	24	0	0	0	0	0	0	0	0
STRANGER	T or E	25	3	0	0	0	0	0	0	0
FREEZE	T or E	26	0	0	0	0	0	0	0	0
END	T or E	27	0	0	0	0	0	0	0	0
ABORT	time	28	event #	0	0	0	0	0	0	0
RENDEZVOUS	T or E	29	0	0	0	0	0	0	0	0

SEGMENT DEFINITION OUTPUT FORMATS

A language preprocessor will be used to translate segment definition source commands to produce AN ASCII output file for each segment. These translated files will be used by ACE as input for segment training.

The translated file is composed of variable length ASCII records describing the translated commands. The first word contains a command length indicator, the second word contains a command identifier, and any successive words contain command modifiers. A word, as used here, is defined to be a series of ASCII characters delimited by spaces. Each command record is terminated by a carriage return (or line feed).

Table B10 contains a description of the translations generated for the segment definition commands.

TABLE B10. SEGMENT DEFINITIONS TRANSLATION FORMAT

COMMAND (MODIFIERS)	TRANSLATION IN WORD FORMAT WORD 1	WORD 2	ADDITIONAL WORDS
ADV (step #)	3	1	Word 3 - step #
AV/CLR	4	2	Word 3 - 0 4 - 0
AV/F (frame #)	4	2	Word 3 - frame # 4 - 0
AV/N (seq #, chan #)	4	3	Word 3 - sequence # 4 - audio channel #
BEEP	2	4	
BUZZ (#)	3	5	Word 3 = # of seconds
CHAL (step #)	3	6	Word 3 - step #
CK (#)	3	200	Word 3 - check #
CK/A (button, status)	4	202	Word 3 - button # 4 - status #
CK/E	2	205	
CK/H (time limit, second try, referral, presentation, no error, omission, sequence, other, second fail, last button, last status)	13	201	Word 3 - # of seconds 4 - N = 0, Y = 1 5 - N = 0, Y = 1 6 - step # 7 - step # 8 - step # 9 - step # 10 - step # 11 - step # 12 - button # 13 - status #
CK/O (button, status)	4	203	Word 3 - button # 4 - status #
COMM/ON (channel, status)	4	48	Word 3 - comm channel # 4 - ON - 1, OFF = 2

TABLE B10. SEGMENT DEFINITIONS TRANSLATION FORMAT - continued

COMMAND (MODIFIERS)	TRANSLATION IN WORD FORMAT WORD 1	WORD 2	ADDITIONAL WORDS
CP/H (run error freezes, freeze-free runs, PMV failures)	5	301	Word 3 - # of runs 4 - # of runs 5 - # of PMVs
CRT/B (string)	string length + 2	7	Word 3 - string
CRT/D (page #)	3	8	Word 3 - page #
CRT/D/CLR	2	9	
CRT/T (page #)	3	10	Word 3 - page #
CRT/T/CLR	2	11	
DRO (cell, content)	4	13	Word 3 - cell # 4 - content #
DRO/CLR	2	14	
END	2	-1	
FP/H (PMV failures, pass runs)	4	401	Word 3 - # of PMVs 4 - # of runs
GS (phrase type, phrase #, additional data)	additional data + 4	17	Word 3 - type # 4 - phrase # 5+ - data
IAT/H	2	501	
LAB_STATE (state)	3	53	Word 3 - UP = 0, LOW = 1
LED (LED name)	4	18	Word 3 - button or switch # 4 - 1 if button or switch command code or 129 for flash
LED/CLR (LED name)	4	19	Word 3 - button # 4 - 0

TABLE B10. SEGMENT DEFINITIONS TRANSLATION FORMAT - continued

COMMAND (MODIFIERS)	TRANSLATION IN WORD FORMAT WORD 1	WORD 2	ADDITIONAL WORDS
NED (number)	7	20	Word 3 - digit 1 4 - digit 2 5 - digit 3 6 - digit 4 7 - digit 5
NEXT (message #)	3	21	Word 3 - message #
NTDS/D	3	47	Word 3 - 0
PMV/A (PMV #, score) repeat, up to 5 pairs	2 + twice # of PMVs	22	Word 3 - PMV # 4 - score repeat, up to five pairs
PMV/F (PMV #) repeat, up to 5	2 + # of PMVs	23	Word 3 - PMV # Repeat up to five
POSIT/BT (state, bearing, range)	7	52	Word 3 - ball tab state # 4 - real x miles 5 - real y miles
PRESS (button)	4	46	Word 3 - button #
R (step #, step #)	4	25	Word 3 - step # 4 - step #
REC/KBD (key)	3	26	Word 3 - NEXT = -1 YES = -2 NO = -3 or numerical key #
REC/TEC (button, status)	4	27	Word 3 - button or switch # 4 - 0 for button or switch status #
RUN/CP	2	29	
RUN/FP	2	30	
RUN/S	2	31	
SC (scenario #)	3	32	Word 3 - scenario #
SC/C	2	33	
SC/CLR	2	34	

TABLE B10. SEGMENT DEFINITIONS TRANSLATION FORMAT - continued

COMMAND (MODIFIERS)	TRANSLATION IN WORD FORMAT WORD 1	WORD 2	ADDITIONAL WORDS
SC/END	2	35	
SC/W	2	36	
SP/C (element #)	3	37	Word 3 - element #
SP/P (element #)	3	38	Word 3 - element #
SP/V (element #)	3	39	Word 3 - element #
STEP (step #)	3	42	Word 3 - step #
T (#)	3	100	Word 3 - test #
T/E	2	103	
T/H (# questions, score, presentation, no error, pass, first fail, second fail, test type, page #, second try, referral)	13	101	Word 3 - # of questions 4 - passing score % 5 - step # 6 - step # 7 - step # 8 - step # 9 - step # 10 - MATCH = 1, OTHER = 0 11 - match page # 12 - N = 0, Y = 1 13 - N = 0, Y = 1

T/Q (answer type,
 	
presentation,
button,
status,
 	
correct,
incorrect)
	8	102	Word 3 - T/F = -1,
			TEC = 0, else #
			4 - step #
			5 - button # or KBD = 0
			6 - status # or
			NEXT = -1,
			YES = -2,
			NO = -3 or #
			7 - step #
			8 - step #

TEC (alert #)	3	45	Word 3 - alert #
TRKSYM/HK (#)	3	49	Word 3 - track #
TRKSYM/KL (#)	7	50	Word 3 - track # 4 - real x miles 5 - real y miles

TABLE B10. SEGMENT DEFINITIONS TRANSLATION FORMAT - continued

COMMAND (MODIFIERS)	TRANSLATION IN WORD FORMAT WORD 1	WORD 2	ADDITIONAL WORDS
TRKSYM/PC (#)	3	51	Word 3 - track #
WAIT (#)	3	43	Word 3 - # of seconds
W/F	4	44	Word 3 - switch # 4 - status #

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$       SHARED PAGE LITERALS, POINTERS AND BASED DECLARATIONS
$       --------------------------------------------------
$
$       REVISIONS:              G. SLEMON       8-28-80 CUR_MOD RANGE
$
$                               M. GANNIS       10-2-80         ADDED LITERAL DRO_TYPE
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$ BASED VARIABLE AND ARRAY DECLARATIONS:
$ -------------------------------------

        BASED INTEGER I2;
        BASED INTEGER(2) I4;
        BASED REAL R4;
        BASED REAL(4) R8;
        BASED BOOLEAN B2;

        BASED INTEGER ARRAY IARR[0:30];
        BASED INTEGER(2) ARRAY DIARR[0:30];
        BASED REAL ARRAY RARR[0:30];
        BASED STRING (24) STR24;
        BASED STRING (10) STR10;

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ SYSTEM 1:
$ ---------
$
        EXTERNAL INTEGER SYS1PAGE;              $ DEFINED ON THE SHARED PAGE
                                                $ FOR SYS 1 DATA

        EXTERNAL POINTER SYS1STRT;              $ TO USE POINTER, YOU MUST EXECUTE
                                                $ SYS1STRT:=ADDRESS(SYS1PAGE);

        LITERAL SY_USE((SYS1STRT) -> I2);       $ SYSTEM GLOBAL USE COUNT:
                                                $     0: FREE); 1:BUSY
        LITERAL CUR_ST((SYS1STRT+1) -> I2);     $ CURRENT SIMULATOR STATUS:
                                                $     0: RESERVED
                                                $     1: INITIALIZE
                                                $     2: READY
                                                $     3: START
                                                $     4: FREEZE
                                                $     5: UNFREEZE
                                                $     6: STOP CURRENT OPERATION
                                                $     7: STOP, RETURN TO CLI
        LITERAL PREV_ST((SYS1STRT+2) -> I2);    $ LAST STATUS OF ACE, AS DEFINED FOR CUR_ST
        LITERAL CUR_MOD((SYS1STRT+3) -> I2);    $ CURRENT ACE MODE
                                                $     0: DORMANT
                                                $     1: IDLE
                                                $     2: WARMUP
                                                $     3: REVIEW MENU
                                                $     4: SEGMENT EXECUTION
```

Exhibit B1. SHAREDPAGE.LT: Shared Page Variable Definitions

```
                                              $      5: ABORT MENU
                                              $      6: STUDENT STATS
                                              $      7: RETRAIN
                                              $      8: VALIDATION
                                              $      9: VOICE TEST
                                              $     10: ACTIVITY SELECTION
                                              $     11: INITIALIZATION
                                              $     12: CLEANUP FOR SHUTDOWN
                                              $     13: ABORTING (NOT USED)
                                              $     14: OVERRIDE MENU
                                              $     15: BREAK
                                              $     16: RETRAIN MENU
                                              $     17: RETRAIN/VALIDATION MENU
                                              $     18: REPEAT MENU
                                              $     19: HELP
                                              $     20: INSTRUCTOR REFERRAL
                                              $     21: REPLAY
    LITERAL PREV_MO((SYS1STRT+4) -> I2);      $ PREVIOUS ACE MODE (AS FOR CUR_MOD)
    LITERAL INST_NA((SYS1STRT+5) -> STR24);   $ CURRENT INSTRUCTOR NAME
    LITERAL INST_ID((SYS1STRT+17) -> STR10);  $ CURRENT INSTRUCTOR ID
    LITERAL IN_SIGN((SYS1STRT+22) -> I4);     $ WALL CLOCK TIME WHEN INSTR. SIGNED ON
                                              $  IN HRS, MINS
    LITERAL STUD_NA((SYS1STRT+24) -> STR24);  $ CURRENT STUDENT NAME
    LITERAL STUD_ID((SYS1STRT+36) -> STR10);  $ CURRENT STUDENT ID
    LITERAL ST_SIGN_HRS((SYS1STRT+41) -> I2);        $ WALL CLOCK TIME WHEN STUDENT SIGNED ON
    LITERAL ST_SIGN_MINS((SYS1STRT+42) -> I2);       $ IN HRS, MINS
    LITERAL MX_CLCK((SYS1STRT+43) -> I4);     $ MASTER EXERCISE CLOCK (.1 SEC)
    LITERAL SEG_NO((SYS1STRT+45) -> I2);      $ CURRENT SEGMENT, HUNDREDS DIGIT IS LEVEL
    LITERAL SEG_MOD((SYS1STRT+46) -> I2);     $ CURRENT SEGMENT MODE
                                              $      0: DORMANT
                                              $      1: TEACHING
                                              $      2: NTDS RUNNING FOR IAT/AVTT
                                              $      3: NTDS RUNNING FOR CP
                                              $      4: NTDS RUNNING FOR FP
                                              $      5: SCENARIO RUNNING FOR IAT/AVTT
                                              $      6: SCENARIO RUNNING FOR CP
                                              $      7: SCENARIO RUNNING FOR FP
    LITERAL ACE_CLCK((SYS1STRT+47) -> I4);    $ IDLE TIME CLOCK, UNITS AS MX_CLCK
    LITERAL SCEN_RUNNING((SYS1STRT+49) -> B2);$T TRUE WHILE MX_CLCK IS RUNNING

    LITERAL DEBUG_SWITCH((SYS1STRT+50) -> I2);$ 1 IF MINOR_ERRORS ARE TO GO TO SCREEN
    LITERAL OLD_SCEN_RUNNING((SYS1STRT+51) -> B2);   $ TEMPORARY FOR SCEN_RUNNING
    LITERAL KMON_FREEZE((SYS1STRT+52) -> B2);        $ KMON FREEZE IN EFFECT
    LITERAL TEACHING_FREEZE((SYS1STRT+53) -> B2);    $ TEACHING FREEZE IN EFFECT
    LITERAL SIGNOFF_IN_PROGRESS((SYS1STRT+54) ->B2);$ SET FALSE WHEN FILE XFER COMPLETE
$ FREE SPACE STARTS AT SYS1STRT+55
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ SYSTEM 2:
$ ---------
$               LITERALS, POINTERS, AND BASES
$               FOR DATA ITEMS ON THE SHARED PAGE
$               MAINTAINED BY THE TEC PROCESS
$
$               THESE ITEMS SHOULD BE READ-ONLY
```

Exhibit B1. SHAREDPAGE.LT: Shared Page Variable Definitions (Contin.)

```
$               FOR EVERYBODY EXCEPT THE TEC PROCESS
$               AND BASIC SCENARIO CONTROL!
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$ ASSIGNMENT OF SYS 2 SHARED PAGE ARRAY START ADDRESSES:

        EXTERNAL INTEGER TECPAGE;               $ N.B.!
        EXTERNAL POINTER TECSTART;              $ BEFORE USE, EXECUTE
                                                $ TECSTART:=ADDRESS(TECPAGE);
                                                $ BEGINNING OF TEC PORTION OF SHARED PAGE
                                                $ BEGINNING OF TABLE OF MISCELLANEOUS SINGLE ITEMS
                                                $ SEE LITERAL POINTER EXPRESSIONS

$       OFFSETS FOR POINTERS:

        LITERAL TECHEADER((TECSTART-5));        $ MCA HEADER AREA
        LITERAL TECSWITCH((TECSTART + 30));     $ BEGINNING OF TEC SWITCH DATA TABLE
                                                $ USE BASED ARRAY IARR

        LITERAL SYMBL_TYPE((TECSTART + 50));    $ TRACK SYMBOL ARRAY [0:18]
                                                $ USE BASE ARRAY IARR

        LITERAL SYMBL_X((TECSTART + 69));       $ TRACK SYMBOL X COORD ARRAY [0:30]
                                                $ UNITS ARE MILES
                                                $ USE BASED ARRAY RARR

        LITERAL SYMBL_Y((TECSTART + 131));      $ TRACK SYMBOL Y COORD ARRAY [0:30]
                                                $ UNITS ARE MILES
                                                $ USE BASED ARRAY RARR

        LITERAL SYMBL_DX((TECSTART + 193));     $ TRACK SYMBOL X VEL ARRAY [0:13]
                                                $ UNITS ARE MILES/SEC
                                                $ USE BASED ARRAY RARR

        LITERAL SYMBL_DY((TECSTART + 221));     $ TRACK SYMBOL Y VEL ARRAY [0:13]
                                                $ UNITS ARE MILES/SEC
                                                $ USE BASED ARRAY RARR

        LITERAL UPDATE_TIME((TECSTART + 249));  $ TRACK SYMBOL UPDATE TIME ARRAY [0:13]
                                                $ UNITS ARE CLOCK UNITS: 100 = 1 SECOND
                                                $ USE BASED ARRAY DTARR

        LITERAL AC_HDG((TECSTART + 277));       $ TRACK SYMBOL HEADING ARRAY [0:13]
                                                $ UNITS ARE DEGREES TRUE
                                                $ USE BASED ARRAY IARR

        LITERAL AC_SPD((TECSTART + 291));       $ TRACK SYMBOL SPEED ARRAY [0:13]
                                                $ UNITS ARE MACH.TENTHS
                                                $ USE BASED ARRAY IARR

        LITERAL AC_ALT((TECSTART + 305));       $ TRACK SYMBOL ALTITUDE ARRAY [0:14]
```

Exhibit B1. SHAREDPAGE.LT: Shared Page Variable Definitions (Contin.)

```
                                         $ UNITS ARE HUNDREDS OF FEET
                                         $ USE BASED ARRAY IARR
      LITERAL AC_TN((TECSTART + 320));   $ TRACK SYMBOL NTDS TRACK ID NUMBER ARRAY [0:13]
                                         $ 0 TO 17777 - OCTAL DIGITS
                                         $ USE BASED ARRAY IARR
      LITERAL AC_SIF((TECSTART + 334));  $ TRACK SYMBOL SIF (PIF) CODE ARRAY [0:13]
                                         $ 0 TO 177777 - OCTAL DIGITS
                                         $ USE BASED ARRAY IARR
      LITERAL AC_FUEL((TECSTART + 348)); $ TRACK SYMBOL ASSIGNED FUEL ARRAY [0:12]
                                         $ UNITS ARE HUNDREDS OF POUNDS
                                         $ USE BASED ARRAY IARR
      LITERAL AC_ENG((TECSTART + 361));  $ TRACK SYMBOL ENGAGEMENT ARRAY [0:12]
                                         $ VALUE OF ARRAY ELEMENT I IS THE TRACK
                                         $ TO WHICH TRACK I IS ENGAGED
                                         $ USE BASED ARRAY IARR
      LITERAL AC_STAT((TECSTART + 374)); $ TRACK SYMBOL ENGAGEMENT STATUS ARRAY [0:12]
                                         $    0 = NOT ENGAGED
                                         $    1 = TRIAL GEOMETRY
                                         $    2 = AVS
                                         $ USE BASED ARRAY IARR
      LITERAL LAST_BEARING((TECSTART + 387)); $ VALUE OF ARRAY ELEMENT I IS THE LAST BEARING
                                         $ VALUE DISPLAYED WHILE TRACK SYMBOL I
                                         $ WAS HOOKED
                                         $ ARRAY [0:18]
                                         $ USE BASED ARRAY IARR
      LITERAL LAST_RANGE((TECSTART + 406));  $ VALUE OF ARRAY ELEMENT I IS THE LAST RANGE
                                         $ VALUE DISPLAYED WHILE TRACK SYMBOL I
                                         $ WAS HOOKED
                                         $ ARRAY [0:18]
                                         $ USE BASED ARRAY IARR
      LITERAL OGD_HDG((TECSTART + 425)); $ TRACK SYMBOL INTERCEPT GEOMETRY HEADING ARRAY [0:12]
                                         $ UNITS ARE DEGREES TRUE
                                         $ USE BASED ARRAY IARR
$     FREE SPACE STARTS AT TECSTART + 438

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ LITERAL POINTER EXPRESSIONS FOR MISCELLANEOUS INDIVIDUAL SYS 2 SHARED PAGE DATA ITEMS:

      LITERAL OFFSET_X((TECSTART) -> R4);          $ PPI X OFFSET, IN MILES
      LITERAL OFFSET_Y((TECSTART + 2) -> R4);      $ PPI Y OFFSET, IN MILES
      LITERAL HOOKED_TRACK((TECSTART + 4) -> I2);  $ NUMBER OF TRACK IN CLOSE CONTROL
```

Exhibit B1. SHAREDPAGE.LT: Shared Page Variable Definitions (Contin.)

```
LITERAL POINTED_TRACK((TECSTART + 5) -> I2);    $ NUMBER OF TRACK SYMBOL IN POINTER SQUARE

LITERAL BT_X((TECSTART + 6) -> R4);             $ X COORD OF BALL TAB, IN MILES

LITERAL BT_Y((TECSTART + 8) -> R4);             $ Y COORD OF BALL TAB, IN MILES

LITERAL CROSS_X((TECSTART + 10) -> R4);         $ X COORD OF INTERCEPT POINT, IN MILES

LITERAL CROSS_Y((TECSTART + 12) -> R4);         $ Y COORD OF INTERCEPT POINT, IN MILES

LITERAL TRIAL_HDG((TECSTART + 14) -> I2);       $ CURRENT TRIAL HEADING, IN DEGREES

LITERAL BT_STATUS((TECSTART + 15) -> I2);       $ ACTIVITY STATUS OF BALL TAB:
                                                $ 0 = OFF
                                                $ 1 = DISPLAYED, NOT ENABLED
                                                $ 2 = ENABLED AND DISPLAYED

LITERAL BOG_RANGE((TECSTART + 16) -> R4);       $ BEEPER-ON-GUARD POINT RANGE FROM O/S, IN MILES

LITERAL BOG_BEAR((TECSTART + 18) -> I2);        $ BEEPER-ON-GUARD POINT BEARING FROM O/S, IN DEGREES TRUE

LITERAL PP_BEAR((TECSTART + 19) -> I2);         $ LATEST BEARING DISPLAYED IN PPIRO

LITERAL PP_RANGE((TECSTART + 20) -> I2);        $ LATEST RANGE DISPLAYED IN PPIRO

LITERAL PP_TRACK((TECSTART + 21) -> I2);        $ TRACK NUMBER OF ORIGIN FOR PPI DATA
LITERAL CAP_BANK((TECSTART + 22) -> I2);        $ CAP'S BANK ANGLE, IN DEGREES

LITERAL BOGEY_BANK((TECSTART + 23) -> I2);      $ BOGEY'S BANK ANGLE, IN DEGREES

LITERAL DRO_TYPE((TECSTART + 24) -> I2);        $ TYPE OF CURRENTLY-DISPLAYED DRO


$ SPECIFIC ARRAY ELEMENTS NEEDED FOR INITIALIZATION:

    LITERAL CAP_FUEL((AC_FUEL + 1) -> I2);      $ CAP FUEL ASSIGNED VALUE, IN 100-LB UNITS

    LITERAL TARGET_1((AC_ENG + 1) -> I2);       $ VALUE IS NUMBER OF TRACK ENGAGED BY
                                                $ TRACK 1; -1 = NO ENGAGEMENT

    LITERAL TARGET_2((AC_ENG + 2) -> I2);       $ VALUE IS NUMBER OF TRACK ENGAGED BY
                                                $ TRACK 2; -1 = NO ENGAGEMENT

$ TEC SWITCH DATA TABLE - NAMES OF INDIVIDUAL SWITCH DATA ITEMS:

    LITERAL $NED1((TECSWITCH) -> I2);           $ NED 1
                                                $ HI BYTE = 0, LO BYTE = NED 1

    LITERAL $NED23((TECSWITCH + 1) -> I2);      $ NED 2 AND NED 3 - BCD
                                                $ HI BYTE = NED 2, LO BYTE = NED 3
```

Exhibit B1. SHAREDPAGE.LT: Shared Page Variable Definitions (Contin.)

```
        LITERAL $NED45((TECSWITCH + 2) -> I2);        $ NED 4 AND NED 5 - BCD
                                                      $ HI BYTE = NED 4, LO BYTE = NED 5

        LITERAL $RANGE((TECSWITCH + 3) -> I2);        $ RANGE SWITCH
                                                      $ 5 = 16 MILES, 6 = 32 MILES, 7 = 64 MILES

        LITERAL $CENTER((TECSWITCH + 4) -> I2);       $ CRT CENTER SWITCH
                                                      $ 1 = OFFSET POINT, 2 = OWNSHIP

        LITERAL $STDLDRS((TECSWITCH + 5) -> I2);      $ STANDARD LEADERS SELECT
                                                      $ 1 = OFF, 2 = AIR, 3 = SEA

        LITERAL $RADARSEL((TECSWITCH + 6) -> I2);     $ RADAR SELECT
                                                      $ 1 = OFF, 2 = RADAR 1, 3 = RADAR 2,
                                                      $ ... 8 = RADAR 7

        LITERAL $VIDEOSEL((TECSWITCH + 7) -> I2);     $ RADAR VIDEO SELECT
                                                      $ 1 = OFF, 2 = VIDEO 1, 3 = VIDEO 2,
                                                      $ ... 8 = VIDEO 7

        LITERAL $VIBRI((TECSWITCH + 8) -> I2);        $ VIDEO BRIGHTNESS
                                                      $ 0 TO 15; 15 = BRIGHTEST

        LITERAL $SWBRI((TECSWITCH + 9) -> I2);        $ SWEEP BRIGHTNESS
                                                      $ 0 TO 15; 15 = BRIGHTEST

        LITERAL $SYBRI((TECSWITCH + 10) -> I2);       $ SYMBOL BRIGHTNESS
                                                      $ 0 TO 15; 15 = BRIGHTEST

        LITERAL $RMBRI((TECSWITCH + 01) -> I2);       $ RANGE MARK BRIGHTNESS
                                                      $ 1 TO 15; 15 = BRIGHTEST

        LITERAL $TALKSW((TECSWITCH + 12) -> I2);      $ 1 = TALK LEFT, 2 = TALK BOTH, 3 = TALK RIGHT
        LITERAL $LPHONE((TECSWITCH + 13) -> I2);      $ 1 = RADIO, 2 = SOUND POWERED,
                                                      $ 3 = INTERCONSOLE

        LITERAL $RPHONE((TECSWITCH + 14) -> I2);      $ 1 = RADIO, 2 = SOUND POWERED,
                                                      $ 3 = INTERCONSOLE

        LITERAL $FOOTSW((TECSWITCH + 15) -> I2);      $ 1 = ON, 2 = OFF

        LITERAL $COMM1((TECSWITCH + 16) -> I2);       $ COMM 1 (POINTER)
                                                      $ 1 = ON, 2 = OFF

        LITERAL $COMM2((TECSWITCH + 17) -> I2);       $ COMM 2 (SWC)
                                                      $ 1 = ON, 2 = OFF, 3 = FLASHING
$       TWO SPARE WORDS:
$               TECSWITCH + 18
$               TECSWITCH + 19

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$ EXTERNALS, POINTERS AND LITERALS FOR THE TRACK DATA TABLE PORTION
```

Exhibit B1. SHARED PAGE.LT: Shared Page Variable Definitions (Contin.)

```
$ OF THE SYSTEM 2 SHARED PAGE
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

        EXTERNAL INTEGER        TDT;                        $DEFINED ON THE SHARED PAGE FOR SYS2 TDT DATA
        EXTERNAL POINTER        TDT_PTR;                    $ POINTER TO THE TRACK DATA TABLE

        $************************** BASED TYPE DEFINITIONS ***************************

        BASED BOOLEAN           BB;
        BASED INTEGER           BI;
        BASED REAL              BR;
        BASED INTEGER ARRAY     BIARR[0:13];
        BASED REAL ARRAY        BRARR[0:13];
        BASED BOOLEAN ARRAY     BBARR[0:13];
        BASED STRING (4)        BS4;

        $************ POINTERS TO INFORMATION OF THE TRACK DATA TABLE ***************

        LITERAL         TRK_ON(TDT_PTR)                 $ BOOLEAN INDICATES WHETHER A TRACK IS ON
                                                        $ TRUE = TRACK ON
                                                        $ FALSE= TRACK OFF

                       ,VID_ON(TDT_PTR+14)              $ BOOLEAN INDICATES WHETHER A VIDEO IS ON
                                                        $ TRUE  = VIDEO IS VISIBLE
                                                        $ FALSE = VIDEO IS FADED

                       ,VID_MOTION(TDT_PTR+28)          $ INTEGER - INDICATES TRACK'S MOTION TYPE
                                                        $ 1 = SIMPLE (STRAIGHT)
                                                        $ 2 = TURNING
                                                        $ 3 = ORBIT
                                                        $ 4 = STATIONARY

                       ,VID_HEADING(TDT_PTR+42)         $ REAL - INDICATES TRACK'S HEADING

                       ,VID_DIR_HEAD(TDT_PTR+70)        $ REAL - INDICATES TRACK'S DIRECTED HEADING

                       ,VID_ALTITUDE(TDT_PTR+98)        $ INTEGER - INDICATES TRACK'S ALTITUDE

                       ,VID_ANG(TDT_PTR+112)            $ INTEGER - INDICATES ANGLE AT WHICH VIDEO IS TO BE DISPLAYED

                       ,VID_RNG(TDT_PTR+126)            $ INTEGER - RANGE IN MEG UNITS AT WHICH VIDEO IS TO BE DISPLAYED

                       ,VID_SIZE(TDT_PTR+140)           $ INTEGER - INDICATES SIZE OF VIDEO
                                                        $ 1 = SMALL (2 DEG)
                                                        $ 2 = MEDIUM (4 DEG)
                                                        $ 3 = LARGE (6 DEG)

                       ,VID_OFFANG(TDT_PTR+154)         $ INTEGER - INDICATES ANGLE AT WHICH VIDEO IS TO BE TURNED OFF

                       ,VID_TURN_RATE(TDT_PTR+168)      $ REAL - INDICATES TRACK'S TURN RATE

                       ,VID_SPEED(TDT_PTR+196)          $ REAL - INDICATES TRACK'S SPEED
```

Exhibit B1. SHAREDPAGE.LT: Shared Page Variable Definitions (Contin.)

```
                    ,VIDEO_X(TDT_PTR+224)          $ REAL - INDICATES TRACK'S X REC COORDINATE IN MILES

                    ,VIDEO_Y(TDT_PTR+252)          $ REAL - INDICATES TRACK'S Y REC COORDINATE IN MILES

                    ,VIDEO_DX(TDT_PTR+280)         $ REAL - INDICATES TRACK'S DELTA X IN MILES/SEC

                    ,VIDEO_DY(TDT_PTR+308)         $ REAL - INDICATES TRACK'S DELTA Y IN MILES/SEC

                    ,DEG_CNT((TDT_PTR+336) -> BI)   $ INTEGER - INDICATES THE NUMBER OF 1/2 DEGREE COUNTS 1-720
                    ,DISK_ID ((TDT_PTR+337) -> BS4) $ STRING (4) FOR DISK ID ON SYS2
                    ;

$
$ END OF SHAREDPAGE.LT
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Exhibit B1. SHAREDPAGE.LT: Shared Page Variable Definitions (Contin.)

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$     NAME:           IPC_FORMATS.LT
$                     --------------
$
$     FUNCTION:       "INCLUDE" FILE -
$                     DEFINES LITERALS TO PROVIDE OFFSETS AND CONSTANTS FOR
$                     IPC AND MCA COMMUNICATIONS
$
$     CALLING SEQUENCE:
$
$                     INCLUDE IPC_FORMATS.LT;
$
$     REVISION HISTORY:
$
$             OTHERS BEFORE ME, I KNOW NOT WHEN
$
$             M. GANNIS       AUGUST 5, 1980          ACE
$             J. VOIT         AUGUST 6, 1980
$                             CHANGED ID 6 LITERALS
$             G. SLEMON       AUGUST 6, 1980
$                             CLEANED UP ID 1-4,208
$             M. HICKLIN      AUGUST 12: ADDED IPC_HDR_LEN
$                 "           AUGUST 13: RE-ADDED SOME ID_1** IPCS WHICH
$                             WERE DELETED
$                 "           AUGUST 15: ADDED ID_17* DEFNS
$             G. SLEMON       AUGUST 19: EMBELLISHED COMMENTS
$                             ,CHANGED MSG 200, 203, 204
$                             ,DELETED MSG 252, 253, 254, 301, 302
$                             AND MADE SURE DATA OFFSETS INCLUDE
$                             DATA_START.
$                 "           AUGUST 21: REMOVED ATC VOIT MESSAGES
$                             AND MSG 206 AND 207. ALSO,,MOVED ATC
$                             ONLY IPCS TO IPC0200.LT (MSG 201, 202,
$                             205, 208, 209, 242, 243)
$                             ALSO, ADDED AN OPTION TO MSG 260.
$             D. HARRY        AUGUST 25: MOVED INFORMATION FOR
$                             MESSAGES 1, 5, 6, 15, 100, 150, 260
$                             AND 2300-2306 TO SEPARATE INCLUDE FILES
$                             (NAMED IPC####.LT)
$                             ALSO INCLUDED INFORMATION  BOUT SPEECH
$                             REQUESTS (MSG 2100).
$             M. HICKLIN      ADDED INCLUDE IPC0008.LT
$             G. SLEMON       ADDED A LITERAL FOR SIGNOFF TO IPC0001.LT
$             G. SLEMON       SEPT.4, 1980
$                             ADDED IPC 260 LITERALS FROM IPC0200.LT
$             M. HICKLIN      CREATED IPC0010.LT
$             G. SLEMON       9-9-80: ADDED IPC 300 LITERALS
$                                     ADDED IPC 6 LITERAL FOR SCEN #
$             G. SLEMON       9-12-80:RELEASE IPC0100.LT AS SHOWN IN
$                                     KNOW MEMO AND CHANGED INIT MSG
$                                     FORMAT.  IPC0100.LT SHOULD BE
$                                     INCLUDED EXPLICITLY IF USED
$                                     SINCE IT IS A RATHER LONG LITERAL
$                                     FILE.
```

Exhibit B2. IPC_FORMATS.LT: MCA/IPC Summary and Adaptive Training Commands

```
$               G. SLEMON       9-15-80:CHANGED IPC0100.LT
$               H. LEE          9/22/80: ADDED COMMENT REGARDING MESSAGES 401,
$                               402,411,412 WHICH ARE DECLARED IN IPC0400.LT
$               D. HARRY        9-30-80: ADDED LITERAL FOR SPEECH REQUESTS
$                               "ID_2100_SPEECH_REQUEST".
$               M. HICKLIN      10-8-80: ADDED SYS3_MCA_LEN
$               G. SLEMON       10-28-80: CHANGED ID_200_MAX_DMSG VALUE
$                               10-29-80: ADDED A LITERAL FOR NEW T/E IN MSG 200
$               G. SLEMON       11-4-80: ADDED LITERAL TO IPC 1 FOR FRZ CLR
$               G. SLEMON       11-5-80: ADDED COMMENTS TO IPC 244
$               H. LEE          11-9-80: ADDED NEW DATA_START+1 FOR MSGID# 204
$               G. SLEMON       11-17-80: ADDED NEW SFE FOR MSGID# 200
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
$
$       NOTE:   ALL UPDATES TO THIS FILE SHALL USE THE SYSTEM 1 COPY AS THE
$               MASTER COPY!
$
$               ALL UPDATES TO THIS FILE SHALL BE LOGGED ABOVE!  (WE GOT ENOUGH
$               TROUBLES AS IT IS, WITHOUT HAVING INCOMPATIBLE INCLUDE FILES
$               RUNNING AROUND.)
$
$@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@


$ IPC HEADER OFFSETS:

        LITERAL
         ?ISFL  (0)     $ SYSTEM FLAGS
        ,?IUFL  (1)     $ USER FLAGS
        ,?IDPH  (2)     $ HIGH WORD OF GLOBAL DESTINATION PORT NUMBER
        ,?IOPH  (2)     $ HIGH WORD OF GLOBAL ORIGIN PORT NUMBER
        ,?IDPL  (3)     $ LOW WORD, DESTINATION
        ,?IOPL  (3)     $ LOW WORD, ORIGIN
        ,?IOPN  (4)     $ LOCAL SEND PORT
        ,?IDPN  (4)     $ LOCAL DESTINATION PORT
        ,?ILTH  (5)     $ LENGTH OF MESSAGE IN WORDS
        ,?IPTR  (6)     $ ADDRESS OF MESSAGE
        ;

$ IPC MESSAGE OFFSETS:
$
        LITERAL
         ALT_DEST       (0)     $ RESERVED
        ,MSG_ID         (1)     $ MESSAGE ID
        ,DATA_LENGTH    (2)     $ LENGTH GOES IN UPPER BYTE
        ,SEND_ID        (2)     $ SENDER'S LOCAL PORT GOES IN LOWER BYTE
        ,TIME_TAG       (3)     $ ACE_CLCK FROM SHARED PAGE INTO WORDS 3 & 4
        ,IPC_HDR_LEN    (5)     $ LENGTH OF THE FIXED PORTION OF THE IPC HEADER
        ,DATA_START     (5)     $ VARIABLE DATA STARTS HERE
        ,MX_TIME        (DATA_START)    $ MASTER EXERCISE TIME WHEN INCLUDED
                                        $ AS THE FIRST ENTRY IN DATA AREA
        ,EVNT_NUM       (DATA_START+2)  $ EVENT NUMBER ASSOCIATED WITH IPC
                                        $ 5,6,7,8,2300 & 2301
```

Exhibit B2. IPC_FORMATS.LT: MCA/IPC Summary and Adaptive Training Commands (Contin.)

```
        $
        LITERAL SYS3_MCA_LEN    (300);  $ FIXED LENGTH MESSAGE SIZE FOR SYS3
$
$ DATA FORMATS FOR MCA MESSAGES
$
$       STATE OF THE WORLD - MSG ID 1
$
        INCLUDE IPC0001.LT;
$
$
$       NTDS EVENTS - MSG ID 5
$
        INCLUDE IPC0005.LT;
$
$       COMPLEX SYSTEM EVENTS - MSG ID 6
$
        INCLUDE IPC0006.LT;
$
$       VOTRAX REQUESTS - MSG ID 8
$
        INCLUDE IPC0008.LT;
$
$       SHARED PAGE UPDATE ID'S 10 AND 11
$
        INCLUDE IPC0010.LT;
$       NTDS INPUTS - MSG ID 15
$
        INCLUDE IPC0015.LT;
$
$       EMON-RECEIVED MESSAGES - MSG ID 100
$
$$$$    INCLUDE IPC0100.LT;     $ PLEASE INCLUDE THIS FILE EXPLICITLY IF USED
$
$       EXEC RECEIVED MESSAGES - MSG ID 150, 160, 170
$
        INCLUDE IPC0150.LT;
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$       ATC RECEIVED MESSAGES
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$       ATC CONTROL MESSAGES USES THIS COMMON LITERAL REFERENCE

LITERAL ID_200_MSG_CODE (DATA_START);   $ ATC CONTROL MESSAGE CODE

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGES 200-209 ARE CONTROL MESSAGES
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MAX_CMSG (10);   $ THERE IS A RANGE OF 10 CONTROL MESSAGES

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 200: ATC SPK NOTIFICATION FROM EMON
```

Exhibit B2. IPC_FORMATS.LT: MCA/IPC Summary and Adaptive Training Commands (Contin.)

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_SFK (200),           $ MSG_ID
        ID_200_FIRST_CMSG (200);        $ THIS IS THE FIRST CONTROL MESSAGE
                                $    IF THIS CHANGES THEN MAX_CMSG MUST, TOO.

LITERAL ID_200_ATC_SFK (DATA_START),            $ SPECIAL FUNCTION KEY ID OFFSET
$       ITS VALUES ARE:
        ID_200_SSTP_MSG (0),                    $ SHIFT STOP KEY
        ID_200_ABRT_MSG (1),                    $ ABORT KEY
        ID_200_HELP_MSG (2),                    $ HELP KEY
        ID_200_RPLY_MSG (3),                    $ REPLAY KEY
        ID_200_INVT_MSG (4),                    $ INIT VOICE TEST KEY
        ID_200_STVT_MSG (5),                    $ STOP VOICE TEST KEY
        ID_200_OVRD_MSG (6),                    $ OVERRIDE KEY
        ID_200_BREK_MSG (7),                    $ BREAK KEY
        ID_200_CONT_MSG (8),                    $ CONTINUE KEY
        ID_200_RTRN_MSG (9),                    $ RETRAIN KEY
        ID_200_STTS_MSG (10),                   $ (INSTRUCTOR) STATS KEY
        ID_200_BYEE_MSG (13),                   $ BYE KEY
        ID_200_NEWT_MSG (14),                   $ NEW T/E KEY

        ID_200_SRC_KBD (DATA_START+1),  $ SOURCE KEYBOARD OFFSET
$       ITS VALUES ARE:
        ID_200_STU_KBD (1),                     $ STUDENT CONSOLE
        ID_200_INS_KBD (2);                     $ INSTRUCTOR CONSOLE

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGES 201, 202 ARE DEFINED IN IPC0200.LT
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 203: SYSTEM COMMANDS FROM THE EXEC
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_EXC (203);           $ MSG_ID

LITERAL ID_200_CMD_ID (DATA_START),             $ COMMAND NUMBER OFFSET
$       ITS VALUES ARE:
$                       MSG_ID 1, ID_1_SD_SIGNON IS TRANSLATED TO THIS:
        ID_200_SION_MSG (23),                   $ STUDENT HAS SIGNED ON
        ID_200_STOP_MSG (24),                   $ SHUTDOWN AFTER SEGMENT
$                       MSG_ID 1, ID_1_SD_STOP IS TRANSLATED TO THIS:
        ID_200_HALT_MSG (25);                   $ SHUTDOWN IMMEDIATELY

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 204: STATISTICS SHOWING COMPLETED FROM ISSTSSUM & ISSTUSUM
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_STM (204);           $ MSG_ID

$THE FOLLOWING ARGUMENT IS FOR THE MSG_CODE OFFSET

LITERAL ID_200_STEM_MSG (26);                   $ STATISTICS COMPLETED
```

Exhibit B2. IPC_FORMATS.LT: MCA/IPC Summary and Adaptive Training Commands (Contin.)

```
LITERAL ID_200_WHOSE_STATS (DATA_START+1),      $ OFFSET TELLS WHAT IS DONE
$       ITS VALUES ARE:
        ID_200_STU_SUM (1),                     $ STUDENT SUMMARY DONE
        ID_200_INS_SUM (2),                     $ INSTRUCTOR SUMMARY DONE
        ID_200_NEW_TE (3);                      $ NEW T/E (REGISTRATION) DONE

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGES 205, 208, AND 209 ARE IN IPC0200.LT
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       THERE ARE PRESENTLY 14 DATA MESSAGES RECEIVED AS IPCS BY ATC.
$         TEC PROCESS SENDS A MESSAGE WITH ID = 5, 1361, 1362.
$         DSC SENDS MESSAGE WITH ID = 6 FOR SCENARIO END.
$         THE EXEC SENDS MESSAGE WITH ID = 151 FOR FILE TRANSFER COMPLETE.
$       THEREFORE, THE FOLLOWING NUMBERS ARE RECEIVED AS IPCS:
$         5, 6, 151, 240, 242-244, 250-251, 260, 270-271, 1361, 1362
$         (VERIFY THIS WITH IACOM.SB)
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MAX_DMSG (14);   $ MAXIMUM NUMBER OF DATA IPCS


$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 240: REPLAY COMPLETED MESSAGE
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_RPY (240);           $ MSG_ID

$NO DATUM IS ASSOCIATED WITH THIS MESSAGE

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGES 242, 243 ARE DEFINED IN IPC0200.LT
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 244: REQUESTED KEY INPUT FROM EMON TO ACTIVITY
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_KEY (244);           $ MSG_ID

LITERAL ID_200_KEY_INPUT (DATA_START),  $ INPUT FROM KEYBOARD

        ID_200_NEXT_KEY (-1),                           $ -1 NEXT ENTERED
        ID_200_YES_KEY (-2),                            $ -2 YES ENTERED
        ID_200_NO_KEY (-3);                             $ -3 NO ENTERED

$       THE INPUT CAN ALSO CONSIST OF A NUMERICAL VALUE.


$THE SECOND DATA WORD IS SRC_KBD AS DESCRIBED FOR MESSAGE 200

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 250: PM READY
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Exhibit B2. IPC_FORMATS.LT: MCA/IPC Summary and Adaptive Training Commands (Contin.)

```
LITERAL ID_200_MES_PMR (250);           $ MSG_ID

$NO DATUM IS ASSOCIATED WITH THIS MESSAGE

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 251: SPEECH UNDERSTANDING READY
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_SPR (251);           $ MSG_ID

$NO DATUM IS ASSOCIATED WITH THIS MESSAGE

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 260: SPEECH ACTIVITY COMPLETED FROM SPEECH
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

        INCLUDE IPC0260.LT;

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 270: PM RUN ERROR DETECTED FROM PM
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_PME (270);           $ MSG_ID

LITERAL ID_200_MX_TM (DATA_START),      $ EXERCISE CLOCK TIMER

        ID_200_PMV_ID (DATA_START+2),   $ PMV NUMBER

        ID_200_PMV_PART (DATA_START+3), $ PMV PART NUMBER IN ERROR

        ID_200_PMV_LOSS (DATA_START+4), $ POINTS LOST DUE TO THE ERROR

        ID_200_PMV_MSG (DATA_START+5);  $ ERROR MESSAGE NUMBER

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 271: PMVC FILE COMPOSED
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_PMC (271);           $ MSG_ID

$NO DATUM IS ASSOCIATED WITH THIS MESSAGE

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ PERFORMANCE MEASUREMENT (IP) RECEIVED CONTROL MESSAGE ID'S
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_300_SETUP    (300),  $ LIST OF PROCEDURES TO BE RUN
        ID_300_NUM_PRO  (DATA_START),    $ OFFSET FOR # OF PMV PROCEDURES TO RUN
        ID_300_PRO_START(DATA_START+1), $ OFFSET FOR START OF PROCEDURE NUMBERS
        ID_300_PERIODIC (303);  $ PERIODIC UPDATE OF SHARED PAGE OCCURRED
```

Exhibit B2. IPC_FORMATS.LT: MCA/IPC Summary and Adaptive Training Commands (Contin.)

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$       MESSAGES 401, 402, 411, AND 412 ARE IN IPC0400.LT
$
$$$$             MUST INCLUDE THIS FILE EXPLICITLY IN PROGRAM
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$---------------------------------------------------------------------------
$       HANDSHAKE: NTDS-HAS-BEEN-RESET - MSG ID 1361
$---------------------------------------------------------------------------
        LITERAL
         ID_1361_NTDS_RESET      (1361)  $ MSG_ID VALUE
        ;
$---------------------------------------------------------------------------
$       HANDSHAKE: NTDS-IS-RUNNING - MSG ID 1362
$---------------------------------------------------------------------------
        LITERAL
         ID_1362_NTDS_RUNNING   (1362)  $ MSG_ID VALUE
        ;
$
$       SPEECH REQUESTS - MSG ID 2100
$
        INCLUDE IPC2100.LT;
$
$       SWC VOXBOX REQUESTS ISSUED BY THE SWC MODEL - MSG ID 2300
$       PSUEDO BOGEY & MAC VOXBOX REQUESTS (2ND PILOT'S VOICE) - MSG ID 2301
$       DEMO MESSAGE REQUESTS FOR VOXBOX - MSG ID 2303
$       IAT MESSAGE REQUESTS FOR VOXBOX - MSG ID 2304
$       AIC MESSAGE VOCALIZATION REQUESTS (SPEECH TRAINING) - MSG ID 2306
$
        INCLUDE IPC2300.LT;

$ END OF IPC_FORMATS.LT
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Exhibit B2. IPC_FORMATS.LT: MCA/IPC Summary and Adaptive Training Commands (Contin.)

```
$
$       STATE OF THE WORLD - MSG ID 1
$
$
$             G. SLEMON        11/4/80          ADDED ID_1_SD_SIGNOFF
$
        LITERAL
         ID_1_STATUS_DATA        (1)      $ MSG_ID
        ,ID_1_STAT_WORD (DATA_START)     $ DATA WORDS BELOW GO AT THIS OFFSET
        ,ID_1_SD_INIT            (1)      $ DATA WORDS FOR ID_1_STAT_WORD
        ,ID_1_SD_READY           (2)
        ,ID_1_SD_SIGNON          (3)
        ,ID_1_SD_FREEZE          (4)
        ,ID_1_SD_CONTINUE        (5)
        ,ID_1_SD_ABORT           (6)
        ,ID_1_SD_STOP            (7)
        ,ID_1_SD_SIGNOFF         (8)
        ,ID_1_SD_CLR_FRZ         (9)
        ,ID_1_GRP       (DATA_START+1)  $ SON'S GLOBAL RECEIVE PORT OFFSET
        ,ID_1_LRP       (DATA_START+3)  $ SON'S LOCAL RECEIVE PORT OFFSET
        ;
$
```

Exhibit B3. IPC0001.LT: Status Data

```
$
$       NTDS EVENTS - MSG ID 5
$
        LITERAL
         ID_5_NTDS_EVENT          (5)      $ MSG_ID VALUE
$       ,USES MI_TIME AND EVNT_NUM OFFSETS AS DEFINED PREVIOUSLY
        ,ID_5_EVENT_DATA          (DATA_START+3)  $ OFFSET OF DATA ASSOCIATED WITH EVENT
        ;
$
```

Exhibit B4. IPC0005.LT: Micronova Reported Switch Button Action Events

```
$
$       COMPLEX SYSTEM EVENTS - MSG ID 6
$
        LITERAL
         ID_6_SYS_EVNT            (6)       $ MSG_ID VALUE
        ,ID_6_TRACK_ID            (DATA_START+3)  $ TRACK IDENTIFICATION NUMBER OFFSET
        ,ID_6_SCENARIO_NUMBER     (DATA_START+3)  $ SCENARIO NUMBER TO BE INITED
        ,ID_6_X1                  (DATA_START+4)  $ X COORDINATE OFFSET(2 WORDS)
        ,ID_6_Y1                  (DATA_START+6)  $ Y COORDINATE OFFSET(2 WORDS)
        ;
$
```

Exhibit B5. IPC0006.LT: Scenario Events

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ MESSAGE ID 8 LITERALS
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

        LITERAL
         ID_8_VRO_SPEAK (8)             $ MESSAGE ID
        ,ID_8_MX_CLCK    (DATA_START)   $ OFFSET: MX_CLCK
        ,ID_8_EVENT_ID  (DATA_START+2)  $ OFFSET: EVENT ID (VOTRAX MESSAGE NUMBER)
        ;
```

Exhibit B6. IPC0008.LT: Votrax Events

```
$
$       SYS1 SHARED PAGE DATA - MSG ID 10
$
        LITERAL
         ID_10_SP1_DATA          (10)     $ MSG_ID
        ;
$
$       SYS2 SHARED PAGE DATA - MSG ID 11
$
        LITERAL
         ID_11_SP2_DATA          (11)     $ MSG_ID
        ;
$
$       SYS3 SHARED PAGE DATA - MSG ID 12
$
        LITERAL
         ID_12_SP3_DATA          (12)     $ MSG_ID
        ,ID_12_SPG_BEGIN                  (DATA_START)          $ INITIAL DATA OFFSET
        ,ID_12_ACE_CLCK                   (ID_12_SPG_BEGIN)     $ OFFSETS OF SHARED PAGE VARIABLES IN SYID_12
        ,ID_12_MX_CLCK                    (ID_12_SPG_BEGIN+2)
        ,ID_12_DEBUG_SWITCH               (ID_12_SPG_BEGIN+4)
        ,ID_12_$RPHONE                    (ID_12_SPG_BEGIN+5)
        ,ID_12_$LPHONE                    (ID_12_SPG_BEGIN+6)
        ,ID_12_$TALKSW                    (ID_12_SPG_BEGIN+7)
        ,ID_12_$FOOTSW                    (ID_12_SPG_BEGIN+8)
        ,ID_12_CAPS_STATE                 (ID_12_SPG_BEGIN+9)
        ,ID_12_RNG_FRM_CAP_TO_BOGEY       (ID_12_SPG_BEGIN+10)
        ,ID_12_RNG_FRM_CAP_TO_STRNGR      (ID_12_SPG_BEGIN+11)
        ,ID_12_RNG_FRM_CAP_TO_STATN       (ID_12_SPG_BEGIN+12)
        ,ID_12_DIS_HEAD                   (ID_12_SPG_BEGIN+13)
        ,ID_12_DOG_ALT                    (ID_12_SPG_BEGIN+14)
        ,ID_12_STRNGRS_ALT                (ID_12_SPG_BEGIN+15)
        ,ID_12_BEAR_FRM_CAP_TO_BOGEY      (ID_12_SPG_BEGIN+16)
        ,ID_12_BEAR_FRM_CAP_TO_STRNGR     (ID_12_SPG_BEGIN+17)
        ,ID_12_BEAR_FRM_CAP_TO_STATN      (ID_12_SPG_BEGIN+18)
        ,ID_12_BOGEYS_TRACK               (ID_12_SPG_BEGIN+19)
        ,ID_12_STRANGERS_TRACK            (ID_12_SPG_BEGIN+20)
        ,ID_12_BOGEYS_SPEED               (ID_12_SPG_BEGIN+21)
        ,ID_12_CUR_MOD                    (ID_12_SPG_BEGIN+22)
        ,ID_12_SEG_MOD                    (ID_12_SPG_BEGIN+23)
        ,ID_12_SCEN_RUNNING               (ID_12_SPG_BEGIN+24)
        ,ID_12_SEG_NO                     (ID_12_SPG_BEGIN+25)
        ,ID_12_$COMM2                     (ID_12_SPG_BEGIN+26)
        ,ID_12_STUD_NA                    (ID_12_SPG_BEGIN+27)
        ,ID_12_SIZE_SH_PAGE               (39)
        ;
```

Exhibit B7. IPC0010.LT: Shared Page Data

```
$
$       NTDS INPUTS - MSG ID 15
$
        LITERAL
         ID_15_NTDS_INPUT        (15)     $ MSG_ID VALUE
$       ,USES NI_TIME OFFSET AS DEFINED PREVIOUSLY
        ,ID_15_INPUT_NUMBER      (DATA_START+2)  $ HIGH BYTE OF EVNT_NUM
        ,ID_15_INPUT_DATA        (DATA_START+2)  $ LOW BYTE OF EVNT_NUM
        ,ID_15_TEC_DATA          (DATA_START+3)  $ START OF 40-WORD DATA ARRAY [1:40]
        ;
$
```

Exhibit B8. IPC0015.LT: NTDS Simulation Commands

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$      KMON-RECEIVED MESSAGES - MSG ID 100
$
$
$      Revision:
$             G. Slemon        9-12-80        Initialize ipc format change
$                              9-15-80        Enable inputs literal change
$             H. Lee           9-19-80        System Summary literal defined
$
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$$
$$            KMON will understand the following commands from user
$$     processes for the instructor and student consoles:
$$
$$          COMMAND AT     DATA AT OFFSET FROM DATA_START:
$$          DATA_START     1      2           3          4          5
$$          ------------- ---- ----------- ---------- ---------- ----------
$$
$$          INITIALIZE     type  destination  --         byte_count  text_string
$$
$$          DISPLAY_PAGE   type  destination  --         page_number --
$$
$$          DISPLAY_LINE   --    destination  --         byte_count  text_string
$$
$$          ERASE_SCREEN   --    destination  --          --          --
$$
$$          RELEASE        type  destination  --          --          --
$$
$$          GO_BACK        --    --           --          --          --
$$
$$          ENABLE_INPUT   reply destination  input_word lo_numeric  hi_numeric
$$                         _id
$$
$$          DISABLE_INPUT  --    destination  --          --          --
$$
$$          ENABLE_SFK     --    --          sfk_word     --          --
$$
$$          DISABLE_SFK    --    --          sfk_word     --          --
$$
$$
$$            KMON will understand the following commands from user
$$     processes for the TEC CRT:
$$
$$          COMMAND AT     DATA AT OFFSET FROM DATA_START:
$$          DATA_START     1      2           3          4          5
$$          ------------- ---- ----------- ---------- ---------- ----------
$$
$$          INITIALIZE     type  destination  --         byte_count  text_string
$$
$$          DISPLAY_PAGE   type  destination form_feed  page_number --
```

Exhibit B9. IPC0100.LT: Keyboard and CRT Display Commands

```
$$
$$        DISPLAY_TEXT   --     --          form_feed  byte_count  text_string
$$
$$        ERASE_SCREEN   --   destination    --          --          --
$$
$$        RELEASE       type  destination    --          --          --
$$
       LITERAL
        ID_100_SYS1_EMON_REQUEST       (100)    $ MSG_ID FOR INSTRUCTOR
                                                $ OR STUDENT CONSOLES

       ,ID_100_SYS2_EMON_REQUEST       (101)    $ MSG_ID FOR TEC CRT

       ,ID_100_REQUEST_TYPE     (DATA_START)    $ TYPES FOLLOW:
       ,ID_100_INITIALIZE       (1)
       ,ID_100_ENABLE_SFK       (2)
       ,ID_100_DISABLE_SFK      (3)
       ,ID_100_ENABLE_INPUT     (4)
       ,ID_100_DISABLE_INPUT    (5)
       ,ID_100_DISPLAY_PAGE     (6)
       ,ID_100_DISPLAY_TEXT     (7)
       ,ID_100_DISPLAY_LINE     (8)
       ,ID_100_RELEASE          (9)
       ,ID_100_GO_BACK          (10)
       ,ID_100_ERASE_SCREEN     (11)
       ,ID_100_TYPE             (DATA_START+1)  $ OFFSET FOR TYPE (FILE/CRT ID)
       ,ID_100_INST_SUMMARY     (1)             $       INSTRUCTOR SUMMARY.
       ,ID_100_ST_SUMMARY       (2)             $       STUDENT SUMMARY.
       ,ID_100_ST_SPEECH        (3)             $       STUDENT SPEECH.
       ,ID_100_TEC_SPEECH       (4)             $       TEC CRT SPEECH.
       ,ID_100_ST_ATC           (5)             $       STUDENT TEACHING.
       ,ID_100_TEC_ATC          (6)             $       TEC TEACHING.
       ,ID_100_INST_MENUS       (7)             $       INSTRUCTOR MENUS.
       ,ID_100_ST_MENUS         (8)             $       STUDENT MENUS.
       ,ID_100_TEC_RUN_ERRORS   (9)             $       TEC RUN ERRORS.
       ,ID_100_REPLY_ID         (DATA_START+1)  $ OFFSET FOR VALID INPUT MES-
                                                $ SAGE SENDING:
$$     ,ID_200_MES_KEY          (244)           $       ATC'S INPUTS
$$     ,ID_400_INST_INPUT       (402)           $       INSTRCTR SUMMARY INPUTS
$$     ,ID_400_STDT_INPUT       (412)           $       STUDENT SUMMARY INPUTS
       ,ID_100_CONSOLE          (DATA_START+2)  $ OFFSET FOR TARGET CONSOLE:
       ,ID_100_ST_KBRD          (1)             $       STUDENT KEYBOARD
       ,ID_100_INST_KBRD        (2)             $       INSTRUCTOR KEYBOARD
       ,ID_100_TEC_CRT          (3)             $       TEC CRT

       ,ID_100_INPUT_WORD       (DATA_START+3)  $ OFFSET FOR INPUT_WORD
       ,ID_100_INPUT_SHIFT_MASK(001000R8P1)     $       MASK FOR SHIFT STYLE
                                                $       INPUTS_WORD CREATION.
       ,ID_100_MK_ALL_INPUTS    (177777R8P1)    $       MASK FOR ALL INPUTS
       ,ID_100_MK_STRING        (020000R8P1)    $       MASK FOR STRINGS
       ,ID_100_MK_NO            (010000R8P1)    $       MASK FOR "NO"
       ,ID_100_MK_YES           (004000R8P1)    $       MASK FOR "YES"
       ,ID_100_MK_NEXT          (002000R8P1)    $       MASK FOR "NEXT"
       ,ID_100_MK_NUMERIC       (001000R8P1)    $       MASK FOR NUMERIC INPUTS
       ,ID_100_ID_STRING        (-4)            $       ID # FOR STRING
```

Exhibit B9. IPC0100.LT: Keyboard and CRT Display Commands (Contin.)

```
,ID_100_ID_NO             (-3)              $       ID # FOR "NO"
,ID_100_ID_YES            (-2)              $       ID # FOR "YES"
,ID_100_ID_NEXT           (-1)              $       ID # FOR "NEXT"
,ID_100_ID_NUMERIC        (0)               $       ID # FOR NUMERICS

,ID_100_SFK_WORD          (DATA_START+3)    $ OFFSET FOR SKF_WORD
,ID_100_SFK_SHIFT_MASK    (100000R8P1)      $       MASK FOR SHIFT STYLE
                                            $       SFK_WORD CREATION.
,ID_100_ALL_SFKS          (177777R8P1)      $       MASK FOR ALL SFKS
,ID_100_MK_SHIFT_STOP     (100000R8P1)      $       MASK FOR "^STOP"
,ID_100_MK_ABORT          (040000R8P1)      $       MASK FOR "ABORT"
,ID_100_MK_HELP           (020000R8P1)      $       MASK FOR "HELP"
,ID_100_MK_REPLAY         (010000R8P1)      $       MASK FOR "REPLAY"
,ID_100_MK_INIT_VT        (004000R8P1)      $       MASK FOR "INIT V/T"
,ID_100_MK_STOP_VT        (002000R8P1)      $       MASK FOR "STOP V/T"
,ID_100_MK_OVERRIDE       (001000R8P1)      $       MASK FOR "OVERRIDE"
,ID_100_MK_BREAK          (000400R8P1)      $       MASK FOR "BREAK"
,ID_100_MK_CONTINUE       (000200R8P1)      $       MASK FOR "CONTINUE"
,ID_100_MK_RETRAIN        (000100R8P1)      $       MASK FOR "RETRAIN"
,ID_100_MK_STATS          (000040R8P1)      $       MASK FOR "STATS"
,ID_100_MK_DISABLE_KBRD   (000020R8P1)      $       MASK FOR "DISABLE KBRD"
,ID_100_MK_MENU           (000010R8P1)      $       MASK FOR "MENU"
,ID_100_MK_BYE            (000004R8P1)      $       MASK FOR "BYE"
,ID_100_MK_ENABLE_KBRD    (000002R8P1)      $       MASK FOR "ENABLE KBRD"
,ID_100_MK_NEW_TE         (000001R8P1)      $       MASK FOR "NEW T/E"
,ID_100_ID_SHIFT_STOP     (0)               $       ID # FOR "^STOP"
,ID_100_ID_ABORT          (1)               $       ID # FOR "ABORT"
,ID_100_ID_HELP           (2)               $       ID # FOR "HELP"
,ID_100_ID_REPLAY         (3)               $       ID # FOR "REPLAY"
,ID_100_ID_INIT_VT        (4)               $       ID # FOR "INIT V/T"
,ID_100_ID_STOP_VT        (5)               $       ID # FOR "STOP V/T"
,ID_100_ID_OVERRIDE       (6)               $       ID # FOR "OVERRIDE"
,ID_100_ID_BREAK          (7)               $       ID # FOR "BREAK"
,ID_100_ID_CONTINUE       (8)               $       ID # FOR "CONTINUE"
,ID_100_ID_RETRAIN        (9)               $       ID # FOR "RETRAIN"
,ID_100_ID_STATS          (10)              $       ID # FOR "STATS"
,ID_100_ID_DISABLE_KBRD   (11)              $       ID # FOR "DISABLE KBRD"
,ID_100_ID_MENU           (12)              $       ID # FOR "MENU"
,ID_100_ID_BYE            (13)              $       ID # FOR "BYE"
,ID_100_ID_ENABLE_KBRD    (14)              $       ID # FOR "ENABLE KBRD"
,ID_100_ID_NEW_TE         (15)              $       ID # FOR "NEW T/E"

,ID_100_FORM_FEED         (DATA_START+3)    $ OFFSET FOR FORM FEED FLAG
,ID_100_NEW_PAGE          (TRUE)            $       BEGIN A NEW PAGE OF TEXT
,ID_100_SCROLL            (FALSE)           $       APPEND TO EXISTING TEXT

,ID_100_PAGE_NUMBER       (DATA_START+4)    $ OFFSET FOR PAGE NUMBER OF TEXT
,ID_100_BYTE_COUNT        (DATA_START+4)    $ OFFSET LENGTH OF TEXT IN BYTES
,ID_100_LO_NUMERIC        (DATA_START+4)    $ OFFSET FOR LOWER BOUND FOR
                                            $ NUMERIC INPUTS.
,ID_100_HI_NUMERIC        (DATA_START+5)    $ OFFSET FOR UPPER BOUND FOR
                                            $ NUMERIC INPUTS.
,ID_100_PAGE_FILE         (DATA_START+5)    $ OFFSET FOR NAME OF TEXT FILE
,ID_100_TEXT_STRING       (DATA_START+5)    $ OFFSET OF START OF TEXT

;
```

Exhibit B9. IPC0100.LT: Keyboard and CRT Display Commands (Contin.)

```
$
$       EXEC RECEIVED MESSAGES - MSG ID 150, 160, 170
$
$ FILE TRANSFER REQUESTS
$ ----------------------

        LITERAL
         ID_150_FILE_X           (150)          $ MSG-ID, FILE XFER REQUEST
        ,ID_151_FILE_X_RESP      (151)          $ MSG_ID, FILE XFER RESPONSE
        ,ID_152_FILE_REQ         (152)          $ MSG_ID, EXEC - EXEC REQUEST
        ,ID_153_FILE_DATA        (153)          $ MSG_ID, FILE DATA MSG
        ,ID_154_FILE_REQ         (154)          $ MSG_ID, REGISTRATION - EXEC REQUEST
        ,ID_150_PKG     (DATA_START)            $ OFFSET: PACKAGE # OR PKG/FILE #
        ,ID_150_NAME    (DATA_START+1)          $ OFFSET: STUDENT NAME ENDING WITH <0>, OR -1 IF STUD_NA
        ,ID_150_TX_OPT  (DATA_START+1)          $ OFFSET: TRANSFER OPTION--
        ,ID_150_OPT_SEND         (1)            $ DATUM: ID_152 TX_OPT = SEND FILES
        ,ID_150_OPT_ASK          (2)            $   "       "     "    = REQUEST THESE FILES
        ,ID_150_OPEN             (1)            $   "    ID_153 TX_OPT = OPEN FILE
        ,ID_150_DATA             (2)            $   "       "     "    = DATA FOLLOWS
        ,ID_150_EOF              (3)            $   "       "     "    = EOF
        ,ID_150_EOP              (4)            $   "       "     "    = END PKT
        ,ID_150_ERROR   (DATA_START+1)          $ OFFSET: ERROR CODE OR 0 IF OK
        ,ID_150_ALT_DEST(DATA_START+2)          $ OFFSET: ALT_DEST FOR ID_151
        ,ID_150_BLK_NO  (DATA_START+3)          $ OFFSET: BLOCK NUMBER
        ,ID_150_PATH    (DATA_START+3)          $ OFFSET: PATHNAME
        ,ID_150_BYTES   (DATA_START+4)          $ OFFSET: NUMBER OF BYTES IN DATA
        ,ID_150_BLOCK   (DATA_START+5)          $ OFFSET: START OF FILE DATA BLK
        ;
$
$ SYNC ID'S
$ ---------

        LITERAL
         ID_160_R1_SYNC          (160)          $ MSG_ID, CPU1->2->3->1 SYNC
        ,ID_161_R2_SYNC          (161)          $ MSG_ID, CPU1->3->2->1 SYNC
        ;
$
$ ERRORS
$ ------

        LITERAL
         ID_170_MAJ_ERR          (170)          $ MSG_ID, MAJOR ERROR CALLED
        ,ID_171_MIN_ERR          (171)          $ MSG_ID, MINOR ERROR CALLED
        ,ID_170_ER_CD   (DATA_START)            $ OFFSET: ERMES ERROR CODE
        ,ID_170_DATUM   (DATA_START+1)          $ OFFSET: DATUM SENT TO MAJ/MIN ER
        ,ID_170_ST_LEN  (DATA_START+2)          $ OFFSET: LENGTH OF STRING
        ,ID_170_STR     (DATA_START+3)          $ OFFSET: STRING
        ;

$
```

Exhibit B10. IPC0150.LT: File Transfer Commands

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$NAME:  IPC0200.LT
$DESCRIPTION:
$               THIS LITERAL FILE CONTAINS THE IPC LITERALS FOR THE
$               IPCS SENT AND RECEIVED ONLY BY THE ATC PROCESS.  ALSO,
$               THIS LITERAL FILE CONTAINS THE MESSAGE LITERALS
$               FOR ATC VOIT MESSAGES WITH IPC TYPE FORMATS.
$
$       NOTE:   IPC_FORMATS.LT MUST BE USED IN CONJUNCTION WITH
$               THIS LITERAL FILE SO THAT ALL THE MESSAGES ARE
$               FULLY DEFINED.
$
$
$CALLING SEQUENCE:
$               NONE
$
$MODULE DEVELOPED BY G. SLEMON FOR ACE
$REVISION DATE: 11-5-80
$               2-11-81 ADDED ANOTHER ITM FOR ATC FOR CONTINUE ACTIVITY
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

$**********************************************************************
$       THE FOLLOWING MESSAGES ARE IPC MESSAGES:
$**********************************************************************

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 201: ATC MENU REQUESTS FROM ATC AND MANUAL SELECT
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_MNU (201);           $ MSG_ID

LITERAL ID_200_MENU_ID (DATA_START),            $ MENU TYPE NUMBER OFFSET
$       ITS VALUES ARE:
$       USES ABRT_MSG, OVRD_MSG, AND RTRN_MSG DEFINED IN MSG 200
        ID_200_REVW_MSG (10),                   $ REVIEW MENU REQUEST
        ID_200_STAT_MSG (11),                   $ STUDENT STATS REQUEST
        ID_200_REPT_MSG (12),                   $ REPEAT MENU REQUEST

$THE SECOND DATA WORD OFFSET IS SRC_EBD AS DESCRIBED FOR MESSAGE 200

        ID_200_EMON_CHN (DATA_START+2); $ EMON ASSIGNED FILE CHANNEL OFFSET

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 202: ATC ACTIVITY INITIATION REQUEST FROM MANUAL SELECT
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_ACT (202);           $ MSG_ID

$ ID_200_ACT_ID IS DEFINED IN IPC_FORMATS (MSG 260) AND IS THE OFFSET FOR THE FOLLOWING:
$       ITS VALUES ARE:
LITERAL ID_200_SIOF_MSG (14),                   $ SIGN OFF REQUESTED
        ID_200_NRMS_MSG (15),                   $ NORMAL SEGMENT REQUESTED
        ID_200_RVWS_MSG (16),                   $ REVIEW SEGMENT REQUESTED
        ID_200_RPTS_MSG (17),                   $ REPEAT SEGMENT REQUESTED
        ID_200_ADVS_MSG (18),                   $ ADVANCE SEGMENT REQUESTED
```

Exhibit B11. IPC0200.LT (IPCS ONLY): Adaptive Training Commands

```
        ID_200_RTNS_MSG (19),                 $ RETRAIN ELEMENT REQUESTED
        ID_200_VALD_MSG (20),                 $ VALIDATE RETRAINED ELEMENT

        ID_200_ACT_MD (DATA_START+1);   $ SELECTION OFFSET
$       ITS VALUES ARE:                   0, FOR SIOF,RRMS,ADVS,VALD
$                                         SELECTION NUMBER, FOR REST

$ THE THIRD DATA WORD OFFSET IS KNOW_CMN AS DESCRIBED FOR MESSAGE 20!

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 205: RUN/SEGMENT/GIZMO COMPLETED FROM ACTIVITY TASK
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_ADN (205);          $ MSG_ID

$THE FIRST DATUM IS ACT_ID AS FOR MESSAGE 202 EXCEPT THAT THE MESSAGE
$INDICATES ACTIVITY COMPLETION NOT A REQUEST FOR INITIATION (AT DATA_START).

LITERAL ID_200_RNEN_MSG (27),                 $ RUN COMPLETED
        ID_200_SGEN_MSG (28),                 $ SEGMENT COMPLETED
        ID_200_IREF_MSG (29),                 $ SEGMENT EXITED ON INSTRUCTOR REFERRAL
        ID_200_GZEN_MSG (30),                 $ GIZMO COMPLETED

$ ID_200_ACT_RESULT (DATA_START+1) IS DEFINED IN IPC_FORMATS (MSG 260) AND IS THE
$ OFFSET FOR THE FOLLOWING:

$FOR ACT_ID = RNEN_MSG

        ID_200_NEXT_RUN (1),                  $ DO ANOTHER RUN
        ID_200_AFTR_RUN (2),                  $ CONTINUE SEGMENT AFTER RUN

$FOR ACT_ID = SGEN_MSG

        ID_200_PASS_IT (11),                  $ SEGMENT PASSED
        ID_200_PMVF_IT (12),                  $ FREEZE PMV REMEDIATION REQUIRED
        ID_200_PMVA_IT (13),                  $ ACTIVE PMV REMEDIATION REQUIRED
        ID_200_CHLF_IT (14),                  $ CHALLENGE SEGMENT FAILED
        ID_200_CHLD_IT  (15),                 $ SEGMENT CHALLENGED

$FOR ACT_ID = GZEN

        ID_200_RTRN_DN (21),                  $ RETRAIN COMPLETED
        ID_200_VALD_DN (22),                  $ VALIDATION COMPLETED
        ID_200_VCTS_DN (23),                  $ VOICE TEST COMPLETED
        ID_200_RPLY_DN (24),                  $ REPLAY COMPLETED

        ID_200_ACT_CPP (DATA_START+2);  $ COMMENTED PRACTICE PASSED

$ALWAYS FALSE EXCEPT WHEN ACT_ID = SGEN_MSG AND ACT_RESULT = PMVA_IT
$AND COMMENTED PRACTICE SEGMENT HAS BEEN PASSED.

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 206: MAJOR ERROR REQUEST FROM OTHER ATC TASKS
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Exhibit B11. IPC0200.LT (IPCS ONLY): Adaptive Training Commands (Contin.)

```
LITERAL ID_200_MES_MJR (208);           $ MSG_ID

$THE FIRST DATUM OFFSET IS REFERRED TO VIA MSG_CODE

LITERAL ID_200_MAJR_MSG (32);                      $ MAJOR ERROR REQUEST VALUE

LITERAL ID_200_MJR_ERRC (DATA_START+1);            $ MAJOR ERROR CODE OFFSET

LITERAL ID_200_MJR_DATA (DATA_START+2); $ MAJOR ERROR DATUM OFFSET

LITERAL ID_200_MJR_WHOM (DATA_START+3); $ MAJOR ERROR REPORTER NAME OFFSET

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 209: ACTIVITY TASK INITIALIZATION COMPLETE FROM ACTIVITY TASK
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_INI (209),           $ MSG_ID
        ID_200_LAST_CMSG (209); $ THIS IS THE LAST CONTROL MESSAGE
                                $    IF THIS CHANGES THEN MAX_CMSG MUST, TOO.

$ONLY ONE DATUM IS PRESENT AND IS REFERRED TO VIA MSG_CODE

LITERAL ID_200_IRDY_MSG (33),                      $ ACTIVITY TASK INITIALIZATION COMPLETE
        ID_200_MAX_MSG_CODE (33);                  $ THIS IS THE LAST CONTROL MSG_CODE

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 242: TIMER TIMEOUT MESSAGE FROM TIMER TO ACTIVITY
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_TMO (242);           $ MSG_ID

$NO DATUM IS ASSOCIATED WITH THIS MESSAGE

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 243: VIDEO DISC DONE MESSAGE FROM VIDEO TO ACTIVITY
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_VID (243);           $ MSG_ID

$NO DATUM IS ASSOCIATED WITH THIS MESSAGE
```

Exhibit B11. IPC0200.LT (IPCS ONLY): Adaptive Training Commands (Contin.)

```
$ REVISION MADE ON SEPT 4 1980 BY G. SLEMON
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$        MESSAGE 260: SPEECH ACTIVITY COMPLETED FROM SPEECH
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_200_MES_SPD (260);            $ MSG_ID

LITERAL ID_200_ACT_ID (DATA_START);      $ OFFSET FOR ACTIVITY DONE ID

                                         $ VALUES:
LITERAL ID_200_SPP_DN (1),                       $ SPEECH PRACTICE COMPLETED
        ID_200_SPC_DN (2),                       $ SPEECH COLLECTION COMPLETED
        ID_200_SPV_DN (3),                       $ SPEECH VALIDATION COMPLETED
        ID_200_CLN_DN (4),                       $ IAT SEGMENT END OR RETRAIN END
                                                 $ CLEANUP COMPLETED
        ID_200_IVT_DN (5),                       $ INIT VOICE TEST SETUP COMPLETED
        ID_200_SVT_DN (6),                       $ STOP VOICE TEST COMPLETED
        ID_200_RTN_DN (7);                       $ RETRAIN COMPLETED

LITERAL ID_200_ACT_RESULT (DATA_START+1),        $ OFFSET FOR DONE STATUS

                                                 $ VALUES:
        ID_200_OKAY_IT (1),                      $ SPEECH ACTIVITY COMPLETED PROPERLY
        ID_200_INRF_IT (2);                      $ NOTIFY THE INSTRUCTOR
```

Exhibit B12. IPC0260.LT: Speech Activity Completed Adaptive Training Command

```
$                      *** IPC0400.LT ***
$ REV: 5                                                       DATE: 11/17/80
$*****************************************************************************
$
$  NAME:              IPC0400.LT
$
$  DESCRIPTION:
$       THIS IS THE LITERAL FILE DESCRIBING THE IPC MESSAGE LITERALS
$       FOR THE SUMMARY PROCESSES -> 'SYSSUM' AND 'STUSUM'
$
$  CALLING SEQUENCE:
$
$       INCLUDE IPC0400.LT;
$
$  REVISION HISTORY:
$
$              NANCEY LEE      NOV 80          ACE
$
$*****************************************************************************

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 401: INSTRUCTOR STATS REQUESTED (FROM KMON TO SYSSUM)
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_400_SYSSUM_REQ(401),          $ MSG_ID
$  THE VALUE AT DATA_START IS:
       ID_400_STATS_HIT(1),              $ INSTRUCTOR STATS HAS BEEN REQUESTED
       ID_400_NEWSTE_HIT (2),            $ NEW T/R KEY HIT, REGISTRATION REQSTED
       ID_400_SHIFT_STOP (3);            $ SHIFT STOP KEY HAS HIT

$      SINCE THIS IS A ONE WORD MESSAGE, I HAVE NOT DEFINED A LITERAL
$      TO REFER TO THE START OF THE DATA, USE 'DATA_START'

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 402: REQUESTED KEY INPUT FROM KMON TO SYSSUM
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_400_INST_INPUT(402),          $ MSG_ID
       ID_402_INPUT_TYPE(DATA_START),  $ OFFSET FOR THE TYPE OF INPUT
$  POSSIBLE VALUES ARE THE SAME AS IN IPC0100.LT
$         ID_100_ID_STRING(-4)
$         ID_100_ID_NEXT(-1)
$         ANY POSITIVE VALUE IN THIS OFFSET WILL BE TAKEN AS A NUMERIC
$         INPUT.  ITS RANGE IS 0 - 32767.
       ID_402_INPUT_BYTECOUNT(DATA_START+2),    $ OFFSET FOR BYTE COUNT IF
                                                $ TYPE IS A STRING
       ID_402_TEXT_STRING(DATA_START+3);        $ TEXT OF STRING FOLLOWS

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 411: STUDENT STATS REQUESTED (FROM KMON TO STUSUM)
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_400_STUSUM_REQ(411),          $ MSG_ID
```

Exhibit B13. IPC0400.LT: Summary Function Commands

```
$  THE VALUE AT DATA_START IS:
        ID_400_CHOICE3_HIT(1);              $ STUDENT STATS HAS BEEN REQUESTED

$       SINCE THIS IS A ONE WORD MESSAGE, I HAVE NOT DEFINED A LITERAL
$       TO REFER TO THE START OF THE DATA, USE 'DATA_START'

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$       MESSAGE 412: REQUESTED KEY INPUT FROM EMON TO STUSUM
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

LITERAL ID_400_STDT_INPUT(412),             $ MSG_ID
        ID_412_INPUT_TYPE(DATA_START),  $ OFFSET FOR THE TYPE OF INPUT
$  POSSIBLE VALUES ARE THE SAME AS IN IPC0100.LT
$          ID_100_ID_STRING(-4)
$          ID_100_ID_NEXT(-1)
$          ANY POSITIVE VALUE IN THIS OFFSET WILL BE TAKEN AS A NUMERIC
$          INPUT.  ITS RANGE IS 0 - 32767.
$       ID_412_INPUT_BYTECOUNT(DATA_START+2),   $ OFFSET FOR BYTE COUNT IF
                                                $ TYPE IS A STRING
        ID_412_TEXT_STRING(DATA_START+3);       $ TEXT OF STRING FOLLOWS
```

Exhibit B13. IPC0400.LT: Summary Function Commands (Contin.)

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ SPEECH COMMANDS - MSG ID 2100
$
$        Revision history:
$
$                D. HARRY        9-30-80: Added definition of IPC #,
$                                (ID_2100_SPEECH_REQUEST)
$                J. BOLLENBACHER 12-15-80: Added ID_2100_START_REPLAY
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
        LITERAL
         ID_2'00_SPEECH_REQUEST          (2100)  $ IPC # FOR SPEECH REQUESTS
        ,ID_2100_SPEECH_COMMAND (DATA_START)     $ OFFSET FOR COMMAND NUMBER.
        ,ID_2100_SP_P                    (1)     $ INITIATE SPEECH PRACTICE.
        ,ID_2100_SP_C                    (2)     $ INITIATE SPEECH COLLECTION.
        ,ID_2100_SP_V                    (3)     $ INITIATE SPEECH VALIDATION.
        ,ID_2100_RETRAIN                 (4)     $ INITIATE RETRAIN.
        ,ID_2100_PREPARE_TO_RECOGNIZE    (5)     $ INITIATE THE PRE-SCENARIO
                                                 $ PREPARATION FOR SPEECH.
        ,ID_2100_VOICETEST               (6)     $ INITIATE VOICETEST.
        ,ID_2100_END_RECOGNITION         (7)     $ STOP THE RUNTIME RECOGNITION.
        ,ID_2100_STOP_VOICETEST          (8)     $ STOP VOICETEST.
        ,ID_2100_ALLOW_SCENARIO_START    (9)     $ TELLS SPEECH THAT THE NEXT
                                                 $ FOOTKEY TOGGLE SIGNALS THE
                                                 $ BEGINNING OF A SCENARIO.
        ,ID_2100_START_REPLAY            (10)    $ START SPEECH REPLAY
        ,ID_2100_ELEMENT_NO     (DATA_START+1)   $ OFFSET FOR ELEMENT NUMBER TO
                                                 $ BE USED BY SP/* OR RETRAIN.
        ;
$
```

Exhibit B14. IPC2100.LT: Speech Commands

```
$
$       SWC VOXBOX REQUESTS ISSUED BY THE SWC MODEL - MSG ID 2300
$
        LITERAL
          ID_2300_SWC_VOX        (2300)  $ MSG_ID
$
$       PSUEDO BOGEY & MAC VOXBOX REQUESTS (2ND PILOT'S VOICE) - MSG ID 2301
$
        ,ID_2301_MAC_VOX         (2301)  $ MSG_ID
$
$       DEMO MESSAGE REQUESTS FOR VOXBOX - MSG ID 2303
$
        ,ID_2303_DEMO_VOX        (2303)  $ MSG_ID
$
$       IAT MESSAGE REQUESTS FOR VOXBOX - MSG ID 2304
$
        ,ID_2304_IAT_VOX         (2304)  $ MSG_ID
$
$       AIC MESSAGE VOCALIZATION REQUESTS (SPEECH TRAINING) - MSG ID 2306
$
        ,ID_2306_AIC_VOX         (2306)  $ MSG_ID
        ;
```

Exhibit B15. IPC2300.LT: Voice Generation Commands

```
$FILE NAME SYS.LT
$
$LITERALS DEFINING CONSTANTS FOR EVENT NUMBERS AND DATA IN
$ SYSTEM EVENTS MESSAGES
$
$REVISION 0      JAN 21 80       ORIGINAL
$         1      DEC 13 80       MH - ADDED 106
$
LITERAL EXERCISE_START                            (101); $ ATC
LITERAL EXERCISE_END                              (102); $ SC
LITERAL EXERCISE_FREEZE                           (103); $ SC
LITERAL EXERCISE_ABORT                            (104); $ SC
LITERAL EXERCISE_INIT                             (105);  $ ATC
LITERAL EXERCISE_CONTINUE                         (106);  $ ATC

LITERAL STRANGER_VIDEO_UPDATED                    (108); $
LITERAL SPLIT_VIDEO_UPDATED                       (109); $ HUMAN TRACKER
LITERAL CAP_APPEARED                              (110); $ AIRCRAFT
LITERAL CAP_SYMBOL_BUILT                          (111); $ RADAR
LITERAL CAP_OFF_SCREEN                            (112); $ PROX CHECKER
LITERAL CAP_SYMBOL_REPOSITIONED                   (113); $ NTDS
LITERAL CAP_VIDEO_UPDATED                         (114); $ HUMAN TRACKER
LITERAL BOGEY_APPEARED                            (115); $ AIRCRAFT
LITERAL BOGEY_SYMBOL_REPOSITIONED                 (116); $ NTDS
LITERAL BOGEY_VIDEO_UPDATED                       (117); $ HUMAN TRACKER
LITERAL BOGEY_STARTED_SPLIT                       (118); $ BOGEY
LITERAL BOGEY_JINK_LEFT                           (119); $ BOGEY
LITERAL BOGEY_JINK_RIGHT                          (120); $ BOGEY
LITERAL CAP_ENGAGED_TO_BOGEY                      (121); $ NTDS
LITERAL CAP_ENGAGED_TO_SPLIT                      (122); $ NTDS
LITERAL CAP_ENGAGED_TO_STATION                    (123); $ NTDS
LITERAL CAP_ENGAGED_TO_POINT                      (124); $ NTDS
LITERAL PSEUDO_BOGEY_ENGAGED_TO_POINT             (125); $ NTDS
LITERAL CAP_DISENGAGED_FROM_BOGEY                 (126); $ NTDS
LITERAL CAP_DISENGAGED_FROM_SPLIT                 (127); $ NTDS
LITERAL CAP_DISENGAGED_FROM_STATION               (128); $ NTDS
LITERAL CAP_DISENGAGED_FROM_POINT                 (129); $ NTDS
LITERAL PSEUDO_BOGEY_DISENGAGED_FROM_POINT        (130); $ NTDS
LITERAL CAP_FADED                                 (131); $ SC
LITERAL CAP_UNFADED                               (132); $ ROFM
LITERAL BOGEY_FADED                               (133); $ SC
LITERAL BOGEY_UNFADED                             (134); $ ROFM

LITERAL BOGEY_SYMBOL_PUT_ON_SEQ_LIST              (137); $ NTDS

LITERAL HEADING_STATE_UPDATE                      (140); $ NTDS
LITERAL SPEED_STATE_UPDATE                        (141); $ NTDS
LITERAL ALTITUDE_STATE_UPDATE                     (142); $ NTDS
LITERAL FUEL_STATE_UPDATE                         (143); $ NTDS
LITERAL MISSILE_STATE_UPDATE                      (144); $ NTDS
LITERAL AIRCRAFT_ARRIVED_ALERT                    (145); $ NTDS
LITERAL POOR_SITUATION_ALERT                      (146); $ NTDS
LITERAL BREAK_ENGAGE_ORDER_ALERT                  (147); $ SWC
LITERAL SWC_ENGAGE_ORDER_ALERT                    (148); $ SWC
LITERAL NTDS_DOWN                                 (149); $ SC
```

Exhibit B16. SYS.LT: Scenario Events Literals

```
LITERAL BEEPER_ON_GUARD                        (150); $ SC
LITERAL ENTER_OFFSET                           (151); $ NTDS
LITERAL DEPRESSED_SWC_INTERCOM                 (152); $ NTDS

LITERAL CAP_5_MILES_FROM_CAP_STATION           (160); $ PROX CHECKER
LITERAL CAP_8_MILES_FROM_CAP_STATION           (161); $ PROX CHECKER
LITERAL CAP_10_MILES_FROM_CAP_STATION          (162); $ PROX CHECKER
LITERAL CAP_5_MILES_FROM_TACAN_STATION         (163); $ PROX CHECKER
LITERAL CAP_10_MILES_FROM_TACAN_STATION        (164); $ PROX CHECKER
LITERAL CAP_3_MILES_FROM_BOGEY                 (168); $ PROX CHECKER
LITERAL CAP_5_MILES_FROM_BOGEY                 (169); $ PROX CHECKER
LITERAL CAP_8_MILES_FROM_BOGEY                 (170); $ PROX CHECKER
LITERAL CAP_10_MILES_FROM_BOGEY                (171); $ PROX CHECKER
LITERAL CAP_15_MILES_FROM_BOGEY                (172); $ PROX CHECKER
LITERAL CAP_20_MILES_FROM_BOGEY                (173); $ PROX CHECKER
LITERAL CAP_3_MILES_FROM_STRANGER              (174); $ PROX CHECKER
LITERAL CAP_5_MILES_FROM_STRANGER              (175); $ PROX CHECKER
LITERAL CAP_10_MILES_FROM_STRANGER             (176); $ PROX CHECKER
LITERAL CAP_15_MILES_FROM_SPLIT                (177); $ PROX CHECKER
LITERAL CAP_10_MILES_FROM_SPLIT                (178); $ PROX CHECKER
LITERAL BOGEY_10_MILES_FROM_OWNSHIP            (179); $ PROX CHECKER
LITERAL BOGEY_15_MILES_FROM_OWNSHIP            (180); $ PROX CHECKER
LITERAL BOGEY_20_MILES_FROM_OWNSHIP            (181); $ PROX CHECKER
LITERAL BOGEY_OPENING                          (182); $ PROX CHECKER
LITERAL STRANGER_5_MILES_FROM_OWNSHIP          (183); $ PROX CHECKER
LITERAL STRANGER_10_MILES_FROM_OWNSHIP         (184); $ PROX CHECKER
LITERAL STRANGER_IS_OPENING                    (185); $ PROX CHECKER
LITERAL SPLIT_20_MILES_FROM_OWNSHIP            (186); $ PROX_CHECKER
LITERAL SPLIT_OPENING                          (187); $ PROX_CHECKER
LITERAL CAP_2_MILES_FROM_REF_POINT1            (188); $ PROX CHECKER
LITERAL CAP_5_MILES_FROM_REF_POINT1            (189); $ PROX CHECKER
LITERAL CAP_10_MILES_FROM_REF_POINT1           (190); $ PROX CHECKER
LITERAL CAP_2_MILES_FROM_REF_POINT2            (191); $ PROX CHECKER
LITERAL CAP_5_MILES_FROM_REF_POINT2            (192); $ PROX CHECKER
LITERAL CAP_10_MILES_FROM_REF_POINT2           (193); $ PROX CHECKER
LITERAL CAP_2_MILES_FROM_REF_POINT3            (194); $ PROX CHECKER
LITERAL CAP_5_MILES_FROM_REF_POINT3            (195); $ PROX CHECKER
LITERAL CAP_10_MILES_FROM_REF_POINT3           (196); $ PROX CHECKER
LITERAL STRANGER_5_MILES_FROM_REF3             (197); $ PROX CHECKER
$
$ SYMBOL IDENTIFIERS
$
LITERAL OWNSHIP_SYMBOL                         (1);
LITERAL CAP_SYMBOL                             (2);
LITERAL BOGEY_SYMBOL                           (3);
LITERAL STRANGER_SYMBOL                        (4);
LITERAL OTHER_SYMBOL                           (5);
```

Exhibit B16. SYS.LT: Scenario Events Literals

APPENDIX C

SEGMENT DEFINITION SYNTAX

This appendix describes the commands that are available for segment definitions. Table C-1 lists the commands alphabetically, their modifiers, and instructional segments to which the commands may be applied. A definition of each command, in terms of its application, follows the tabulated data for the command.

Commands, arguments and multiple arguments are separated by spaces. Each command line is ended with a new line. All "names" in the syntax will be correlated to their equivalent numbers. Names are limited to a maximum of 31 alphanumeric characters.

TABLE C-1. SEGMENT COMMAND SUMMARY

Command	Modifiers	Segments
ADV	step number	IAT
	ADVANCE TO STEP SENTINEL. Advance to the segment command following the command containing the ID number (range 0 to 65535). See STEP command.	
AV/CLR	----	IAT, CP, FP
	AV CLEAR. Clear the audiovisual screen.	
	AV/F frame number	
	AV FREEZE. Show single visual frame.	
AV/N	sequence number, audio channel number	IAT, CP, FP
	AV SEQUENCE. Show audiovisual sequence at normal speed, complete with audio. Three audio channels exist: channel one, two, and three (both channels one and two selected for mixed output).	
BEEP	----	IAT
	STUDENT BELL. Ring bell at student CRT.	
BUZZ	number of seconds	IAT
	TEC BUZZER. Buzz TEC buzzer for the number of seconds (default 3 seconds).	
CHAL	step number	IAT
	CHALLENGE. Receive CHALLENGE input from the student keyboard. If a YES input is received, the student is allowed to skip the instruc-	

TABLE C-1. SEGMENT COMMAND SUMMARY (continued)

Command	Modifiers	Segments
	tional segments leading to a Free Practice (FP) segment. If NO is received, the student will continue through the instructional segments starting with the segment command following the step specified as an argument.	
CK	check number	IAT
	CHECK. Execute the check simple testing subfunction using the commands which follow as input.	
CK/A	button, status	IAT check
	CHECK TEC INPUTS-ANY ORDER. Check for the specified TEC inputs in any order. Argument 1 is a button name or number. Argument 2 must be supplied as amplifying data associated with argument 1; e.g., COMM1 ON. See Appendix B for possible argument 1 and argument 2 code numbers.	
CK/E	----	IAT check
	CHECK END. End a check specification. CK/E must be the last command in a check specification.	
CK/H	----	IAT check
	DEFINE CHECK. Define the check arguments in the argument list: Argument 1 is the time allotted for CK completion in seconds. Argument 2 is whether a second try is allowed (Y or N). Argument 3 is whether the instructor is notified after a second failure (Y or N). Argument 4 is the presentation step number. Argument 5 is the step number for no-error feedback. Argument 6 is the step number for omission error feedback. Argument 7 is the step number for sequence error feedback. Argument 8 is the step number for other error feedback. Argument 9 is the step number for second failure feedback. Argument 10 is a TEC button name or number or the NEXT key. Argument 11 specifies any amplifying data associated with argument 10. See Appendix B for possible argument 10 argument 11 code numbers.	

TABLE C-1. SEGMENT COMMAND SUMMARY (continued)

Command	Modifiers	Segments
	All step numbers must be in the range 0 to 65535. Step number 0 must be used if feedback is not to be provided.	
CK/O	button, status	IAT check
	CHECK TEC INPUTS-ORDERED. Check for the specified TEC inputs in the order listed. Argument 1 is a button name. Argument 2 must be supplied as amplifying data associated with argument 1; e.g., COMM1 ON. See Appendix B for possible argument 1 and argument 2 code numbers.	
COMM/ON	channel number, on/off	IAT
	Turn communications on. Activate the requested communications channel or turn it off. The desired channel is activated and the COMM light is lit if the second argument is on. The channel is deactivated if the second argument is off.	
CP/H	argument list	CP
	DEFINE CP. Define the commented practice segment. Argument 1 is the number of error freezes allowed before remediation or instructor aid is prescribed. Argument 2 is the number of freeze-free runs required for advancement. Argument 3 is the number of PMV failures allowed in a single run.	
CRT/B	string message	IAT, CP, FP
	STUDENT CRT MSG. Display the string message on the student CRT.	
CRT/D	page number	IAT, CP, FP
	TEC CRT PAGE. Display the indicated page of text contained in the segment associated file on the TEC CRT.	
CRT/D/CLR	----	IAT, CP, FP
	CLEAR TEC CRT. Blank the TEC graphics display.	
CRT/T	page number	IAT, CP, FP
	STUDENT CRT PAGE. Display the indicated page of text contained in the segment associated file on the student CRT.	

TABLE C-1. SEGMENT COMMAND SUMMARY (continued)

Command	Modifiers	Segments
CRT/T/CLR	----	IAT, CP, FP
	CLEAR STUDENT CRT. Blank the student CRT.	
DRO	cell name, content number	IAT
	SHOW TEC DROs. Display the designated readouts on the Data Readout Display. Acceptable readouts and their cell positions are defined in Appendix B. Individual digits must be associated with the desired DRO cell number.	
DRO/CLR	----	IAT
	CLEAR DROs. Blank all 32 cells.	
END	----	IAT, CP, FP
	END SEGMENT. End the segment command file. Segment command file processing is terminated when this command is encountered.	
FP/H	argument list	FP
	DEFINE FP. Define free practice requirements. Argument 1 is the number of PMV failures allowed in a single run. Argument 2 is the number of consecutive passing runs required for advancement.	
GS	phrase type, phrase number, additional data (up to 4)	IAT, CP, FP
	GENERATE SPEECH. Generate the speech identified by name or number. The phrase type will be associated with the speech generation device based on the intended speaker. Additional data refers to dynamic name or numbers such as range, bearing, C/S, etc. The data are to be presented in the order in which they appear in the phrase.	
IAT/H	----	IAT
	DEFINE IAT. Define IAT segment initiation. IAT initialization and validity checking activities are initiated. IAT/H must be the first command in an IAT segment.	
LAB_STATE	desired state	IAT
	SET LABEL STATE. Sets the state of the VAB labels to UP or LOW labels.	

TABLE C-1. SEGMENT COMMAND SUMMARY (continued)

Command	Modifiers	Segments
LED	LED name or ALL	IAT
	LIGHT TEC LEDs. Light the TEC LEDs identified by name. Multiple arguments are allowed. Assigned LED numbers are defined in Appendix B. Turn on all LEDs when the argument is ALL.	
LED/CLR	LED names or ALL	IAT
	CLEAR TEC LEDs. Turn off the TEC LEDs identified by name. Multiple arguments are allowed. Turn off all LEDs when argument is ALL.	
NED	number	IAT
	Display on the NED readout the designated number. Range of numbers is 00000 to 99999. Default is 00000.	
NEXT	message number	IAT, CP, FP
	NEXT. Wait for entry of NEXT special function key at the student keyboard before executing the next command. Display the message by the message number at the student CRT while the system waits for the NEXT entry. Default is message number one.	
NTDS/D	----	IAT
	Simulate NTDS DOWN. Cause the TEC to respond as if NTDS is down.	
PMV/A	PMV number, PMV score	CP, FP
	GRADE PMVs. Identify the performance measurement grading criteria for grading runs in the segment. Values are input directly by providing the category number with the passing score. Multiple paired entries may be supplied by separating them with spaces. PMV pairs not identified will not be graded. Maximum of five pairs allowed per line.	
PMV/F	PMV numbers	CP
	PMV ERROR FREEZE. Identify the performance errors which cause a run scenario freeze upon detection of the error. The performance errors are referred to by performance measurement category name or number. Maximum of five PMV numbers per line.	
POSIT/BT	ball tab state, bearing, range	IAT
	POSITION BALL TAB. Position the ball tab at the desired bearing and range. The ball tab may be left in the following states after the repositioning: disabled, disabled but displayed, or displayed	

TABLE C-1. SEGMENT COMMAND SUMMARY (continued)

Command	Modifiers	Segments
	and enabled. The bearing must be between 1 to 360 degrees. The range must be miles times ten.	
PRESS	button, status	IAT
	SIMULATE VAB PRESSING. Cause the TEC to react as if the desired VAB or FAB was hit.	
R	step number, step number	IAT
	REPEAT SEGMENT STEPS. Repeat the segment steps that are bracketed between the step sentinel numbers. The numbers must be in the range of 0 to 65535. Processing will resume with the next command when the last sentinel is encountered.	
REC/KBD	keycap name	IAT
	RECEIVE STUDENT KEYBOARD. Receive from the student keyboard before proceeding. Anticipated keycap mnemonic is designated.	
REC/TEC	button, status	IAT
	RECEIVE TEC. Receive from the TEC before proceeding to the next command. Argument 1 is a button name or number. Argument 2 must be supplied as amplifying data associated with argument 1; e.g., COMM1 ON. See Appendix B for possible argument 1 and argument 2 code numbers.	
RUN/CP	----	CP
	RUN CP. Start the scenario described in the SC command for a Commented Practice run. RUN/CP must be preceded by a SC command. Speech recognition is automatically enabled by the RUN/CP command. The run will prematurely end if an error is encountered.	
RUN/FP	----	FP
	RUN FP. Start the scenario described in the SC command for a Free Practice run. RUN/FP must be preceded by a SC command. Speech recognition is automatically enabled by the RUN/FP command. A REPLAY option is automatically presented at the completion of each run.	

TABLE C-1. SEGMENT COMMAND SUMMARY (continued)

Command	Modifiers	Segments
RUN/S	----	IAT
	RUN SCENARIO. Start the scenario describd in the SC command. The command immediately following RUN/S will be executed as soon as the scenario begins.	
SC	scenario number	IAT, CP, FP, AVTV
	INITIALIZE SCENARIO. Initialize scenario contained in scenario file identified by number. Normal TEC response wll be enabled at this time.	
SC/C	----	IAT
	CONTINUE SCENARIO. Continue scenario from freeze state.	
SC/CLR	----	IAT, CP, FP
	CLEAR TEC. Stop scenario and clear the TEC lights and grahics display.	
SC/END	----	IAT
	STOP SCENARIO. Stops scenario and does not alter TEC presentation.	
SC/W	----	IAT
	SCENARIO WAIT. Wait for indication that scenario is frozen.	
SP/C	element number	IAT
	COLLECT SPEECH. Collect the speech phrase identified by element number within the number of seconds specified. Wait until phrase has been collected prior to continuing.	
SP/P	element number	IAT
	PRACTICE SPEECH. Practice the speech phrase identified by name or number within the number of seconds specified. Wait until phrase has been spoken prior to continuing.	
SP/V	element number	IAT
	VALIDATE SPEECH. Validate the speech phrase identified by element or number within the number of seconds specified. Wait until phrase has been validated prior to continuing.	

TABLE C-1. SEGMENT COMMAND SUMMARY (continued)

Command	Modifiers	Segments
STEP	step number	IAT
	IDENTIFY STEP. Identify current segment step as a sentinel step identified by the number (0 to 65535). This command is used to identify a position in the command sequence for repeat and advance commands.	
T	test number	IAT
	EXECUTE TEST. Execute the test simple test subfunction.	
T/E	----	IAT test
	END TEST. End a test specification. T/E must be the last command in a test specification. T/H and T/Q must be preceeded by a T command and followed by a T/E command.	
T/H	argument list	IAT test
	DEFINE TEST REQUIREMENTS. Define test requirements according to the argument list: Argument 1 is the number of questions to be given on the test. Argument 2 is the percentage score required to pass the test. Argument 3 is the step number at which the test presentation is given. Argument 4 is the step number for perfect score feedback. Argument 5 is the step for passing score feedback. Argument 6 is the step for first failure feedback. Argument 7 is the step for second failure. Argument 8 is the test type (MATCH or OTHER). Argument 9 is the text page number for a matching test or a blank if the test type is OTHER. Argument 10 is whether a second try is allowed (Y or N). Argument 11 is whether the instructor is to be notified after a second failure (Y or N).	
T/Q	argument list	IAT test
	DEFINE TEST QUESTIONS. Define a test question according to the argument list: Argument 1 is the type of answer options presented (TEC, T/F, or number indicating number of choices). Argument 2 is the step number at which segment commands for this question presentation begins. Argument 3, argument 4 is a correct response pair. Argument 5 is the step number of the correct-entry feedback.	

TABLE C-1. SEGMENT COMMAND SUMMARY (continued)

Command	Modifiers	Segments
	Argument 6 is the step number of the wrong-entry feedback. A step number must be in the range 0 to 65535 and must be 0 if no feedback is to be provided.	
	When a TEC input is the correct respons.. the correct response pair (argument 3 argument 4) is the form: TEC button name and amplifying data per Appendix B.	
TEC	alert name	IAT
	SIMULATE NTDS ALERT. Simulate the indicated alert.	
TRKSYM/HK	track number	IAT
	HOOK TRACK SYMBOL. Bring the desired track into close control.	
TRKSYM/KL	track number	IAT
	KILL TRACK SYMBOL. Stop updating and maintaining the display of the desired track symbol.	
TRKSYM/PC	track number, bearing, range	IAT
	CORRECT TRACK SYMBOL POSITION. Move the desired track symbol to the requested bearing and range. The bearing must be between 1 and 360 degrees. The range is miles times ten.	
WAIT	number of seconds	IAT, AVTV
	WAIT. Wait the designated number of seconds, 1 to 999 seconds.	
W/F	switch, status	IAT, CP, FP
	WAIT FOR. Wait for the specified TEC switch condition to be true.	

APPENDIX D

PERFORMANCE MEASUREMENT VARIABLES

INTRODUCTION

This appendix identifies the variables that are used by ACE to measure the performance of a student. A Performance Measurement Variable (PMV) may be applicable to only one level of training, or the basic skills may apply to several levels of training.

CROSS-REFERENCES

Table D1 contains training levels cross-referenced to performance measurement variables. Table D2 contains performance measurement variables cross-referenced to training levels. These cross-reference tables are provided for rapid association between training levels and performance measurement variables.

TABLE D1. TRAINING LEVELS - PERFORMANCE MEASUREMENT VARIABLES

LEVEL	PMV
1	10, 11
2	1 - 11
3	1 - 4, 7 - 19, 22 - 30
4	1, 5 - 11, 16, 22 - 27, 30 - 35, 82, 87
5	1 - 4, 10 - 20, 36 - 39
6	1, 40 - 46
7	1 - 20, 22 - 39, 47 - 54, 82 - 87
8	1, 7 - 9, 22, 24 - 27, 55 - 59, 62 - 81

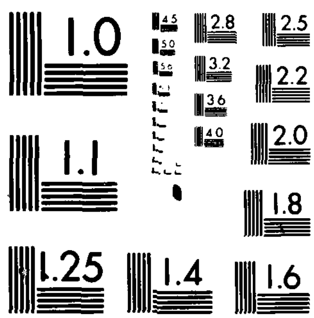
TABLE D2. PERFORMANCE MEASUREMENT VARIABLES - TRAINING LEVELS

PMV	LEVEL	PMV	LEVEL
1	2 - 8	24 - 27	3, 4, 7, 8
2, 3	2, 3, 7	28 - 29	3, 7
4	2, 3, 5, 7	30	3, 4, 7
5, 6	2, 4, 7	31 - 34	4, 7
7 - 9	2 - 4, 7, 8	35	4, 5, 7
10, 11	2 - 5, 7	36 - 39	5, 7
12 - 15	3, 5, 7	40 - 46	6
16	3, 4, 5	47 - 54	7
17 - 19	3, 5, 7	55 - 59	8
20	5, 7	60 - 61	deleted
21	deleted	62 - 81	8
22	3, 4, 7, 8	82 - 87	4, 7
23	3, 4, 7		

AD-A113 209 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
FUNCTIONAL DESIGN FOR AIR INTERCEPT CONTROLLER PROTOTYPE TRAINI--ETC(U)
DEC 81 E REGELSON, G SLEMON, R VERSTEEG N61339-78-C-0182
UNCLASSIFIED 8641-A009 NAVTRAEQUIPC-78-C-0182-8 NL

3 OF 3

END
DATE
FILMED
4-82
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963-A

PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS

Table D3 lists ACE Performance Measurement Variables. Each PMV is identified by number and name, and the performance and measurement are described. The Performance Measurement Variable Definitions contained in Table 3 are reprinted from Section VI of the Instructor Handbook.[1]

1. Prototype Equipment Instructor Handbook for ACE (Air Intercept Controller Prototype Training system), Report NAVTRAEQUIPCEN 78-C-0182-9 (Logicon, Inc.). Naval Training Equipment Center, Orlando, Florida; in press.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS

No.	Name	Performance/Scoring Description
1	MAINTAIN CAP SYMBOL IN VICINITY OF CAP VIDEO	100 points - Maintain the CAP symbol such that 3 out of each 5 times that the video is displayed, missing no more than 2 video displays in a row, the CAP symbol is within 1/8 inch of the video. This requirement takes effect after the first time that the symbol is updated (assigned to the CAP video). Each update omission outside stated tolerance deducts 10 points.
2	ENGAGE CAP TO STATION	a. 60 points - Engage the CAP symbol to the CAP station prior to being 5 miles from station. b. 25 points - Transmit message "c/s (PORT/STARBOARD) xxx". This transmission is optional if the aircraft heading is within +/- 20 degrees of the heading to the station and mandatory otherwise within 24 seconds after check-in. c. 15 points - If the heading is transmitted, it must be within +/- 10 degrees of the correct heading.
3	TRANSMIT STATION BEARING AND RANGE	a. 70 points - Transmit the message "STATION xxx,yy" to the CAP within 42 seconds after the scenario begins; 35 points for response within 54 seconds. b. 15 points - Transmitted bearing must be within +/- 15 degrees of correct bearing. c. 15 points - Transmitted range must be within +/- 2 miles of station.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
4	TRANSMIT BEARING AND RANGE OF STATION	a. 70 points - Transmit the message "STATION xxx,yy" within 12 seconds after the sweep passes the CAP position, 3 out of each 5 sweeps until the AIC transmits "ON STATION" or until the AIC makes a stranger report. The maximum score is given if the transmissions are made 3 out of 5 times and the AIC does not miss a transmission any 2 sweeps in a row. The score is decreased in proportion to the transmission omissions. Each out of tolerance transmission deducts 5 points. b. 15 points - Transmit accurate bearing. A maximum score is given if all transmitted bearings are within the specified tolerance. The score is decreased in proportion to the number of out of tolerance transmissions. Bearing tolerance is +/- 2 degrees different from NTDS displayed data. Each out of tolerance transmission deducts 3 points. c. 15 points - Transmit accurate range. A maximum score is given if all transmitted ranges are within tolerance. The score is decreased in proportion to the number of out of tolerance range transmissions. Range tolerance is +/- 2 miles different from NTDS displayed data. Each out of tolerance transmission deducts 3 points.
5	ENGAGE CAP TO BOGEY	100 points - After the Target Assigned alert is received, depress ORDER SEND to engage the CAP to the bogey, within 18 seconds. 70 points within 24 seconds.
6	VECTOR CAP TO BOGEY	a. 70 points - Transmit message "c/s PORT/VECTOR/STARBOARD xxx FOR [illegible]" to the CAP within 18 seconds of sequ[illegible] to the SWC Engage alert. 30 points fo[illegible]smissions within 24 seconds. b. 30 points - Transmit an accurate vector for bogey to the CAP. The heading must be within +/- 10 degrees of the correct heading.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
7	TRANSMIT INITIAL BOGEY BEARING AND RANGE	a. 40 points - Transmit the message "BOGEY xxx,yy" to the CAP within 18 seconds of vector for bogey message. 20 points for transmission within 24 seconds. b. 30 points - Transmitted bearing must be within +/- 2 degrees of NTDS displayed data. c. 30 points - Transmitted range must be within +/- 2 miles of NTDS displayed data.
8	TRANSMIT INITIAL BOGEY TRACK AND GROUND SPEED	a. 50 points - Transmit the message "BOGEY TRACKING xxx, SPEED POINT y" to the CAP within 18 seconds of initial bogey bearing and range call. 25 points for transmission within 30 seconds. b. 25 points - Transmitted track must be within +/- 10 degrees of the correct track. c. 25 points - Transmitted speed must be within +/- 0.2 mach of the correct speed.
9	TRANSMIT CONTINUING BOGEY BEARING AND RANGE	a. 70 points - Transmit the message "BOGEY xxx,yy" within 12 seconds after the sweep passes the bogey position. The maximum score is given if the transmissions are made 3 out of 5 times and the AIC does not miss making a transmission 2 sweeps in a row. The score is decreased in proportion to transmission omissions. Each out of tolerance transmission deducts 5 points. b. 15 points - Transmit accurate bearing. A maximum score is given if all transmitted bearings are within the specified tolerance. The score is decreased in proportion to the number of out of tolerance transmissions. The tolerance for bearing to bogey is +/- 2 degrees different than NTDS display. Each out of tolerance transmission deducts 3 points.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
9	TRANSMIT CONTINUING BOGEY BEARING AND RANGE (cont.)	c. 15 points - Transmit accurate range. A maximum score is given if all transmitted ranges are within the specified tolerance. The score is decreased in proportion to the number of out of tolerance transmissions. The tolerance for range to bogey is +/- 2 miles different than NTDS display. Each out of tolerance transmission deducts 3 points.
10	ENSURE TEC COMMUNICATIONS SWITCHES ARE CORRECT	a. 40 points - "Talk" switch in left position. b. 30 points - "Left Phone" switch in "RAD" position. c. 30 points - "Right Phone" switch in "INPH" position.
11	ENSURE TEC CONTROL PANEL SWITCHES ARE CORRECT	a. 0 points - CRT Center switch on "OWNSHIP" b. 25 points - Video switch set on "1" c. 25 points - Radar switch set on "4" d. 25 points - Standard leaders switch set to "AIR" e. 25 points - Range switch set to "32"
12	RANGE SCALE AND OFFSET	a. 0 points - The range scale is set to 32 miles at the time the offset is entered. b. 50 points - The offset is entered when the scenario begins. c. 25 points - The range to the PPI center from the ownship position is between 20 and 32 miles when the scenario begins. d. 25 points - The bearing to the center of the PPI from the new ownship position, when the scenario begins, is +/- 20 degrees of the bearing from the new ownship position through the CAP station.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
13	ENTER CAP SYMBOL, PIF, AND STATION ALTITUDE	a. 60 points - The CAP symbol is entered at the time the learner indicates he is ready for the scenario events to commence (foot key + voice input). b. 25 points - The PIF is correct. c. 15 points - The CAP altitude matches the CAP station altitude.
14	"c/s AIRBORNE FOR CONTROL"	100 points - Respond to SWC "c/s AIRBORNE FOR CONTROL" message within 10 seconds with "ROGER". 70 points within 15 seconds.
15	"RUTH, THIS IS c/s ..."	100 points - Respond to CAP CHECK-IN message within 10 seconds with "ROGER". 70 points within 15 seconds.
16	UPDATE CAP SYMBOL	a. 50 points - Update CAP symbol within 18 seconds, after receiving the message "c/s THIS IS c/s ON TACAN STATION xx,yy ANGELS zz, HEADING xxx" from the CAP. 25 points within 24 seconds. b. 50 points for placing CAP symbol within 1/8" of CAP video. 25 points for placing CAP symbol more than 1/8" and less than 1/4".
17	ASK CAP FOR STATE	a. 60 points - Transmit the message "c/s WHAT STATE" to the CAP before arriving on station (outside 5 miles from station) or within 1 minute of SWC request "WHAT STATE". b. 40 points - Respond to CAP message "STATE fff" with "ROGER STATE fff" within 15 seconds. 20 points for response within 20 seconds.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
18	UPDATE NTDS WITH CAP STATE (NON-TRAINING ENVIRONMENT)	a. 100 points - Update state inventory within 75 seconds after each state report from aircrew. b. 70 points - Update state inventory within 90 seconds after each state report from aircrew.
19	NOTIFY SWC OF CONTROL	a. 40 points - Transmit the message "I HAVE CONTROL OF c/s" to the SWC after state report from CAP and before reaching station (5 miles from station). b. 40 points - Respond to SWC "THANK YOU" message with the message "c/s STATE fff" within 24 seconds. 20 points for response within 30 seconds. c. 20 points - Reported state must agree with state message from CAP.
20	"ON STATION"	100 points - Transmit the message "c/s ON STATION" within 36 seconds after the CAP gets within 5 miles of the station. 70 points within 48 seconds within 5 miles.
22	TRANSMIT BOGEY COMPOSITION AND ALTITUDE	a. 60 points - Transmit the message "BOGEY (SINGLE/MULTIPLE) ALTITUDE zz THOUSAND" within 18 seconds of bearing and range call. 30 points for transmisison within 24 seconds. b. 20 points - Transmitted altitude must be the correct altitude. c. 20 points - Transmitted composition must be correct.
23	PLACE BOGEY ON SEQUENCE LIST	100 points - Place bogey on sequence list within 18 seconds of giving initial bogey bearing and range call. 70 points within 24 seconds.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
24	RESPOND TO "JUDY" OR "TALLY HO"	100 points - Do not transmit any bogey data (groundspeed, track, etc.) after the message "TALLY HO" or "JUDY" is received until "LOST CONTACT" or "BREAKAWAY" call is received.
25	LOST CONTACT	100 points - Respond to the message "LOST CONTACT" by transmitting the bearing and range to the bogey starting within 15 seconds of the lost contact message. 70 points within 20 seconds.
26	CONTACT	a. 40 points - Respond to the message "CONTACT xxx,yy" within 10 seconds with one of the following voice calls: "ROGER, YOUR BOGEY TRACKING xxx" or "NEGATIVE, BOGEY xxx,yy". b. 40 points - Respond to the message "CONTACT xxx,yy" with the correct choice described in previous subparagraph. c. 20 points - Respond with accurate tracking or bearing and range data. If tracking data, transmitted track must be within +/-10 degrees of correct track. If bearing and range data, both the bearing and range must meet the following tolerance specifications. Bearing: +/- 2 degrees different from NTDS display. Range: +/- 2 miles different from NTDS display.
27	DISENGAGE CAP FROM BOGEY AT BREAKAWAY	100 points - Disengage the CAP from bogey within 15 seconds after the pilot transmits the "TALLY HO, FOX 1 BREAKAWAY" message. 70 points within 20 seconds.
28	RE-ENGAGE CAP TO STATION AFTER BREAKAWAY	100 points - Engage the CAP to the CAP station within 12 seconds after disengaging the CAP from the bogey. 70 points within 18 seconds.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
29	VECTOR CAP TO STATION AFTER BREAKAWAY	a. 50 points - Transmit the message "c/s (PORT/STARBOARD) xxx" within 18 seconds after engaging the CAP to the CAP station. 25 points for transmission within 24 seconds. b. 30 points - The transmitted turn bearing direction (PORT/STARBOARD) is such that a minimum change in heading is required to come to the correct new heading. c. 20 points - The transmitted heading must be within +/- 10 degrees of the correct heading.
30	REPORT RESULTS OF ENGAGEMENT	a. 40 points - Transmit the message "c/s BREAKING AWAY" to the SWC within 30 seconds after the CAP reports the results of the engagement. 20 points for transmission within 40 seconds. b. 30 points - Transmit the messages "HEADS UP y BOGEYS" or "SPLASH y BOGEYS" or both to the SWC within 35 seconds from the time the CAP reports the results of the engagement. 15 points for transmission within 45 seconds. c. 30 points - The messages must agree with the results as transmitted by the CAP.
31	TRANSMIT "JINK" CALL	a. 40 points - Transmit the message "BOGEY JINKING (LEFT/RIGHT)" within 24 seconds of jink initiation. 25 points for transmission within 30 seconds. b. 40 points - The transmitted direction (LEFT/RIGHT) must be in the correct jink direction. c. 20 points - At the time the jink is initiated no previous jink transmission (BOGEY JINKING LEFT/RIGHT) shall have been made.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
32	TRANSMIT VECTOR TO COUNTER JINK	a. 50 points - Transmit the message "c/s (PORT/STARBOARD) [HARD] xxx" to counter the jink within 36 seconds after jink initiation. 25 points for transmission within 48 seconds after jink initiation. b. 30 points - The transmitted revised heading direction (port/starboard) is such that a minimum change in heading is required to come to the correct new heading. c. 20 points - The new transmitted heading must be within +/- 15 degrees of the correct new heading.
33	TRANSMIT UPDATED BOGEY TRACK	a. 70 points - Transmit the message "BOGEY TRACKING xxx" to the CAP within 1 minute after jink initiation. 35 points for transmission within 90 seconds. b. 30 points - Transmitted new track must be within +/- 15 degrees of correct track.
34	TRANSMIT "BOGEY SPLITTING"	a. 50 points - Transmit the message "BOGEY SPLITTING" within 24 seconds after the bogey initiates a split. 25 points for transmission within 30 seconds. b. 50 points - Disengage bogey and engage new bogey within 24 seconds after the bogey splits.
35	TRANSMIT NEW BOGEY COMPOSITION, ALTITUDE	a. 60 points - Transmit the message "BOGEY SINGLE, ALTITUDE zz" within 18 seconds after the new bogey is engaged. 30 points for transmission within 24 seconds. b. 40 points - Bogey altitude transmitted must be the correct altitude, as displayed.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
36	DETECT AND REPORT STRANGERS	a. 60 points - Transmit to CAP message "STRANGER xxx,yy" after the stranger closes within 10 miles of the CAP and prior to closing within 5 miles of the CAP. b. 20 points - Bearing must be within +/- 10 degrees of correct bearing. c. 20 points - Range must be within +/- 2 miles of correct range.
37	CALL STRANGER BEARING AND RANGE	a. 70 points - Transmit, to CAP, stranger's bearing and range 3 out of 5 times, missing making transmissions no more than 2 sweeps in a row, following initial report until either "VISUAL" call is received from CAP or until stranger range increases for 2 consecutive sweeps. Each out of tolerance transmission deducts 5 points. b. 15 points - Bearing must be within +/- 10 degrees of correct bearing. Each out of tolerance transmission deducts 3 points. c. 15 points - Range must be within +/- 2 miles of correct range. Each out of tolerance transmission deducts 3 points.
38	TRANSMIT STRANGER'S TRACK AND ANGELS	a. 70 points - Transmit, to CAP, stranger's track and angels prior to the stranger closing within 3 miles of the CAP. b. 15 points - Transmitted track must be within +/- 10 degrees of the stranger's heading. c. 15 points - Transmitted angels must be +/- 1000 feet of the stranger's altitude.
39	"STRANGER OPENING"	100 points - Transmit message to CAP "STRANGER OPENING" within 10 seconds of stranger range steadily increasing for 2 sweeps in a row. 70 points within 15 seconds.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
40	TRANSMIT VECTORS FOR RENDEZVOUS	a. 60 points - Respond to the CAP message "REQUEST RENDEZVOUS WITH CRACKERJACK" within 10 seconds with the message "SILVER HAWK VECTOR xxx". 30 points for response within 15 seconds. b. 25 points - Respond to the CAP message "REQUEST RENDEZVOUS WITH CRACKERJACK" within 24 seconds after the CAP request with the message "CRACKERJACK VECTOR xxx" c. 15 points - Silver Hawk vector must be within +/- 10 degrees of the bearing from the CAP to the MAC.
41	ATTAIN CORRECT LATERAL SEPARATION	The lateral separation, perpendicular distance from the line of flight of the CAP to the maneuvering aircraft (MAC), should be the mach number of the MAC X 10 miles when the range between the MAC and the CAP is ((combined mach of the two A/C) X 10) + 2 miles. Score is dependent on the lateral separation error as: 100 points for 0 to 1 mile 75 points for 1 to 2 miles 50 points for 2 to 3 miles 0 points for greater than 3 miles
42	TRANSMIT TO THE MAC THE BEARING AND RANGE TO THE CAP	100 points - Transmit message "C/S, xxx,yy" prior to MAC turn for rendezvous.
43	TRANSMIT MAC'S ALTITUDE TO CAP FOR RENDEZVOUS	a. 70 points - Respond to the CAP message "REQUEST RENDEZVOUS WITH CRACKERJACK" with the message "CRACKERJACK ANGELS zz" within 36 seconds after the vector call. 35 points for response within 48 seconds. b. 30 points - The transmitted altitude must be the correct altitude as displayed in the DRO.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
44	MEASURE RENDEZVOUS FLIGHT PATH	The perpendicular distance from the MAC to the line of flight of the CAP, after turn for rendezvous is complete, should be zero miles. Scores for distance errors are: 100 points for 0 to 1 mile 70 points for 1 to 2 miles 0 points for greater than 2 miles
45	MEASURE RENDEZVOUS SEPARATION	The distance forward from the CAP to the intersecting perpendicular of the MAC along the CAP's line of flight, after the rendezvous is complete, should be within 2 miles. Scores for distance errors are: 100 points for 0 to 1 mile 70 points for 1 to 2 miles 0 points for greater than 2 miles 0 points if the CAP is in front of the MAC
46	TRANSMIT TO THE CAP THE BEARING AND RANGE TO THE MAC	a. 70 points - Transmit the message "c/s xxx,yy" within 10 seconds after the MAC turn for rendezvous is initiated. The maximum score is given if the transmissions are made 3 out of 5 times and the AIC does not miss making transmissions any 2 sweeps in a row. The score is decreased in proportion to the transmission omissions. Each transmission omission deducts 5 points. b. 15 points - Transmit accurate bearing. A maximum score is given if all transmitted bearings are within the specified tolerance. The score is decreased in proportion to the number of out of tolerance transmissions. Bearing tolerance is +/- 2 degrees different from NTDS display. Each out of tolerance transmission deducts 3 points.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
46	TRANSMIT TO THE CAP THE BEARING AND RANGE TO THE MAC (cont.)	c. 15 points - Transmit accurate range. Maximum score is given if all transmitted ranges are within tolerance. The score is decreased in proportion to the number of out of tolerance range transmissions. The tolerance for range to station is +/- 2 miles different from NTDS display. Each out of tolerance transmission deducts 3 points.
47	"FIGHTER IN THE DARK"	100points - Transmit the message "FIGHTER IN THE DARK" if the CAP is on station and 3 consecutive fades of the CAP video occurs, or if the CAP is intercepting the bogey and 2 consecutive fades occur. Message must be transmitted within 15 seconds of the criterion fade. 70 points within 20 seconds.
48	"BOGEY IN THE DARK"	100 points transmits the message "BOGEY IN THE DARK" if the CAP is on station and 3 consecutive fades of the bogey video occurs, or if the CAP is intercepting the bogey and 2 consecutive fades occur. Message must be transmitted within 15 seconds of the criterion fade. 70 points within 20 seconds.
49	TRANSMITTING NTDS DOWN MESSAGE	100 points - Within 18 seconds of NTDS program failure the message "MY OCTOPUS IS BENT" is to be transmitted to the CAP. 70 points within 24 seconds.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
50	INITIAL BEARING AND RANGE TRANSMIT, NTDS DOWN	a. 70 points - Transmit (to the CAP) the message "BOGEY xxx,yy" or "STATION xxx,yy" defining the range and bearing to the bogey or the station within 30 seconds after NTDS program fails. 35 points for transmission within 42 seconds. b. 15 points - Transmitted bearing must be within +/- 5 degrees of the correct bearing for all ranges. c. 15 points - Transmitted range must be within the following tolerances: +/- 3 miles for 0 - 10 miles separation +/- 5 miles for 10 - 20 miles separation +/- 7 miles for 20 - 40 miles separation +/- 10 miles for greater than 40 miles separation.
51	CONTINUOUS BEARING AND RANGE TRANSMIT, NTDS DOWN	a. 60 points - Transmit the message to the CAP "BOGEY xxx,yy" or "STATION xxx,yy" defining the range and bearing to the bogey or the station, 3 out of 5 sweeps (missing making transmissions on no more than 2 sweeps in a row) after the NTDS program fails and the initial call has been made. Each out of tolerance transmission deducts 5 points. b. 20 points - Transmitted bearing must be within +/- 5 degrees of the correct bearing at any range. Each out of tolerance transmission deducts 5 points. c. 20 points - Transmitted range must be within the following tolerances: +/- 3 miles for 0 - 10 miles separation +/- 5 miles for 10 - 20 miles separation +/- 7 miles for 20 - 40 miles separation +/- 10 miles for greater than 40 miles separation. Each out of tolerance transmission deducts 5 points.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
52	ESTABLISHING COMMUNICATIONS AFTER ALARM SOUNDS (BEEPER ON GUARD)	100 points - Transmit message "c/s RADIO CHECK" within 10 seconds after beeper on guard alarm sounds. 70 points within 15 seconds.
53	REPORTING CAP EMERGENCY TO SWC	100 points - Transmit message to SWC "c/s EMERGENCY" within 30 seconds of unsuccessful attempted communications with the CAP. 70 points within 48 seconds.
54	CHECK EMERGENCY PLOT POSITION	a. 70 points - In response to the request, "State bearing and range from ownship to emergency," the learner will make the transmission "EMERGENCY xxx,yy" within 20 seconds. 35 points for response within 30 seconds. b. 15 points - The bearing accuracy must be +/- 5 degrees. c. 15 points - The range accuracy must be +/- 2 miles.
55	SELECT 32 MILE RANGE SCALE FOR SETUPS	Select the 32 mile range scale prior to the CAP going into the OPAREA (100 points).
56	KEEP AIRCRAFT IN THE AREA	Once the aircraft have entered the area, scoring for keeping them in the area is: 10 point deduction for each boundary penetration 100 points for 100 percent of the time.
57	BREAKAWAY	100 points - Transmit the message "c/s (PORT/STARBOARD) xxx" within 25 seconds after receiving the message "FOX 1, BREAKAWAY" from the CAP. 70 points for transmission within 38 seconds.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
58	DISENGAGE PSEUDO BOGEY FROM A POINT-IN-SPACE ("B")	After the pseudo bogey has been engaged from point-in-space "A," hit "BREAK/CANTCO" to disengage after 100 seconds: before 100 seconds = 70 points before 175 seconds = 100 points before 205 seconds = 70 points.
59	DISENGAGE CAP FROM A POINT-IN-SPACE ("A")	When, after breakaway request or the wingman has been detached, the CAP has been engaged to a point-in-space, hit "BREAK/CANTCO" to disengage after 105 seconds: before 105 seconds = 70 points before 180 seconds = 100 points before 210 seconds = 70 points.
62	ENGAGE PSEUDO BOGEY TO PPOI	After the pseudo bogey has been disengaged from a point-in-space, engage pseudo bogey to PPOI by ball tabbing a new point and hitting GEOM within: 12 seconds = 100 points 18 seconds = 70 points.
63	ENGAGE CAP TO PPOI	After engaging pseudo bogey to PPOI, engage CAP to PPOI by sequencing to CAP, by ball tabbing the same point +/- 2 miles, and hitting GEOM within: 12 seconds = 100 points 18 seconds = 70 points.
64	DISENGAGE CAP FROM PPOI	After turn for intercept "FOR BOGEY" is transmitted, sequence to CAP and hit BREAK/CANTCO VAB: before 36 seconds = 70 points within 36 - 48 seconds = 100 points within 49 - 60 seconds = 70 points.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
65	ESTABLISH INITIAL AND FINAL INTERCEPT CONDITIONS	a. When the two aircraft have been turned for the intercept and are steady on course, scoring for the target aspect angle is: 70 points if within 5 degrees of planned 40 points if within 10 degrees of planned 0 points if greater than 10 degrees of planned. b. 30 points - CAP was able to close within 3 miles of pseudo bogey at closest point before crossing.
66	VECTOR CAP TO BOGEY IN TRAINING	After the pseudo bogey is turned for vector as bogey, vector the CAP to the FPOI within: 12 seconds = 100 points 18 seconds = 70 points 24 seconds = 30 points.
67	ENGAGE CAP TO PSEUDO BOGEY IN TRAINING	After the pseudo bogey is steady on the vector as bogey, engage CAP within: 9 seconds = 100 points 15 seconds = 70 points 21 seconds = 30 points.
68	MEASURE SETUP SEPARATION	When the two aircraft have been turned for the intercept and are first steady on course, scoring for separation is: 100 points if within 3 miles of planned 70 points if more than 3 miles but less than 8 miles from planned 0 points if greater than 8 miles from planned.
69	ESTABLISH LOST COMMUNICATIONS	a. 50 points - Transmit the message "c/s LOST COMMUNICATIONS INTENTIONS, OVER" to the CAP within 2 minutes of entering operating area. b. 50 points - Respond to the message "RENDEZVOUS POINT WHISKEY, ANGELS zz" with the message "ROGER LOST COMM" within 5 seconds.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
70	UPDATE NTDS-STATE	a. 50 points - Update CAP state, if required, before CAP turns for intercept ("c/s xxx FOR BOGEY"). b. 50 points - Update pseudeo bogey state, if required, before pseudo bogey turns for intercept ("c/s xxx AS BOGEY").
71	REQUEST PSEUDO BOGEY STATE (TRAINING)	a. 70 points - Transmit the message "c/s WHAT STATE" to the bogey before pseudo bogey turns for intercept ("c/s PSEUDO AS BOGEY"). b. 30 points - Respond to the pseudo bogey's message "STATE fff" with "ROGER, STATE fff" within 15 seconds.
72	REQUEST CAP STATE (TRAINING)	a. 70 points - Transmit the message "c/s WHAT STATE" to the CAP before CAP turns for intercept ("c/s xxx FOR BOGEY"). b. 30 points - Respond to the CAP's message "STATE fff" with "ROGER, STATE fff" within 10 seconds.
73	ENTER CAP SYMBOLS AND PIF	a. 60 points - The CAP symbols for the CAP and pseudo bogey are entered prior to the learner starting the scenario action. b. 20 points - The PIF is correct for the CAP at the start of scenario action. c. 20 points - The PIF is correct for the pseudo bogey at the start of scenario action.
74	RANGE SCALE AND OFFSET (TRAINING ENVIRONMENT)	a. 50 points - The range scale is set to 64 miles at the time the offset is entered. b. 25 points - The offset is entered within 60 seconds from the start of the exercise. c. 25 points - The new ownship position is such that the TACAN station and the area outline are visible on the screen.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
75	UPDATE TURN RATE	The bank angle for both the CAP and the pseudo bogey will be updated to 3 deg/sec in NTDS prior to the beginning of the scenario. (100 points)
76	PSEUDO BOGEY SYMBOL UPDATE	100 points - Maintain the pseudo bogey symbol such that 3 out of 5 times that the video is displayed, missing no more than 2 video displays in a row, the pseudo bogey symbol is within 1/8 inch of the video. This requirement takes effect after the first time that the symbol is updated (assigned to the pseudo bogey video). Each update omission outside stated tolerances deducts 10 points.
77	UPDATE PSEUDO BOGEY SYMBOL	a. 50 points - Update the pseudo bogey symbol within 20 seconds after receiving the CAP check-in message. 25 points for update within 30 seconds. b. Pseudo bogey symbol (CAP2) is placed close to the position of the pseudo bogey video. The scores are: 50 points for 0 to 1/8 inch 25 points for 1/8 - 1/4 inch 0 points for greater than 1/4 inch
78	DIRECT CAP TO CENTER OF AREA	a. 70 points - If aircraft heading is greater than 15 degrees off the heading to the center of the area, transmit message "c/s (PORT/STARBOARD) xxx" within 35 seconds of CAP check-in. [optional] 30 points - within 40 seconds. b. 30 points - If the heading is transmitted, it must be within +/- 15 degrees of the correct heading to the center of the training area.
79	ENGAGE PSEUDO BOGEY TO POINT	After breakaway request or after CAP enters the OPAREA, depress GEOM VAB to get trial engagement of pseudo bogey to a point. within 18 seconds - 100 points within 24 seconds - 70 points.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
80	ENGAGE CAP TO POINT	After receiving a breakaway request or detaching the wingman, ball tab a point, depress GEOM VAB to get trial engagement of CAP to that point within 15 seconds (100 points). Within 21 seconds = 70 points.
81	DETACH WINGMAN	Make the transmission "CRACKERJACK DETACH STARBOARD(p)XXX" within 15 seconds of trial engagement of pseudo bogey (GEOM VAB depressed) to point "B." (point in space) (100 points). 20 seconds = 70 points.
82	DISENGAGE CAP FROM SPLIT AT BREAKAWAY	100 points - Disengage the CAP from the split by hitting BRK/CANTCO within 6 seconds of sequencing to the SWC Break Engage alert DRO. 70 points within 9 seconds.
83	DISENGAGE CAP FROM BOGEY AFTER BREAK ENGAGE ALERT	100 points - After the Break Engagement alert occurs, depress BRK/CANTCO so as to break the CAP's engagement to the bogey within 12 seconds. 70 points within 15 seconds.
84	ENGAGE CAP TO SPLIT	100 points - After the Engage Track alert is received, depress ORD SND to engage the CAP to the split within 12 seconds. 70 points within 15 seconds.
85	VECTOR CAP TO SPLIT	a. 70 points - Transmit the message "c/s PORT/VECTOR/STARBOARD xxx" to the CAP within 18 seconds of sequencing to the SWC Engage alert DRO. 40 points - for transmission within 24 seconds. b. 30 points - Transmit an accurate vector for bogey to the CAP. The heading must be within +/- 10 degrees of the correct heading.

TABLE D3. PERFORMANCE MEASUREMENT VARIABLE DEFINITIONS (CONTIN)

No.	Name	Performance/Scoring Description
86	TRANSMIT INITIAL SPLIT BEARING AND RANGE	a. 40 points - Transmit the message "BOGEY xxx,yy" to the CAP within 18 seconds of vector for split message. 20 points for transmission within 24 seconds. b. 30 points - Transmitted bearing must be within +/- 2 degrees of NTDS displayed data. c. 30 points - Transmitted range must be within +/- 2 miles of NTDS displayed data.
87	TRANSMIT CONTINUING SPLIT BEARING AND RANGE	a. 60 points - Transmit the message "BOGEY xxx,yy" within 12 seconds after the sweep passes the split position. The maximum score is given if the transmissions are made 2 out of 3 times and the AIC does not miss making the transmission any 2 sweeps in a row. The score is decreased in proportion to the transmission omissions. Each out of tolerance transmission deducts 10 points. b. 20 points - Transmit accurate bearing. A maximum score is given if all transmissions are within the specified tolerance. The score is decreased in proportion to the number of out of tolerance transmissions. The tolerance for bearing to split is +/- 2 degrees different than NTDS displayed data. Each out of tolerance transmission deducts 5 points. c. 20 points - Transmit accurate range. A maximum score is given if all transmissions are within the specified tolerance. The score is decreased in proportion to the number of out of tolerance transmissions. The tolerance for range to split is +/- 2 miles different than NTDS displayed data. Each out of tolerance transmission deducts 5 points.

APPENDIX E

INSTRUCTOR/STUDENT MENUS

INTRODUCTION

This appendix presents the menus available for display on the student and instructor CRT terminals. They result from the activation of special function keys on the terminals. A description of the operation and physical layout of the special function keys is presented in the Man-Machine Interaction discussion in Section III. A menu also can be displayed in the course of the execution of the ACE program; for example, some menu selections evoke secondary menus, the identity of which are dependent upon the option selected.

MENUS

Figures E1 through E16 present ACE menu displays. The following conventions apply to the menu display figures:

a. Square brackets ([...]) denote optional lines of CRT text which appear only when applicable. The brackets themselves may not appear.

b. Items within angle brackets (<...>) are generic descriptions of the data to be included. The angle brackets themselves may not appear.

There are eight categories of menus:

THE ABORT MENU SET. The ABORT key invokes the ABORT Key Menu (Figure E1). Selection of item 2 causes the Review Menu (Figure E2) to be displayed. The Current Course Information Menu (Figure E3) is displayed after selection of item 3 of the ABORT Menu.

THE FUNCTION-KEY MENU. Depressing the MENU key causes the Function-key Menu (Figure E4) to be displayed on the CRT. The Idle Mode Menu (Figure E5) is displayed when the instructor station is idle.

THE OVERRIDE MENU SET. The OVERRIDE key invokes the OVERRIDE Key Menu (Figure E6). Selection of option 2 of the OVERRIDE Key Menu causes the Repeat Segment Menu (Figure E7) to be displayed.

THE RETRAIN MENU SET. The RETRAIN key leads to the initial RETRAIN Menu (Figure E8). Subsequently the RETRAIN Option Menu (Figure E9) is displayed.

THE STATS MENU SET. The STATS key invokes the STATS Key Menu (Figure E10). Selecting any student from this menu causes the STATS Type Menu (Figure E11) to be displayed. If option 1 of this menu is chosen, the Segment Summary Request Menu (Figure E12) is displayed. If the STATS Type Menu option 2 is selected, the Speech Training Summary Menu (Figure E13) is displayed. If option 3 of the STATS Type Menu is selected, the Speech Recognition Summary Menu (Figure E14) is displayed.

THE INITIATE VOICE TEST MENU. The INIT VOICE TEST key invokes the INIT VOICE TEST Key Menu (Figure E15). During Voice Test, a Level Vocabulary Menu (Figure E16) will be displayed.

THE NEW T/E MENU SET. The NEW T/E key initiates a series of prompts collecting information necessary to sign on a new student. The information is collected in a NEW T/E Information Display (Figure E17).

THE ^STOP MENU. The ^STOP Key Menu (Figure E18) is displayed as a result of the STOP key being pressed when a student is signed on.

```
                    *** ABORT MENU ***

YOUR OPTIONS AT THIS TIME ARE TO:

    0.  CONTINUE WITH THE INSTRUCTION
    1.  END THIS SESSION WITH ACE
    2.  REVIEW A PREVIOUS SUBJECT
    3.  SEE YOUR CURRENT COURSE INFORMATION

Press the NUMBER of your choice, then press ENTER.
```

May appear on: Student Terminal
Student Terminal, when the instructor functions are enabled there.

Figure E1. Sample ABORT Key Menu

```
*** REVIEW MENU ***

You have the following subjects available for review.
Please press the NUMBER of your choice, then press ENTER.

  0          ESCAPE FROM REVIEW (Return to instruction)
  1 145  IAT Training The Computer (Speech) (Demo)
  2 146  IAT Speech Practice (How To) (Demo)
  3 147  IAT Speech Collection (How To) (Demo)
  4 148  IAT Speech Validation (How To) (Demo)
  5 150  IAT Voice Test (Using This Function) (Demo)
  6 160  IAT Retrain (Using This Function) (Demo)
  7 220  IAT Heading/Bearing and Range To Station (Demo)
  8 221  IAT Tracking The CAP (Demo)
  9 222  IAT Heading To Station (Demo)

[Press NEXT to display additional subjects]
You can end a review at any time by entering ABORT.
```

May appear on: Student Terminal
Student Terminal, when the instructor functions are enabled there.

Figure E2. Sample Review Menu

*** CURRENT COURSE INFORMATION ***

HERE ARE YOUR CHOICES FOR COURSE INFORMATION:

0. CONTINUE WITH THE INSTRUCTION
1. YOUR CURRENT COURSE POSITION
2. THE SCORE FOR YOUR LAST PRACTICE WITHOUT FREEZES
3. YOUR PROGRESS THROUGH THE COURSE

Press the NUMBER of your choice, then press ENTER.

May appear on: Student Terminal

Figure E3. Current Course Information Menu, Option 3 from ABORT Key Menu

```
                    *** FUNCTION-KEY MENU ***

THE FOLLOWING KEYS ARE ENABLED:

    [ ABORT ] [ HELP ] [ OVERRIDE ] [ BREAK ] [ CONTINUE ]
    [ ENABLE_KEYBOARD ] [ DISABLE_KEYBOARD ] [ YES ] [ NO ]
    [ STATS ] [ INIT_VOICE_TEST ] [ STOP_VOICE_TEST ]
    [ RETRAIN ] [ SHIFT_STOP ] [ REPLAY ] [ BYE]
    [ NEW_T/E ]
```

May appear on: Student Terminal
Instructor Terminal
Student Terminal, when the instructor functions are enabled there.

Figure E4. Function-Key Menu

```
*** IDLE MODE MENU ***

LEGAL ENTRIES:

    OVERRIDE -- to override ACE lesson scheduling*
    RETRAIN -- to initiate collection of student speech data*
    STATS -- to display various student statistics
    ENABLE KBRD -- to turn on instructor keys at student station
    DISABLE KBRD -- to turn off instructor keys at student station
    NEW TRAINEE -- enroll a new student on the ACE training system
    ^STOP -- to initiate training system shutdown

              (* only when a student is signed on)
```

May appear on: Instructor Terminal

Figure E5. Idle Mode Menu

```
                    *** OVERRIDE MENU ***

YOUR OPTIONS AT THIS TIME ARE TO:

    0.  LET THE STUDENT CONTINUE
    1.  ADVANCE THE STUDENT TO THE NEXT SEGMENT
    2.  REQUIRE THE STUDENT TO REPEAT A PREVIOUS SEGMENT

Enter the NUMBER of your selection, then press ENTER.
```

May appear on: Instructor Terminal
Student Terminal, when the instructor functions are enabled there.

Figure E6. The OVERRIDE Key Menu

```
*** OVERRIDE REPEAT ***

You have indicated you wish to require a student
to repeat a segment.

Enter the NUMBER of your choice.  Then press ENTER.

    0                CURRENT SEGMENT
    1    101 IAT  Getting On The System
    2    102 IAT  Intro To The Experience Ahead
    3    103 IAT  Intro To The Job Being Trained
    4    104 IAT  Intro To The Pieces Of The System
    5    105 IAT  Quick Look At The Student Station
    6    106 IAT  Quick Look At The Training Console (TEC)
    7    110 IAT  Program Pretest
    8    162 IAT  Break/Review Option
    9    120 IAT  How The Instruction Will Proceed
    [Press NEXT to display additional options]
```

May appear on: Instructor Terminal
Student Terminal, when the instructor functions are entered there.

Figure E7. Sample Repeat Segment Menu, from OVERRIDE Key Menu

*** RETRAIN MENU ***

YOU HAVE ASKED TO TRAIN THE COMPUTER AGAIN. THE FOLLOWING ARE YOUR ACE VOCABULARY CHOICES FOR RETRAINING.

0. CONTINUE WITH THE INSTRUCTION
1. Zero
2. One
3. Two
4. Three
5. Four
6. Five
7. Six
8. Seven
9. Eight

Press the NUMBER of your choice, then press ENTER.

For more choices, press NEXT.

May appear on: Student Terminal
Student Terminal, when the instructor functions are entered there.

Figure E8. Sample RETRAIN Menu

```
        ***  DATA COLLECTION/VALIDATION COMPLETE  ***

THE FOLLOWING ARE YOUR OPTIONS AT THIS TIME:

    0.  CONTINUE WITH THE INSTRUCTION
    1.  RETRAIN ANOTHER PHRASE
    2.  PERFORM SPEECH VALIDATION

Please enter the NUMBER of your choice, then press ENTER.
```

May appear on: Student Terminal
Student Terminal, when the instructor functions are entered there.

Figure E9. RETRAIN Option Menu

```
*** STAT KEY MENU ***

These students' files are available on this disk:

    0.  GO BACK TO WHAT I WAS DOING BEFORE
    1.  <student>
    2.
    3.
    4.
    5.
    6.
    7.
    8.
    9.
   10.
```

May appear on: Instructor Terminal

Figure E10. Identify Other User Menu, Option 3 of STATS Menu Key

```
***  SUMMARY REPORT MENU  ***

HERE ARE YOUR CHOICES FOR STUDENT STATISTICS:

    0.  NONE.  Let's go back to what I was doing before.
    1.  Segment Summary Reports
    2.  Speech Training Summary Reports
    3.  Speech Recognition Summary Reports
        (CP or FP segments only)

Press the NUMBER of your choice, followed by ENTER.
```

May appear on: Instructor Terminal

Figure E11. STATS Type Menu

*** SEGMENT SUMMARY REQUEST ***

YOU ASKED FOR A SEGMENT SUMMARY. WHICH OF THE FOLLOWING CHOICES DO YOU WISH? (SELECT ONE ONLY)

0. NONE. Let's go back to what I was doing before.
1. Summary of all segments (in overall pathway order)
2. Detailed summary of a single segment (by path number)
3. Detailed summary of a specific sequence of segments (by path number)

Press the NUMBER of your choice, followed by ENTER.

May appear on: Instructor Terminal

Figure E12. Segment Summary Request Menu, Option 1 of STATS Type Menu

```
***  SPEECH TRAINING SUMMARY REQUEST  ***

YOU ASKED FOR A SPEECH TRAINING SUMMARY.  WHICH OF THE
FOLLOWING CHOICES DO YOU WISH?

    0.  NONE.  Let's go back to what I was doing before.
    1.  All segments (in overall pathway order)
    2.  A single segment (by path number)
    3.  A specific sequence of segments (by path number)

Press the NUMBER of your choice, followed by ENTER.
```

May appear on: Instructor Terminal

Figure E13. Speech Training Summary Menu, Option 2 of STATS Type Menu

```
*** SPEECH RECOGNITION SUMMARY REQUEST ***

YOU ASKED FOR A SPEECH RECOGNITION SUMMARY. THIS SUMMARY IS AVAILABLE
ONLY FOR THE CP AND FP SEGMENTS THE STUDENT HAS DONE SINCE HE LAST
SIGNED ON. WHICH OF THE FOLLOWING CHOICES DO YOU WISH?

    0. NONE. Let's go back to what I was doing before.
    1. ALL available CP and FP segments (in overall pathway order)
    2. The last completed CP or FP "run"

Press the NUMBER of your choice, followed by ENTER.
```

May appear on: Instructor Terminal

Figure E14. Speech Recognition Summary Menu, Option 3 of STATS Type Menu

*** START VOICE TEST ***

You've asked to test the system's recognition of your voice saying phrases from the training vocabulary. Before proceeding, review the "Rules for Good Speech Recognition" listed in Subsection III A of the Student Guide. Use only the vocabulary items you will see listed on the Student Station CRT.

Your headset and microphone should be adjusted for transmitting.

When you're finished with Voice Test, press STOP VOICE TEST followed by ENTER.

May appear on: Student Terminal
Student Terminal, when the instructor functions are enabled there.

Figure E15. INIT VOICE TEST Key Menu

*** LEVEL 2 VOCABULARY ***

0	10	20	PORT
1	11	30	STARBOARD
2	12	40	VECTOR
3	13	50	FOR BOGEY
4	14	60	STATION
5	15	SILVER HAWK	BOGEY
6	16	CRACKER JACK	BOGEY TRACKING
7	17	ROGER	SPEED POINT
8	18	SAY AGAIN	MARK YOUR TACAN
9	19	CORRECTION	ON STATION

May appear on: Student Terminal
Student Terminal, when the instructor functions are enabled there.

Figure E16. Sample Level Vocabulary (displayed during Voice Test)

```
*** STUDENT SUMMARY ***

Last Name:
Preferred Name:
Social Security Number:
Rate/Rank:
NTDS Input School:
NTDS User School:
Operational NTDS Experience:
Qualified Track Supervisor:

> Is this information correct for this student?  (YES or NO)
```

May appear on: Instructor Terminal

Figure E17. New T/E Information Display

*** STOP KEY MENU ***

A STUDENT IS CURRENTLY SIGNED ONTO THE ACE SYSTEM.

PLEASE DETERMINE THE STUDENT'S STATUS BEFORE YOU SHUT DOWN ACE.

YOUR OPTIONS ARE TO:

0. SHUT DOWN ACE IMMEDIATELY, ACCORDING TO THE INSTRUCTIONS IN THE INSTRUCTOR'S HANDBOOK.
1. SHUT DOWN ACE WHEN THE STUDENT IS DONE WITH HIS CURRENT SEGMENT.

Please enter the NUMBER of your decision, then press ENTER.

May appear on: Instructor Terminal

Figure E18. STOP Key Menu

DISTRIBUTION

Commanding Officer
Naval Training Equipment Center
Orlando, FL 32813 32

Defense Technical Information Center
Cameron Station
Alexandria, VA 22310 12

(All other addressees receive one copy)

Navy Personnel Research and Devel Ctr
Attn: M. McDowell
Library, Code P201L
San Diego, CA 92152

LT Steve Harris
Naval Air Development Center
Code 5021
Warminster, PA 18974

Mr. Warren Lewis
Naval Ocean Systems Center
Human Engineering Branch
Code 8231
San Diego, CA 92152

Commander
Naval Air Systems Command
AIR 413F
Wasington, DC 20361

Chief of Naval Operations
OP-112C1
Washington, DC 20350

Commanding Officer
Fleet Combat Training Center, Pacific
Code 09A
San Diego, CA 92147

Commander
Naval Air Test Center
CT 176
Patuxent River, MD 20670

Commanding Officer
Naval Air Technical Training Center
Code 101, Bldg. 6-54
NAS Memphis (85)
Millington, TN 36054

Mr. Ray Batterfield
Photographer
Naval Air Development Center
Warminister, PA 18974

Dr. Christian P. Skriver
Code 6021
Human Factors Engineering
Naval Air Development Center
Warminister, PA 18974

Dr. Robert A Wisher
Navy Personnel Researach and Development Center
Code P309
San Diego, CA 92152

Director Educational Development
Academic Computering Center
US Naval Academy
Annapolis, MD 21402

Commander
Naval Air Development Center
Attn: Code 6022
Warminister, PA 18974

Dr. Richard W. Oberymayer
Navy Personnel R&D Center
Code P311
San Diego, CA 92152

Mr. Melvyn C. Moy
Navy Personnel Research and Develop.
Information and Decision Processess
Code 305
San Diego, CA 92152

Mr. T. Weiner
Naval Air Development Center
Code 4043
Warminister, PA 10974

Mr. Ernest E. Poor
Naval Air Systems Command
AIR 4138
Room 336
Washington, DC 20361

Dr. Jesse Orlansky
Institute for Defense Analyses
Science and Technology Division
1801 N. Beauregard St.
Alexandria, VA 22311

Commander
Naval Air Force
US Pacific Fleet (Code 316)
NAS North Island
San Diego, CA 92135

Commander
Naval Air Systems Command
AIR 413G
Washington, DC 20361

Chief of Naval Operations
OP-987H
Attn: Dr. R. G. Smith
Washington, DC 20350

Commander
Naval Air Systems Command
Technical Library
AIR-9500
Washington, DC 20361

Commander
Naval Air Force
US Pacific Fleet (Code 342)
NAS North Island
San Diego, CA 92135

Chief of Naval Operations
OP-5936
Washington, DC 20350

Robert F. Lawson, CDR, USN (Ret)
Naval Applications Engineer
ONR Scientific Department
1030 E. Green Street
Pasedena, CA 91106

OUSDR&E (R&AT) (E&LS)
CDR. Paul R. Chatelier
Washington, DC 20301

Dr. Donald Connolly
Research Psychologist
Federal Aviation Administration
FAA NAFEC ANA-230 Bldg. 3
Atlantic City, NJ 08405

Commanding Officer
Fleet Anti-Submarine Warfare Training Center, Pacific
Attn: Code 001
San Diego, CA 92147

Commander
Training Command
Attn: Educational Advisory
US Pacific Fleet
San Diego, CA 92147

Commandant
US Army Field Artillery School
Counterfire Department
Attn: Eugene C. Rogers
Ft. Sill, OK 73503

Dr. Henry J. Dehaan
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Tice De Young
US Army Engineer Topographic Laboratires Research Institute
Ft. Belvoir, VA 22060

Commandant
US Army Field Artillery School
ATSF-TD-TS
Mr. Inman
Ft. Sill, OK 73503

Mr. Lockwood Reed
US Army Avionics R&D Activity
DAVAA-E
Ft. Monmouth, NJ 07703

Army Training Support Center
ATTSD-DS-MSS/MAJ Richardson
Ft. Eustis, VA 23604

Director
US Army Human Engineering Laboratory
Attn: DRXHE-HE (Keesee)
Aberdeen Proving Ground, MD 21005

USAHEL/USAAVNC
Attn: DRXHE-FR (Dr. Hofmann)
P. O. Box 476
Ft. Rucker, AL 36362

Director
Air University Library
Maxwell AFB, AL 36100

Mr. Don F. McKechnie
Research Psychologist
AFAMRL/HEF
Wright-Patterson AFB
Dayton, OH 45433

Mr. Harold A. Kottmann
ASD/YME
Wright Patterson AFB, OH 45433

US Air Force Human Resources Lab
AFHRL-TT (Dr. Rockway)
Technical Training Division
Lowry AFB, CO 80230

Headquarters
Air Training Command, XPTI
Attn: Mr. Goldman
Randolph AFB, TX 78148

AFHRL-OTO
Attn: Mr. R. E. Coward
Luke AFB, AZ 85309

Human Resources Laboratory
Wright-Patterson AFB, OH 45433

LT Dave Cooper
AFHRL/OTT
Wright-Patterson AFB, OH 45433

CAPT Vince Mortimer
AFAMRL/BBM
Wright-Patterson AFB
Dayton, OH 45433

ASD/ENETC
Mr. R. G. Cameron
Wright-Patterson AFB, OH 45433

US Air Force Human Resources Lab
TSZ
Brooks AFB, TX 78235

Chief, Methodology and Standard Staff
Federal Aviation Administration
Academy
Aeronautical Center
P. O. Box 25082
Oklahoma City, OK 73125

Mr. John T. Masterson
US Postal Lab
11711 Parklawn Drive
Rockville, MD 20852

LT COL J. W. Bierman
HQ MC (Code APC)
Washington, DC 20380

Aerospace Psychology Department
Naval Aero Medical Research Lab (L53)
Naval Air Station
Pensacola, FL 32508

Mr. P. J. Andrews
Naval Sea Systems Command
Code 01R2
Room 380, Crystal Plaza 6
Washington, DC 20362

Dr. Henry M. Halff
Office of Naval Research
Code 458
Arlington, VA 22217

Chief of Naval Research
Code 458
800 N. Quincy Street
Arlington, VA 22217

Dr. Marshall J. Farr
Director, Personnel and Training Research Program
Office of Naval Research
Arlington, VA 22217

DATE
ILMED
8