



Research Product 89-05

Embedded Training Software Specifications for the FOG-M System Demonstration

AD-A208 757

January 1989

Manned Systems Group
Systems Research Laboratory

U.S. Army Research Institute for the Behavioral and Social Sciences

SDTICD
ELECTE
JUN 09 1989

[Handwritten signature]

[Handwritten mark]

Approved for public release, distribution is unlimited

89 6 09 031

BEST

AVAILABLE

COPY

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

**A Field Operating Agency Under the Jurisdiction
of the Deputy Chief of Staff for Personnel**

EDGAR M. JOHNSON
Technical Director

JON W. BLADES
COL, IN
Commanding

U.S. ARMY PROJECT MANAGER FOR TRAINING DEVICES

WILLIAM MARROLETTI
Deputy Project Manager

RICHARD J. LUNDSFORD, JR.
Col, EN
Project Manager

Research accomplished under contract
for the Department of the Army

Applied Science Associates, Inc.

Technical review by

Irving N. Alderman
Stanley F. Bolin
Randall M. Chambers

NOTICES

FINAL DISPOSITION: This Research Product may be destroyed when it is no longer needed.
Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: This Research Product is not to be construed as an official Department of the Army document, unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS --		
2a. SECURITY CLASSIFICATION AUTHORITY --			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE --					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) VRI-ASA-1.1FR86-1			5. MONITORING ORGANIZATION REPORT NUMBER(S) ARI Research Product 89-05		
6a. NAME OF PERFORMING ORGANIZATION Vector Research, Inc.		6b. OFFICE SYMBOL (If applicable) --		7a. NAME OF MONITORING ORGANIZATION U.S. Army Research Institute for the Behavioral and Social Sciences	
6c. ADDRESS (City, State, and ZIP Code) P.O. Box 1506 Ann Arbor, MI 48106			7b. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-5600		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Same as 7a.		8b. OFFICE SYMBOL (If applicable) --		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA903-85-C- 0078	
8c. ADDRESS (City, State, and ZIP Code) Same as 7b.			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO. 62717A	PROJECT NO. 790	TASK NO. 142
			WORK UNIT ACCESSION NO. 127C1		
11. TITLE (Include Security Classification) Embedded Training Software Specifications for the FOG-M System Demonstration					
12. PERSONAL AUTHOR(S) Meerschaert, Mark; Rainaldi, William; Smith, Robert; Thompson, David (Vector Research, Inc.); Frederick, Charles; and Wheaton, Kenneth (Continued)					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM 84/11 TO 86/02		14. DATE OF REPORT (Year, Month, Day) 1989, January	
15. PAGE COUNT 1					
16. SUPPLEMENTARY NOTATION Contracting Officer's Representative, Dorothy L. Finley.					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Embedded training; Software specifications;		
			Training system development; Structured specifications;		
			FOG-M missile; Simulation (K +)		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) → Embedded training (ET) concepts are under investigation for the FOG-M system being developed by the U.S. Army Missile Command. This document presents ET software specifications for a demonstration of FOG-M. Specifications are included for embedded mission simulation, computer-aided instruction, and part-task training of the FOG-M gunner. The specifications are intended for use by FOG-M programmers who will design and code the ET software. The approach used in describing the software is that of structured specifications. <i>Keywords: Fibre optic guided missile (FOG-m);</i>					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL Dorothy L. Finley			22b. TELEPHONE (Include Area Code) (401) 91-5523		22c. OFFICE SYMBOL PERI-SM

ARI Research Product 89-05

12. PERSONAL AUTHOR(S) (Continued)

(Integrated Graphics Systems, Inc.)



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Research Product 89-05

Embedded Training Software Specifications for the FOG-M System Demonstration

**Mark Meerschaert, William Rainaldi, Robert Smith,
and David Thompson**

Vector Research, Inc.

Charles Frederick and Kenneth Wheaton

Integrated Graphics Systems, inc.

Manned Systems Group

John F. Hayes, Chief

Systems Research Laboratory

Robin L. Keesee, Director

U.S. Army Research Institute for the Behavioral and Social Sciences

5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel

Department of the Army

January 1989

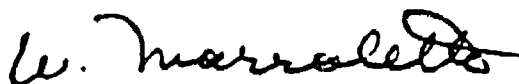
**Army Project Number
2Q162717A790**

**Human Performance
Effectiveness
and Simulation**

Approved for public release; distribution is unlimited

FOREWORD

This paper is one of a series being produced by Applied Science Associates, Incorporated, (ASA) and its subcontractors for the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) to investigate using embedded training (ET) for U.S. Army systems during the system development process. Under contract to ASA, Vector Research, Incorporated, (VRI) and its subcontractor, Interactive Graphic Systems, Incorporated, (IGS) are investigating the incorporation of ET into the Fiber-Optic Guided Missile (FOG-M) system being developed at the Army Missile Laboratory (AML) of the U.S. Army Missile Command at Huntsville, Alabama. This report by VRI and IGS presents structured specifications programmers can use in designing and coding the ET software to be incorporated in a demonstration of the developmental FOG-M system. A companion volume describes the FOG-M ET courseware to be run on the ET software.



WILLIAM MARROLETTI
Deputy Project Manager



EDGAR M. JOHNSON
Technical Director
Army Research Institute

ACKNOWLEDGMENTS

The authors received useful suggestions and insights from several persons involved in the development of the FOG-M system. We are especially grateful for the assistance of Wanda Hengel, Dan Reed, and Jim Baumann of the Army Missile Laboratory (AML) and of Cathy Farless of the Computer Sciences Corporation. Valuable information on the AML FOG-M mission simulation was supplied by Dave Williams of the Computer Sciences Corporation. Paul Beckwith of the Harris Corporation provided information on the Digital Perspective Generator (DPG).

Guidance on the scope of the courseware of interest in a FOG-M ET demonstration was provided by Dr. Jan Ditzian of Applied Science Associates, Incorporated (ASA), and George Purifoy of ASA was the principal investigator.

We would also like to acknowledge the contributions of ARI personnel to the research effort. Dorothy L. Finley served as the Contracting Officer's Representative. She and Dr. Irving Alderman provided the opportunity for project staff to become familiar with ARI's research program and with Army developments related to FOG-M and ET.

EMBEDDED TRAINING SOFTWARE SPECIFICATIONS FOR THE FOG-M SYSTEM DEMONSTRATION

CONTENTS

	Page
SECTION 1. INTRODUCTION	1-1
SECTION 2. APPROACH	2-1
Background to FOG-M ET	2-1
Conventions for the Structured Specifications	2-3
Overview of the Specifications	2-4
SECTION 3. DATA FLOW DIAGRAMS	3-1
SECTION 4. DATA DICTIONARY	4-1
SECTION 5. PROCESS DESCRIPTIONS	5-1
APPENDIX A. DATA FLOW DIAGRAMS OF VIDEO DISK FLIGHT SIMULATION	A-1
B. DESCRIPTIONS OF PROCESSES INTERFACING WITH FOG-M HARDWARE .	B-1
GLOSSARY	G-1
REFERENCES	R-1

LIST OF DATA FLOW DIAGRAMS AND MINI-SPECS IN OUTLINE FORM

0.	FOG-M Embedded Training System (top level)	3-3
1.	Supervise Lesson Menus.	3-5
1.1	Sequence Menus	3-7
1.1.1	Select Menu Item	5-3
1.1.2	Select Menu Type	5-4
1.2	Build Menu	3-9
1.2.1	Sequence Menu Type's Items	5-5
1.2.2	Get Menu Item	5-6
1.2.3	Build Menu Selection Rules	5-7
1.3	Load Menu Item Courseware	3-11
1.3.1	Loan Menu Item Description	5-8
1.3.2	Loan Menu Item Performance Record	5-9
1.4	Build Menu I/O Rules	3-13
1.4.1	Build Menu Output	5-10
1.4.2	Build Menu Input Rules	5-11

CONTENTS (Continued)

	Page
2. Supervise Training	3-15
2.1 Supervise Item	3-17
2.1.1 Load and Distribute Item Courseware	5-12
2.1.2 Sequence Item Parts	5-13
2.2 Supervise M/C Training	3-19
2.2.1 Supervise M/C Segment	3-21
2.2.1.1 Load and Distribute M/C Segment Courseware	5-14
2.2.1.2 Sequence M/C Segment Parts	5-15
2.2.2 Supervise M/C Subsegments	3-23
2.2.2.1 Load and Distribute M/C Subsegment Courseware	5-16
2.2.2.2 Sequence M/C Subsegments	5-17
2.3 Supervise P/D Training	3-25
2.3.1 Supervise P/D Segments	3-27
2.3.1.1 Load and Distribute P/D Segment Courseware	5-18
2.3.1.2 Sequence P/D Segment Parts	5-19
2.3.2 Supervise P/D - S/D Target Subsegment (CGI)	3-29
2.3.2.1 Load and Dist. P/D - S/D Tgt Subsegment Courseware	5-20
2.3.2.2 Sequence P/D - S/D Target Subsegments	5-21
2.3.3 Supervise P/D - M/S Target Subsegment (DMG)	3-31
2.3.3.1 Load and Dist. P/D - M/S Subsegment Courseware	5-22
2.3.3.2 Sequence P/D - M/S Target Subsegments	5-23
2.4 Supervise Mission Training	3-33
2.4.1 Supervise Mission Segment	3-35
2.4.1.1 Load and Distribute Mission Segment Courseware	5-24
2.4.1.2 Sequence Mission Segment Courseware	5-25
2.4.2 Supervise Launch Training Subsegment	3-37
2.4.2.1 Load and Dist. Launch Subsegment Courseware	5-26
2.4.2.2 Sequence Launch Subsegment Courseware	5-27
2.4.3 Supervise Cruise Training Subsegment	3-39
2.4.3.1 Load and Dist. Cruise Subsegment Courseware	5-28
2.4.3.2 Sequence Cruise Subsegment Courseware	5-29
2.4.4 Supervise Target Area Training Subsegment	3-41
2.4.4.1 Load and Dist. Target Area Subsegment Courseware	5-30
2.4.4.2 Sequence Target Area Subsegment Courseware	5-31
2.4.5 Supervise Lock-on Training Subsegment	3-43
2.4.5.1 Load and Dist. Lock-on Subsegment Courseware	5-32
2.4.5.2 Sequence Lock-on Subsegment Courseware	5-33
2.4.6 Supervise Impact Training Subsegment	3-45
2.4.6.1 Load and Dist. Impact Subsegment Courseware	5-34
2.4.6.2 Sequence Impact Subsegment Courseware	5-35
3. Supervise Assessment	3-47
3.1 Update Item Record	5-36

CONTENTS (Continued)

	Page
3.2	Supervise M/C Assessment 3-49
3.2.1	Supervise M/C Subsegment Assessment 3-51
3.2.1.1	Compare M/C Keypad Answer 5-37
3.2.1.2	Compare M/C PDP Answer 5-38
3.2.2	Supervise M/C Segment Assessment 3-53
3.2.2.1	Tabulate M/C Segment 5-39
3.2.2.2	Evaluate M/C Segment 5-40
3.3	Supervise P/D Assessment 3-55
3.3.1	Supervise P/D - M/S Subsegment Assessment 3-57
3.3.1.1	Compare M/S Target Response 5-41
3.3.1.2	Issue M/S Target Locations 5-42
3.3.1.3	Compare M/S Target to Other Targets 5-43
3.3.1.4	Issue M/S Immediate Feedback and Results 5-44
3.3.2	Supervise P/D - M/S Segment Assessment 3-59
3.3.2.1	Tabulate P/D - M/S Segment 5-45
3.3.2.2	Evaluate P/D - M/S Segment 5-46
3.3.3	Supervise P/D - S/D Subsegment Assessment 3-61
3.3.3.1	Compare S/D Target Response 5-47
3.3.3.2	Compare S/D Target Response Time 5-48
3.3.3.3	Sum Over All Target Response Times 5-49
3.3.3.4	Issue S/D Immediate Feedback and Results 5-50
3.3.4	Supervise P/D - S/D Segment Assessment 3-63
3.3.4.1	Tabulate P/D Segment 5-51
3.3.4.2	Evaluate P/D Segment 5-52
3.4	Supervise Mission Assessment 3-65
3.4.1	Supervise Mission Segment Evaluation 3-67
3.4.1.1	Tabulate Mission Segment 5-53
3.4.1.2	Evaluate Mission Segment Courseware 5-54
3.4.2	Supervise Launch Subsegment Assessment 3-69
3.4.2.1	Compare Launch State Vectors 5-55
3.4.2.2	Evaluate Launch Subsegments 5-56
3.4.3	Supervise Cruise Subsegment Assessment 3-71
3.4.3.1	Compare Cruise Parameters 5-57
3.4.3.2	Evaluate Cruise Subsegment 5-58
3.4.4	Supervise Target Area Subsegment Assessment 3-73
3.4.4.1	Compare Target Area Subsegment Courseware 5-59
3.4.4.2	Evaluate Target Area Subsegment 5-60
3.4.5	Supervise Lock-on Subsegment Assessment 3-75
3.4.5.1	Compare Lock-on Coordinates 5-61
3.4.5.2	Evaluate Lock-on 5-62
3.4.5.3	Compare All Coordinates 5-63
3.4.6	Supervise Impact Subsegment Assessment 3-77
3.4.6.1	Compare Impact Locations 5-64
3.4.6.2	Evaluate Impact Subsegment 5-65

CONTENTS (Continued)

	Page
4.1	Perform CAI B-3
4.2a	Perform M/C Keypad Subsegment B-4
4.2b	Perform M/C PDP Subsegment B-5
4.3a	Perform S/D Target Subsegment B-6
4.3b	Perform M/S Target Subsegment B-7
4.4a	Perform Video Disk Simulation A-3
4.4.1	Simulate Missile A-5
4.4.1.1	Determine Present Position B-9
4.4.1.2	Determine Speed and Heading B-10
4.4.1.3	Determine Accelerations B-11
4.4.2	Simulate Seeker Video A-7
4.4.2.1	Calculate Seeker Simulation A-9
4.4.2.1.1	Calculate Projected Position B-12
4.4.2.1.2	Calculate Current and Projected Image of Seeker . . . B-13
4.4.2.1.3	Calculate Seeker Angles and Downlink B-14
4.4.2.2	Determine Video Parameters A-11
4.4.2.2.1	Calculate Manipulation Polynomial B-15
4.4.2.2.2	Calculate Pan Instructions B-16
4.4.2.2.3	Calculate Next Frame B-17
4.4.2.2.4	Calculate Relative Projected Vector B-18
4.4.2.3	Search Video Database A-13
4.4.2.3.1	Digitize and Recover Encoded Information B-19
4.4.2.3.2	Switch Frame Buffers B-20
4.4.2.3.3	Select Sub-frame from Frame B-21
4.4.2.3.4	Manipulate Image (Warper) B-22
4.4.2.3.5	Convert Digital Frame to Analog B-23
4.4.3	Supervise Simulation A-15
4.4.3.1	Initial Conditions B-24
4.4.3.2	Determine Ending Conditions B-25
4.4.3.3	Determine Initial Speed and Heading B-26
4.4.3.4	Retrieve Historical Data B-27
4.4b	Perform DPG Flight Simulation B-8

EMBEDDED TRAINING SOFTWARE SPECIFICATIONS FOR THE FOG-M SYSTEM DEMONSTRATION

SECTION 1

INTRODUCTION

This document presents structured specifications for the embedded training (ET) software of the Fiber-Optic Guided Missile (FOG-M) system demonstration (referred to as the FY87 demonstration). It is written for programmers who will be designing and coding the software to implement embedded training on the FOG-M. For this reason the document is intended to be accessible principally to persons conversant with the hardware and software of the FOG-M system, although other readers will find it useful (e.g., to understand the general scope of courseware that the specifications accommodate).

The overall system concept for FOG-M ET, including the hardware configuration, has been documented previously (ASA, 1985a). Software specifications were designed especially for implementation on that system concept. More specifics of that system concept appear in this report, although others remain to be developed in conjunction with development of prototype software. Whenever technical details remain, the authors attempted to make the specifications sufficiently modular that programmers will be able to fill in programming details as technical solutions are found, without altering the overall structure of the specifications.

The project staff's approach to writing software specifications was that of structured specification, a standard technique in wide use in the software development community, including developers of real-time systems. For specifics of style we followed the approach of DeMarco (1978), although the same principles (and many conventions of style) are described in such sources as (Myers, 1978), (Yourdon, 1976), and (Yourdon and Constantine, 1975).

Section 2 discusses structured specifications, along with other background on the ET software specifications. A complete specification of software is given by data flow diagrams, a data dictionary, and process descriptions (or mini-specs). These terms are defined later in Section 2. Sections 3 through 5 contain the FOG-M ET data flow diagrams, data dictionary, and process descriptions, respectively. Two appendices contain data flow diagrams and process descriptions directly interfacing with FOG-M hardware.

SECTION 2

APPROACH

To help readers understand the specifications, this section presents a brief background. The first part summarizes the background to ET for the FOG-M demonstration; the second summarizes the conventions of the structured software specifications; and the third makes a brief overview of the FOG-M specifications.

Background to FOG-M ET

The specifications of this report were developed for the FOG-M system concept defined in ASA, 1985a. A critical component of that concept was a mission simulation for the FOG-M gunner station. At the time of writing these specifications there were two hardware options, a video disk player and a Digital Perspective Generator (DPG), for simulating visual scenes for the FOG-M gunner's station. Specifications for the software have been structured with this uncertainty in mind. (The video disk player has other uses, such as presentation of stills, however.) That is, some parts of the specifications are relatively unrelated to the remaining technical issues, and the remaining parts of the specifications are of a modular nature (i.e., such that the only responses to resolving the technical issues are likely to be the relabeling of data flows at interfaces and the addition of details to process descriptions). In fact, software specifications are (to a great degree) robust with respect to the details of implementation.

Several factors have combined to determine the types of ET (and manner of delivery of the instructional materials) for which the demonstration specifications were designed. First, the demonstration ET is

limited to tasks related to the use of the gunner's station. Consequently, demonstration software will focus on the gunner's station, with other aspects of the FOG-M (e.g., maintenance) and a more comprehensive set of instructional materials pending work on post-demonstration versions of FOG-M.

Second, there is a more specific emphasis on maintaining the gunner's missile flight skills through mission simulation as an aspect of the demonstration most likely to demonstrate the power of ET. This means the flight of a simulated mission that looks to the gunner as much as possible like a real FOG-M mission and which interacts with the gunner as much as possible as occurs in a real mission. In FOG-M versions following the demonstration it may be possible to do even more of a pedagogical nature with partial mission simulations (e.g., to make feedback assessments of some sort to the trainee while the simulation is running). For the demonstration the simulation will handle fixed segments of a mission and assess performance at the end of the segments -- i.e., it will look exactly like a mission or a segment of a mission to the trainee.

Third, the demonstration does not require certain components of an instructional system that would be of interest in an operational ET system or in a training testbed. In an operational system there might be a requirement for review of trainee performance (say, by a training officer), and the specifications for operational ET would have to include provisions for software to perform the instructor-review function. Specification (and subsequent programming) of review facilities are straightforward, and no barriers are currently foreseen to their inclusion in later FOG-M ET systems. Although performance review facilities are not part of the demonstration specifications, the demonstration specifications maintain records of trainee completion of instructional units (for use in making suggestions to the trainee on sequencing instructional units) during the trainee's session. Thus, the essential information

needed by such a review facility is already in the demonstration specifications. Another aspect of operational systems that lies beyond the scope of demonstration is the development of extensive authoring facilities for ET courseware.

In spite of the demonstration nature of the specifications, the ET system so specified is flexible. While the capabilities of the FOG-M system do not provide an appropriate host for such advanced instructional facilities as are found in artificial intelligence-based CAI systems, for example, the hierarchical (menu-like) organization of instructional materials admits of considerable robustness, with its modular provision for sequencing rules.

Conventions for the Structured Specifications

The project staff's guidelines for structured specifications were the conventions of DeMarco (1978), to which this document adheres very closely. As a specification document, it identifies what the software is to do, but not how to implement it in code. It defines data, specifies processes to be performed on the data, and illustrates data flows, but does not impose control structures. The resulting document is a more rigorous B5 specification than a more informal approach, which is all that MIL-STD-490 (DoD, 1968), requires. (The latter is not very specific about the manner in which the specifications are to be presented.)

The software specifications consist of three parts: data flow diagrams, a data dictionary, and process descriptions. Data flow diagrams record the partitioning of the problem from the point of view of the data: they show functional interfaces, and do not specify flows of control. The data dictionary identifies all the interfaces in detail by defining the data flows; high-level flows are divided into lower-level components -- repeatedly, if need be, until elemental and operationally meaningful data flows are defined. The process descriptions (also called

mini-specs) describe the primitive processes, i.e., those at the lowest levels of the data flow diagrams. They are algorithmic descriptions of the primitive processes and are written either in structured English (which is similar to pseudo-code) or in straight English text (especially when additional guidance is needed for program design).

Overview of the Specifications

At the top level the FOG-M Embedded Training System has been partitioned into four processes: (1) Supervise Lesson Menu; (2) Supervise Training; (3) Supervise Assessment; and (4) Perform ET. The diagram shows the flow of data between each of the major software processes of the system and the FOG-M physical components (Perform Training).

Process 1.0, Supervise Lesson Menus, is composed of the following processes: 1.1 Sequence Menus; 1.2 Build Menus; 1.3 Load Menu Item Courseware; and 1.4 Build Menu Input/Output (I/O) Rules, which sequence the user through the courseware menu structure which consists of top level menus, lesson menus, and topic menus. Upon selecting an item from a topic menu a topic reference is passed to the supervise training process. When a lesson item has been completed, the user is presented with either a passed segment menu or failed segment menu, depending upon the item grade. This allows the user to proceed with training, repeat the item, or select an entirely different lesson.

Process 2.0, Supervise Training consists of four processes: 2.1 Supervise Item; 2.2 Supervise Multiple Choice (M/C) Training; 2.3 Supervise Point Disk (P/D) Training; and 2.4 Supervise Mission Training. Process 2.1, the item supervisor, selects the training supervisor type which is unique to that item (including CAI). Process 2.2 supervises a sequence of multiple choice questions. Process 2.3 supervises point disk training which is used for practicing the manipulative (hand-eye) skills needed by the gunner.

Process 2.4 supervises any mission related training. This consists of training ranging from repeated specific mission phases to an entire (launch to impact) mission simulation.

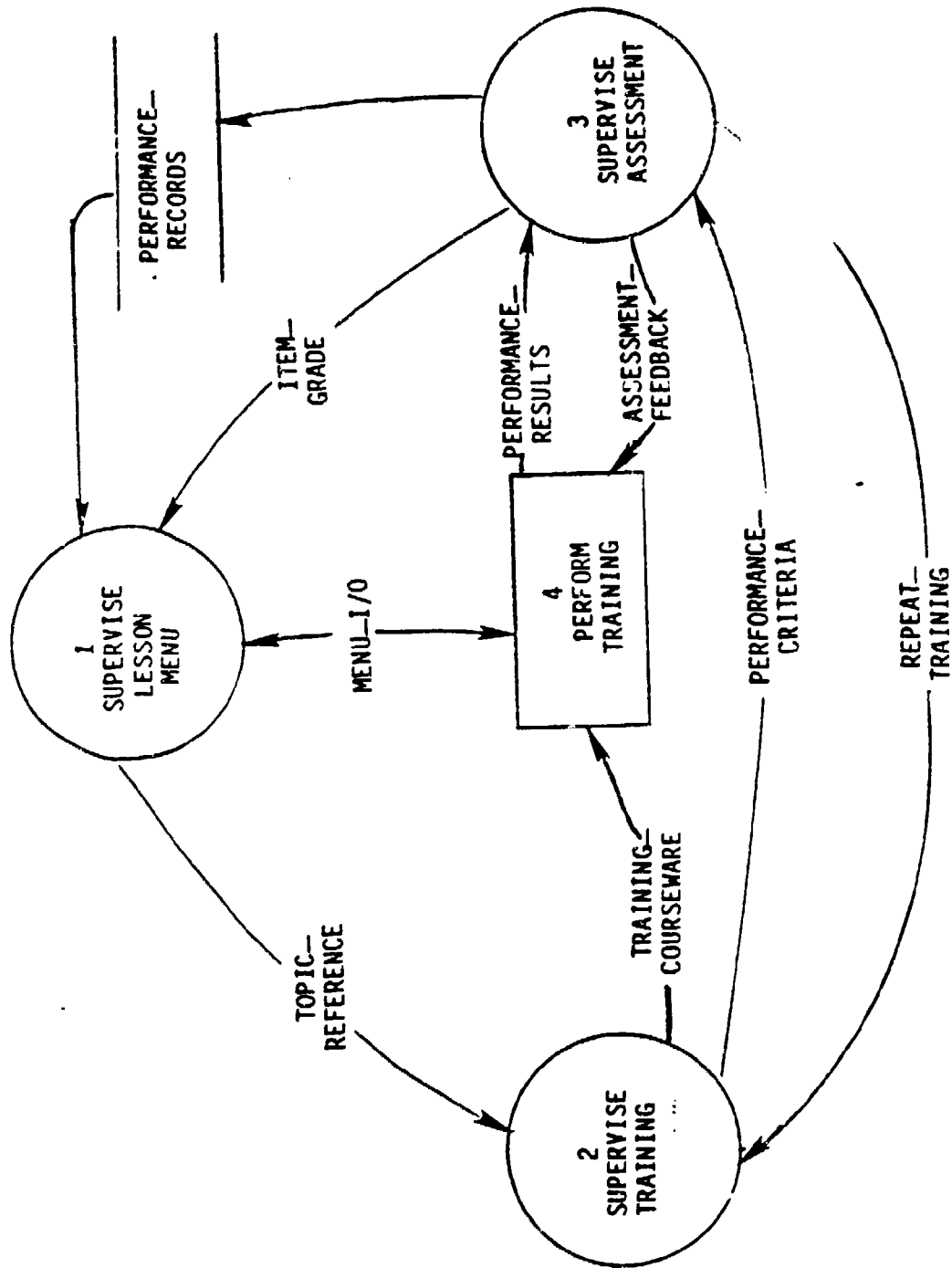
Process 3.0, Supervise Assessment is comprised of four processes: 3.1 Supervise Item Assessment; 3.2 Supervise M/C Assessment; 3.3 Supervise P/D Assessment; and 3.4 Supervise Mission Assessment. These processes complement and directly correspond to the training supervisor process.

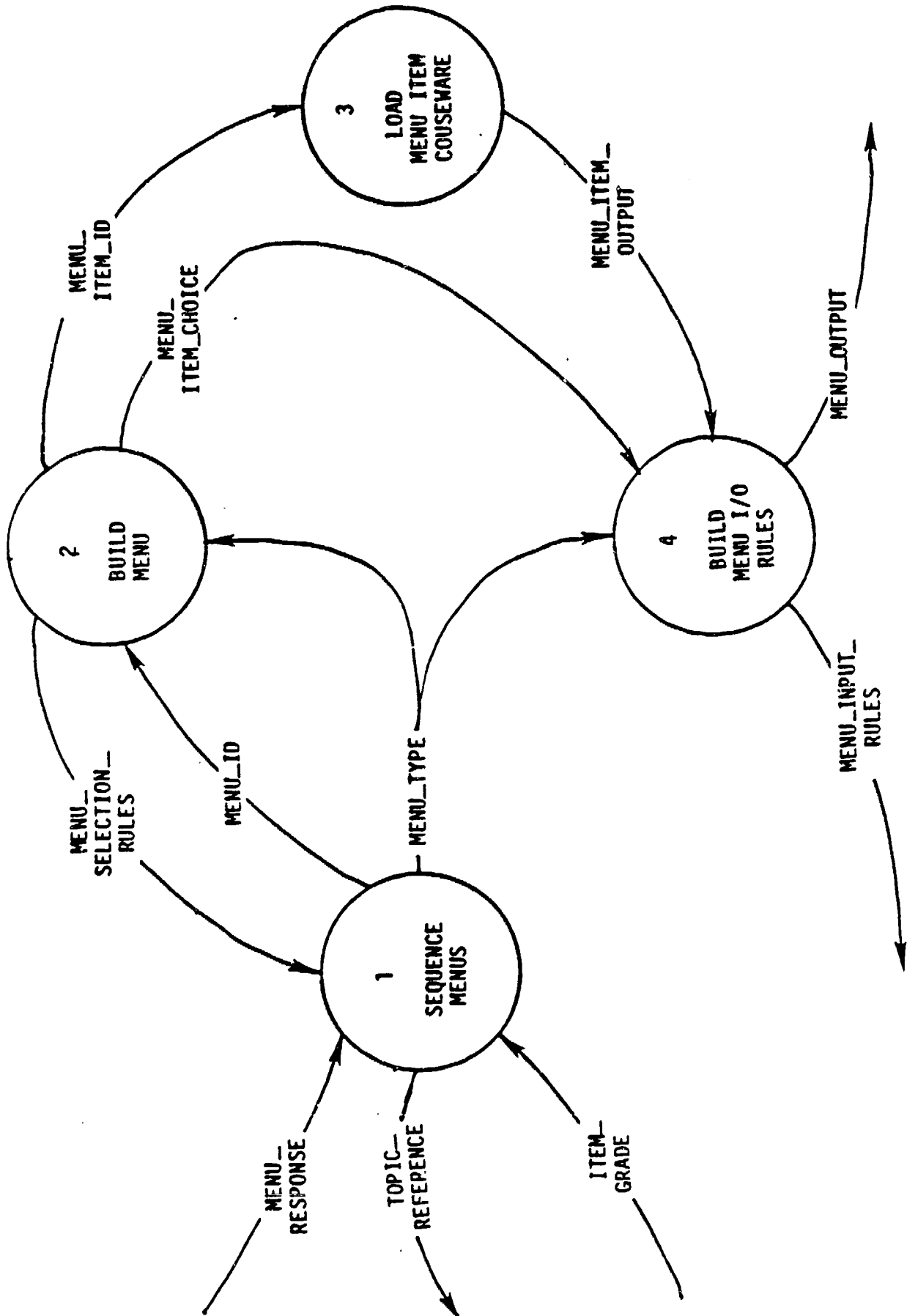
Process 4.0, Perform Training, described in the appendices, consists of implementation-dependent processes (both hardware and software). Except for the missile-seeker simulator process, all processes have been described in a generic fashion, i.e., in terms of what is required of them, not how they operate. This was done because the configuration of the FOG-M system, at the writing of this document, is still subject to some modifications.

SECTION 3

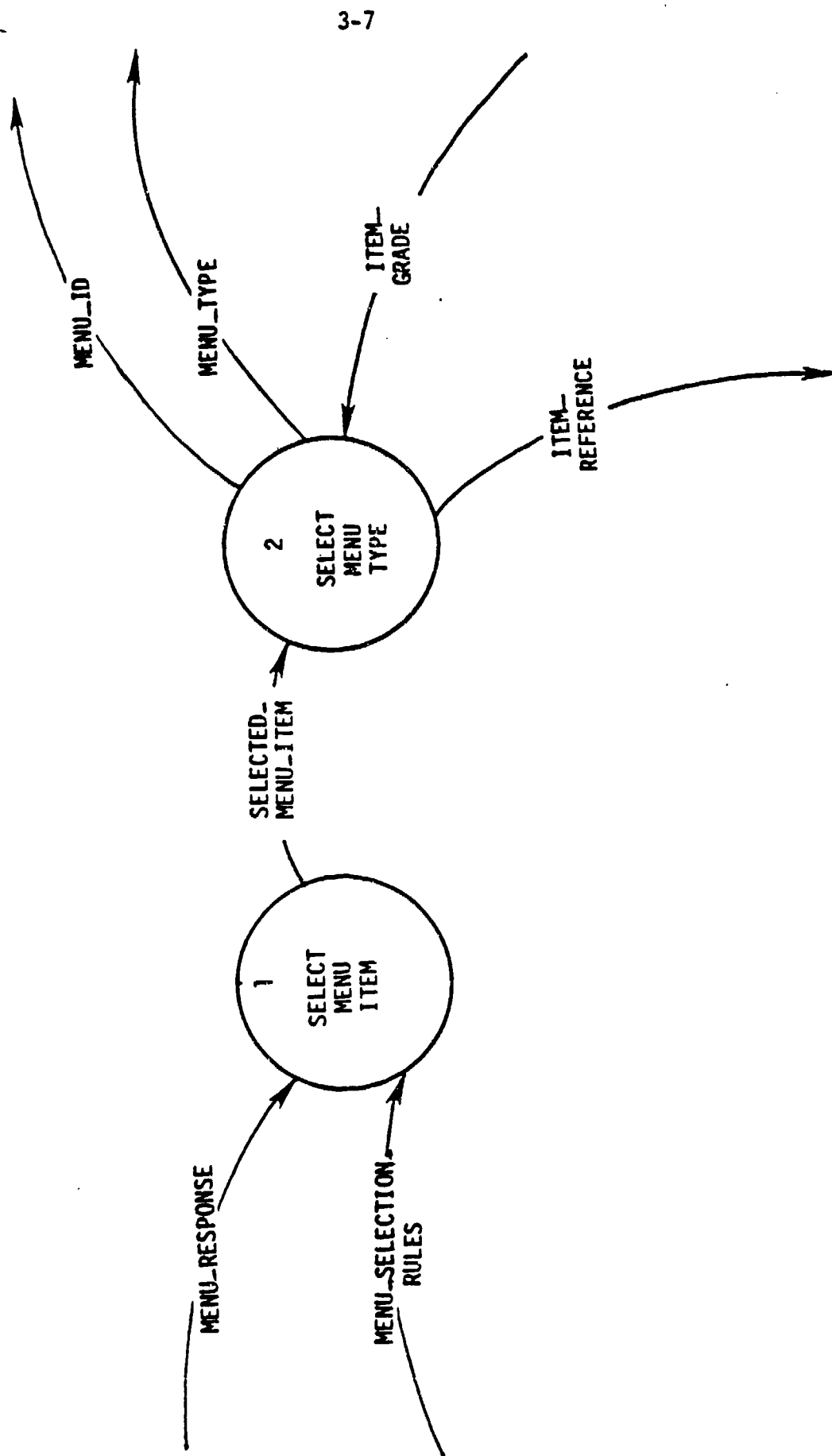
DATA FLOW DIAGRAMS

This section contains the data flow diagrams. Diagrams are ordered lexicographically according to the number assigned to the process explained in a diagram. Each data flow diagram explains a single process and does so in terms of lower level processes (indicated by labeled circles). Each such lower level process is numbered and is in turn defined in one of two ways: either it is "exploded" in a subsequent diagram, or it is a primitive (i.e., lowest level) process requiring no diagram. Each primitive process is described by a mini-spec in section 5. The mini-spec explains how the process generates output flows from input flows. Appendix A contains data flow diagrams of a special, hardware-dependent process, the videodisk flight simulation.

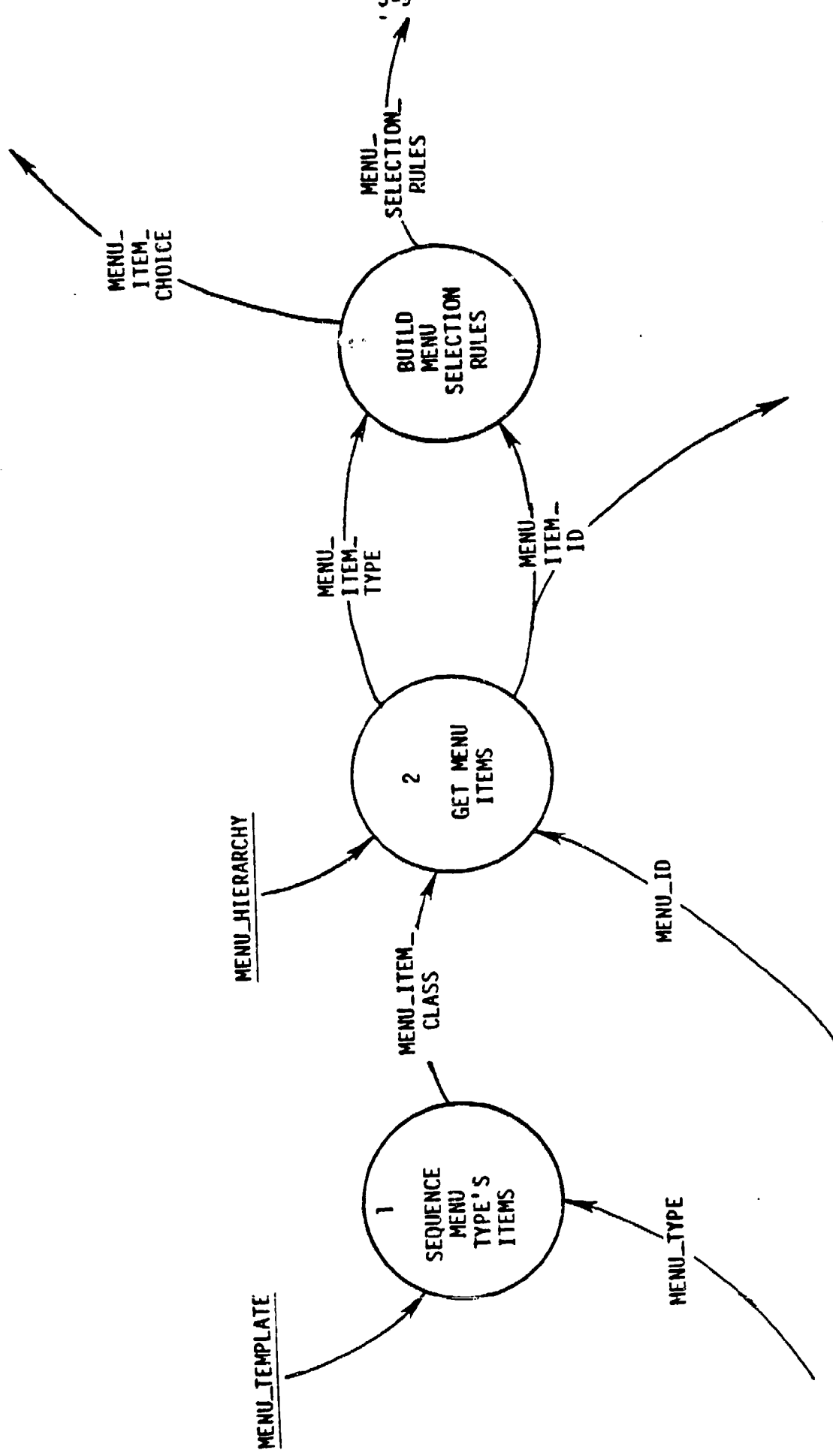
LEVEL 0: FOG-M EMBEDDED TRAINING SYSTEM

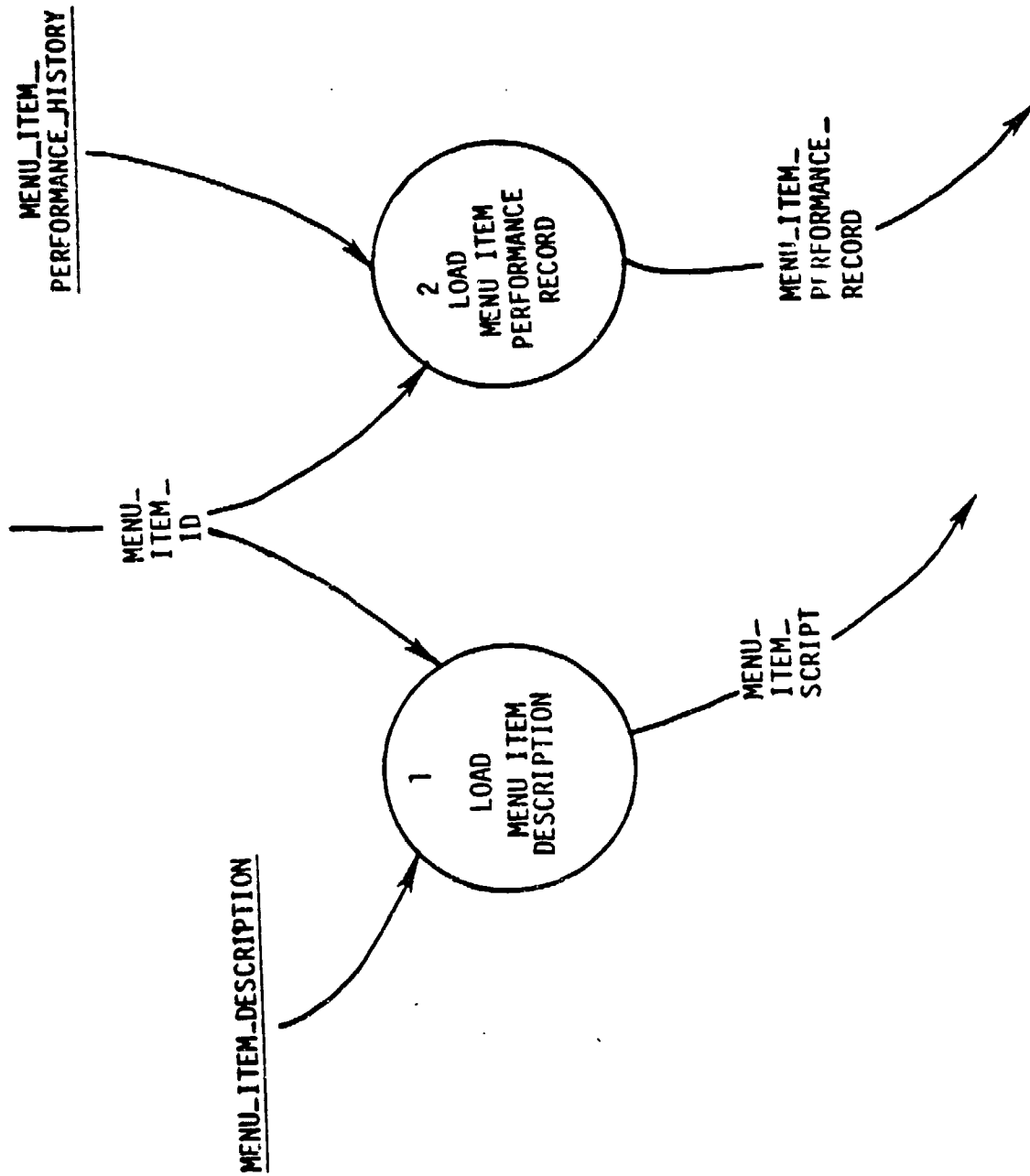
PROCESS 1: SUPERVISE LESSON MENU

PROCESS 1.1: SEQUENCE MENUS

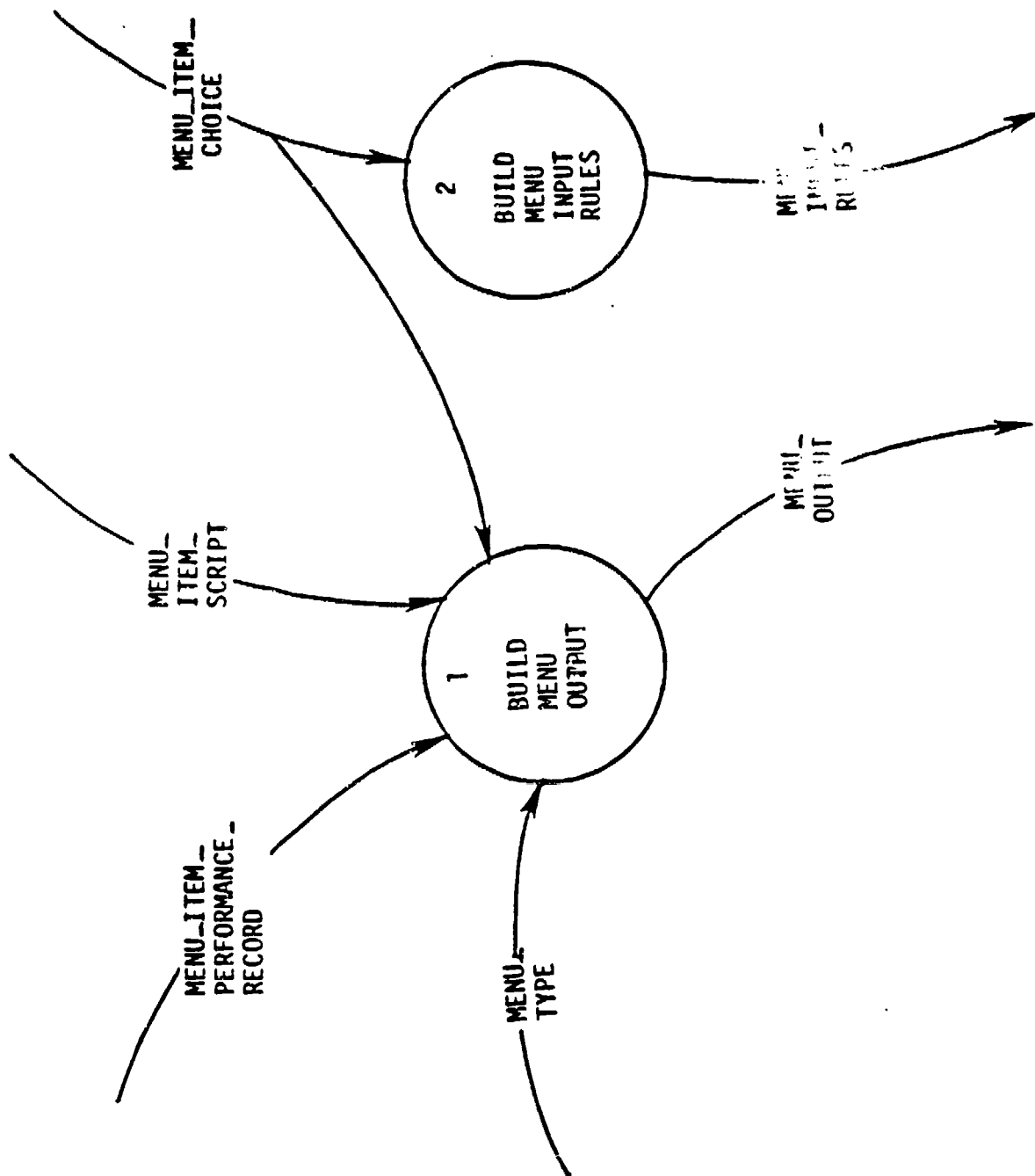


PROCESS 1.2: BUILD MENU

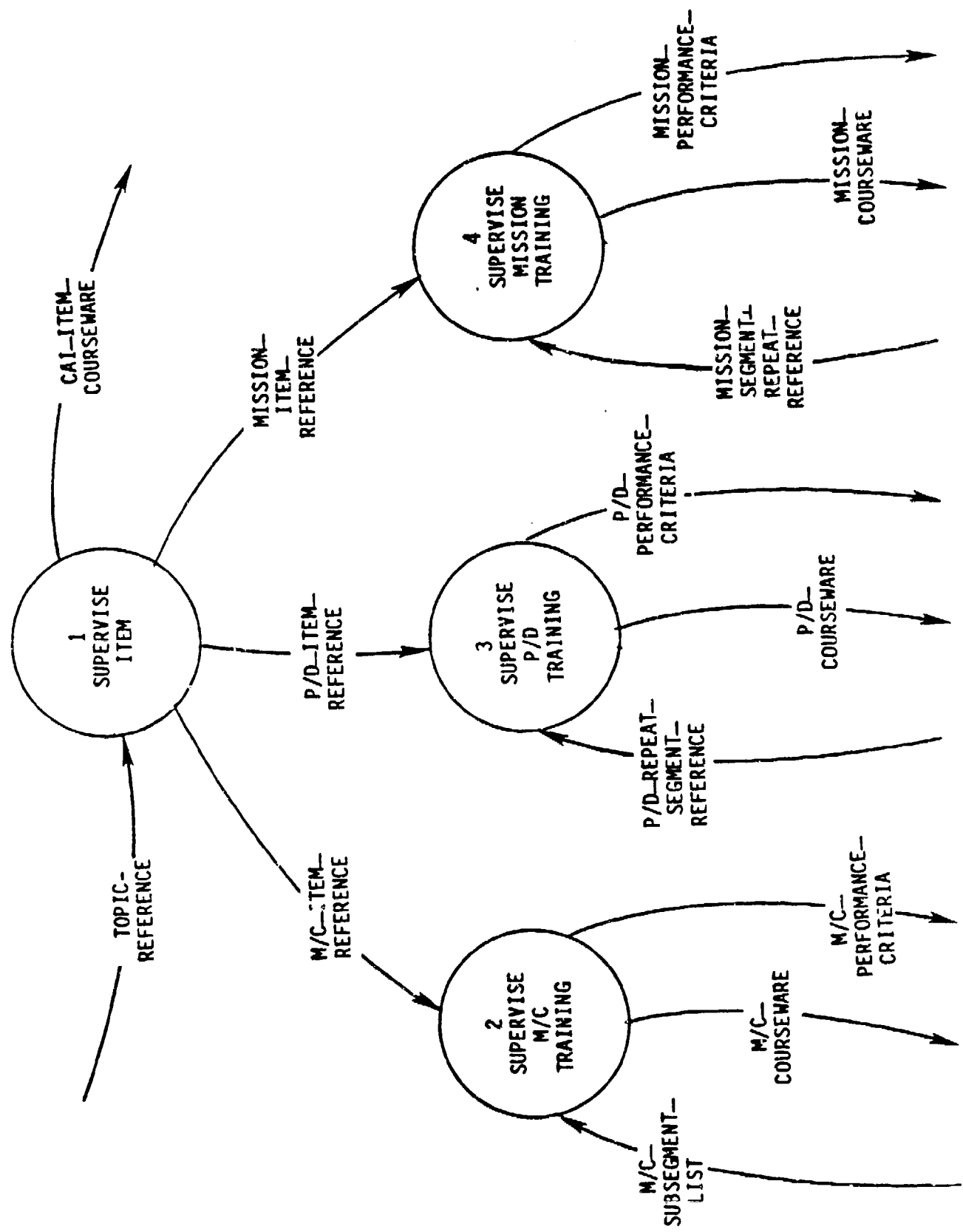


PROCESS 1.3: LOAD MENU ITEM COURSEWARE

PROCESS 1.4: BUILD MENU I/O RULES

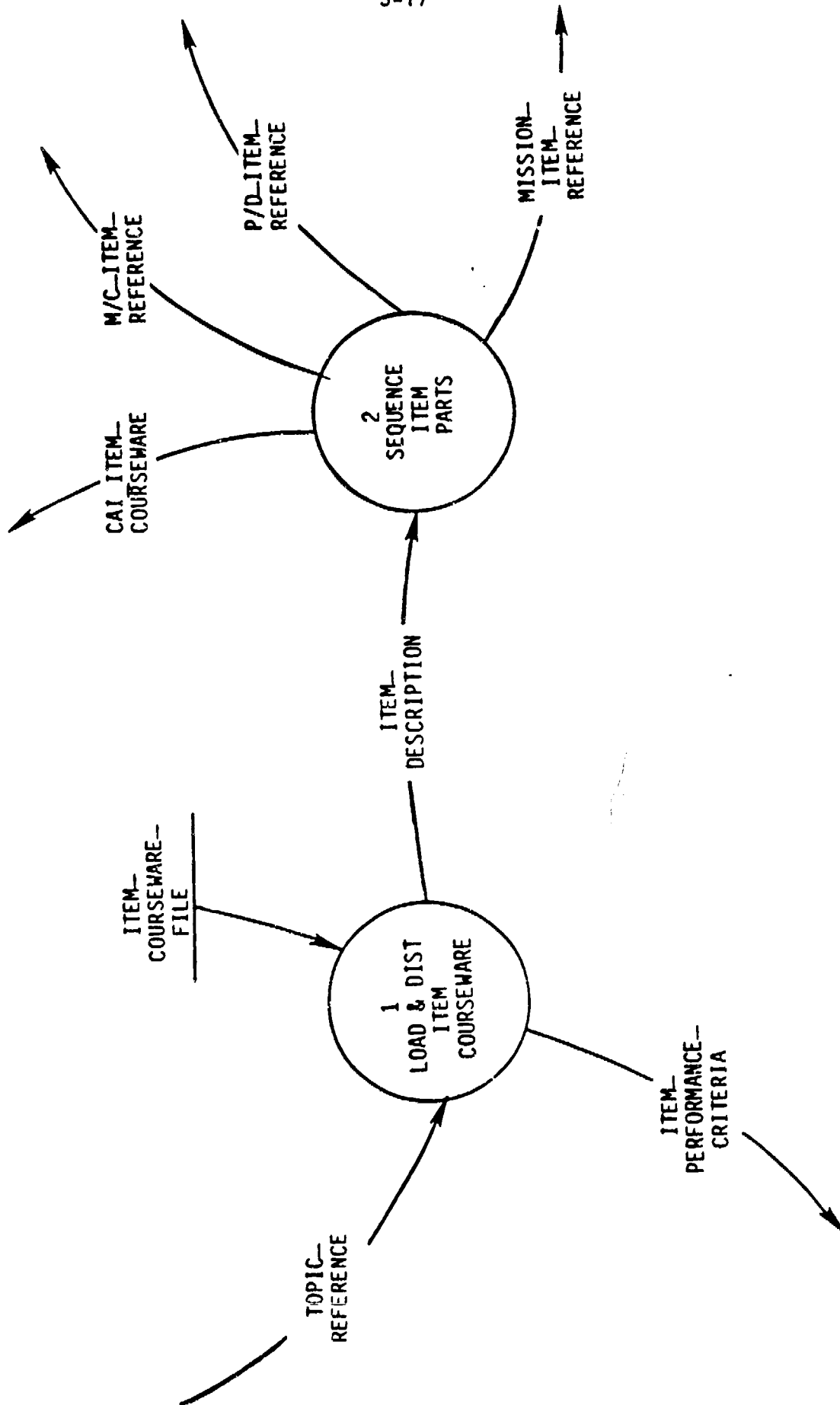


PROCESS 2: SUPERVISE TRAINING

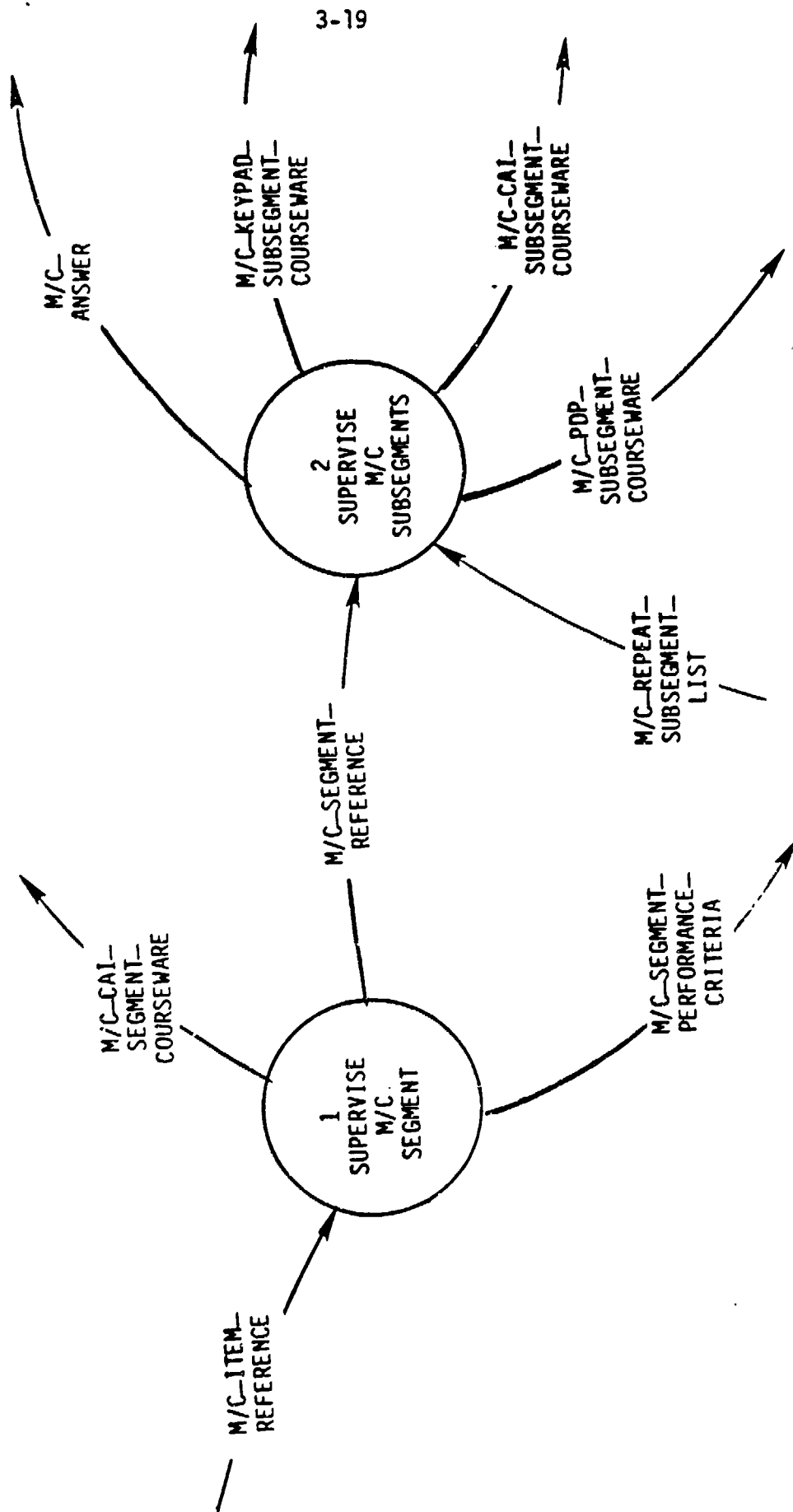


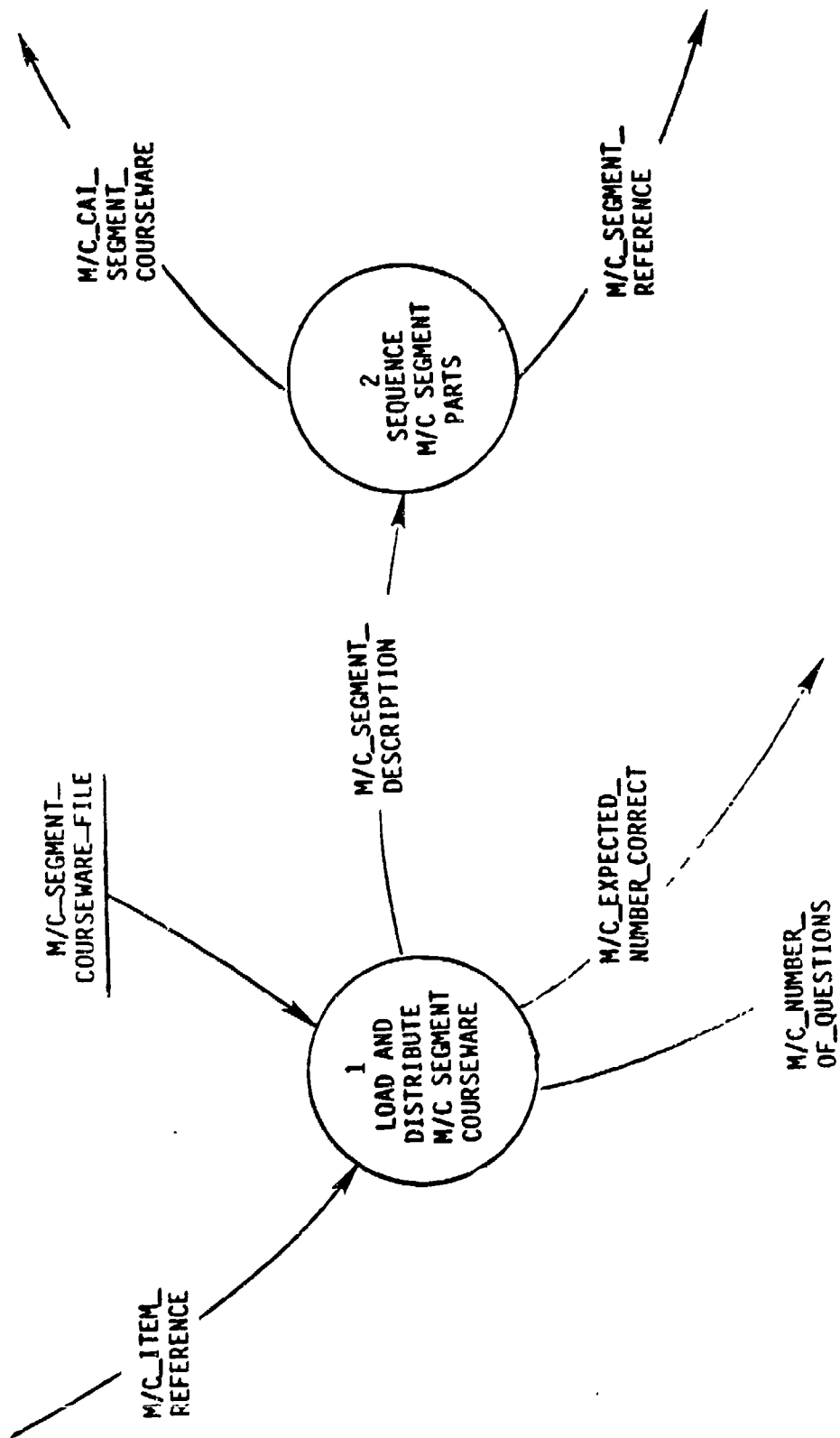
PROCESS 2.1: SUPERVISE ITEM

3-17

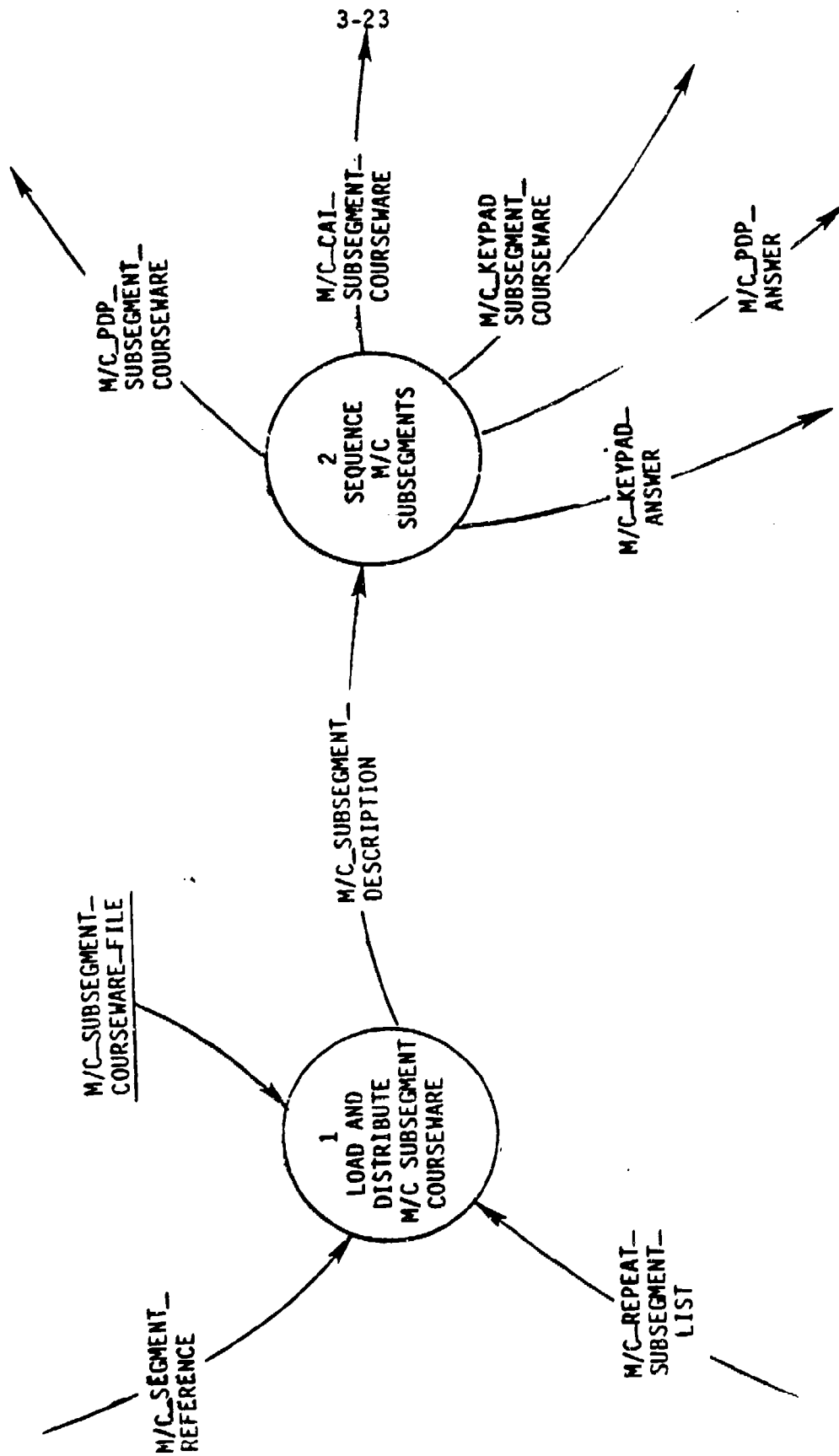


PROCESS 2.2: SUPERVISE M/C TRAINING

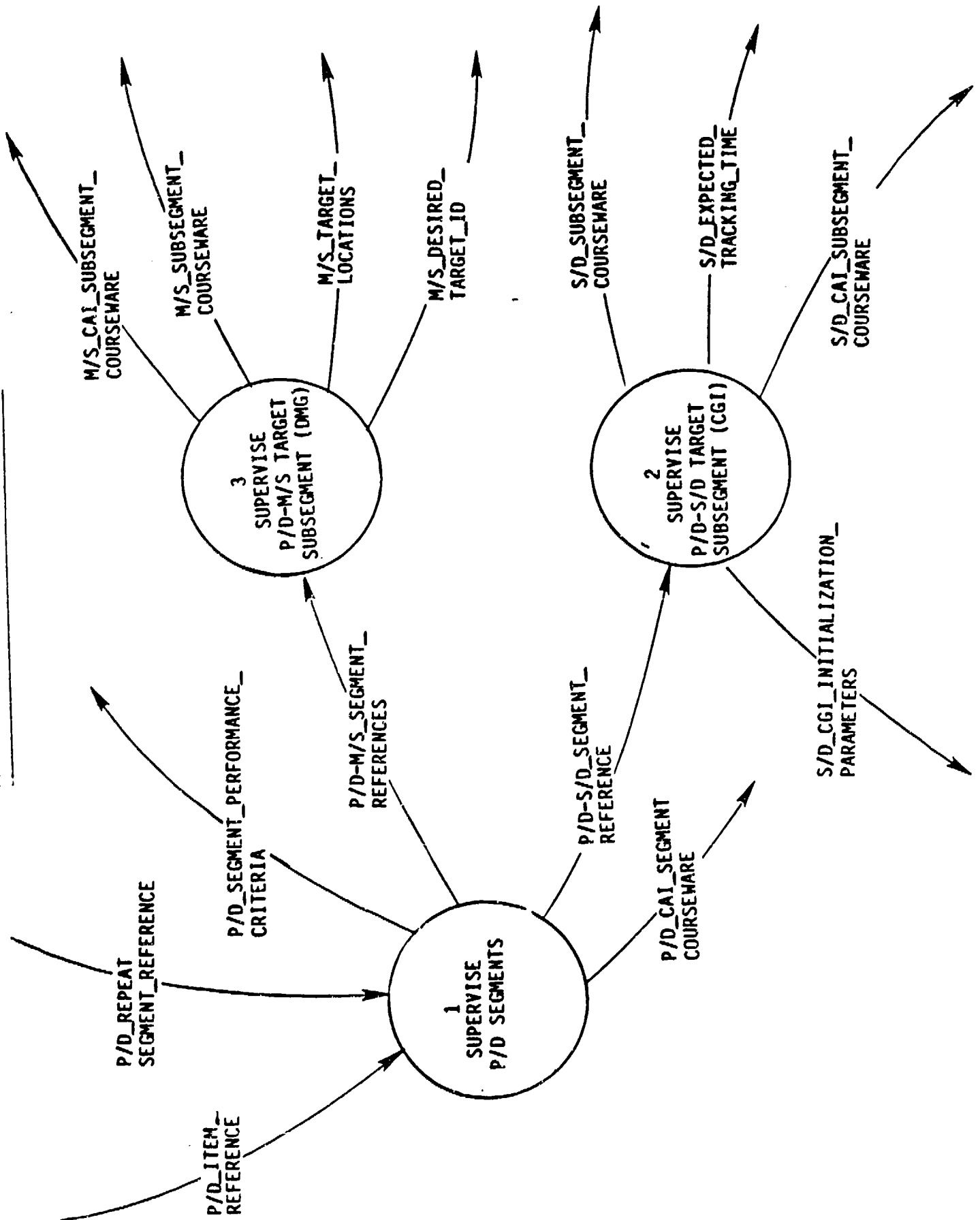


PROCESS 2.2.1: SUPERVISE M/C SEGMENT

PROCESS 2.2.2: SUPERVISE M/C SUBSEGMENT

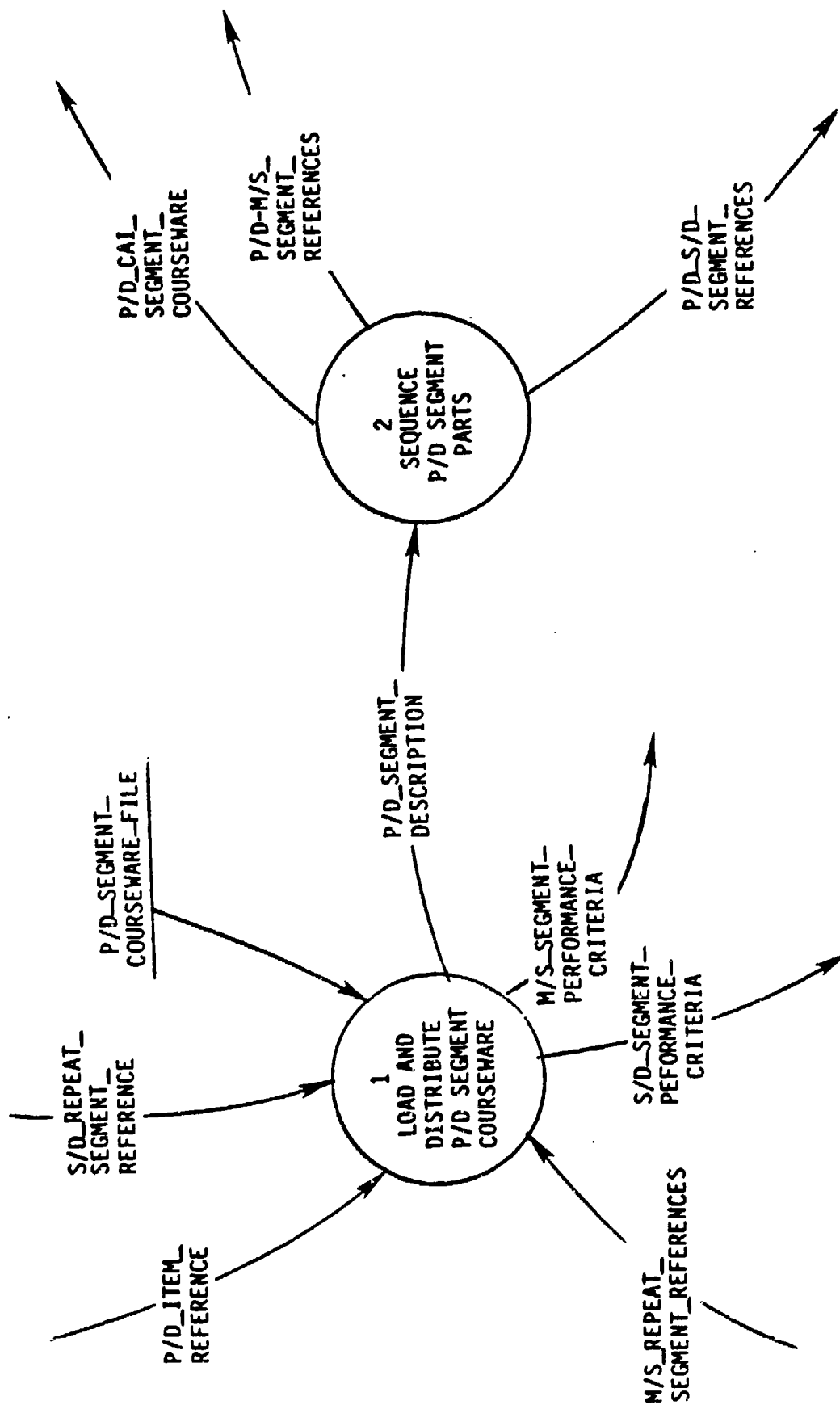


PROCESS 2.3: SUPERVISE P/D TRAINING

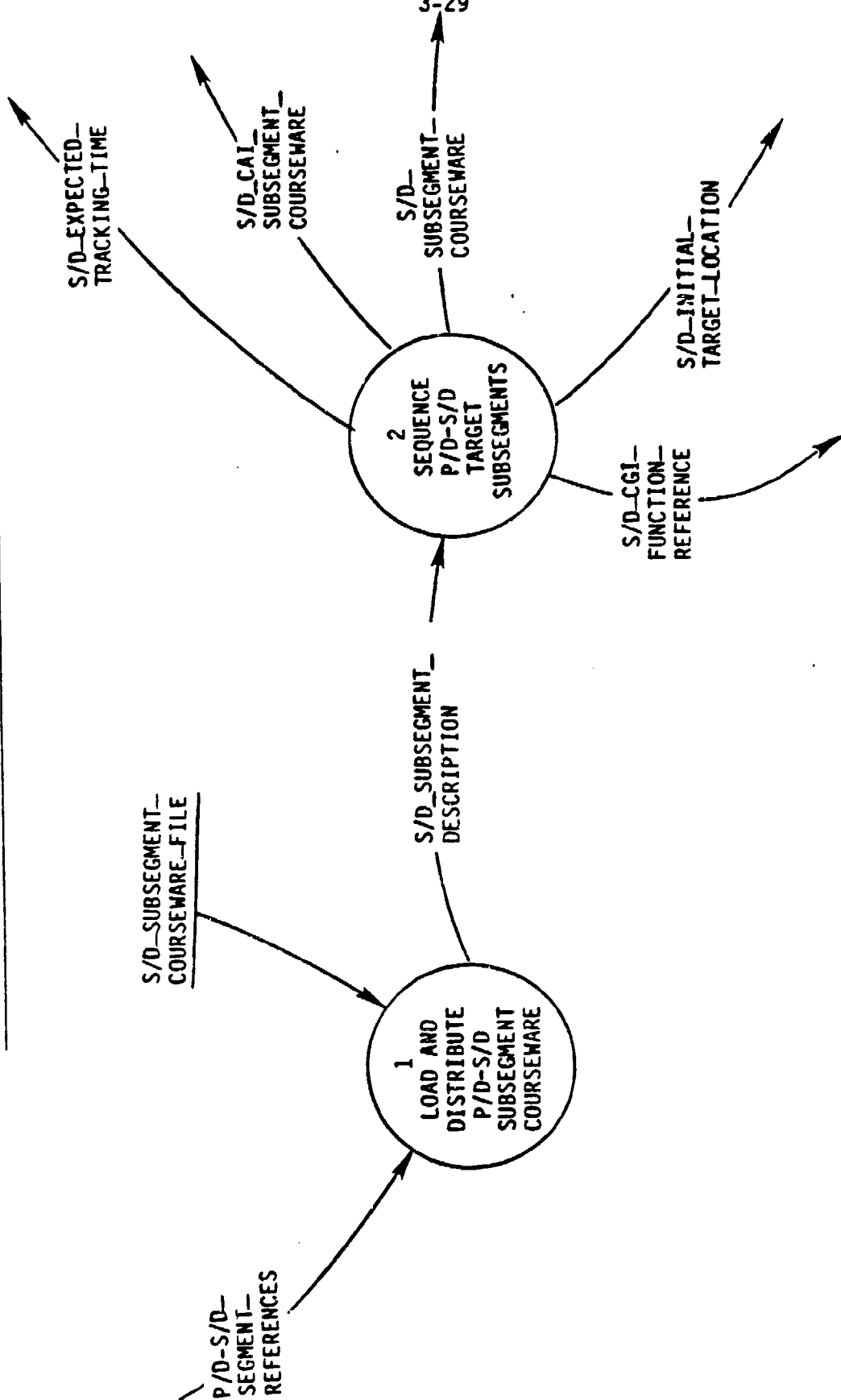


PROCESS 2.3.1: SUPERVISE P/D SEGMENT

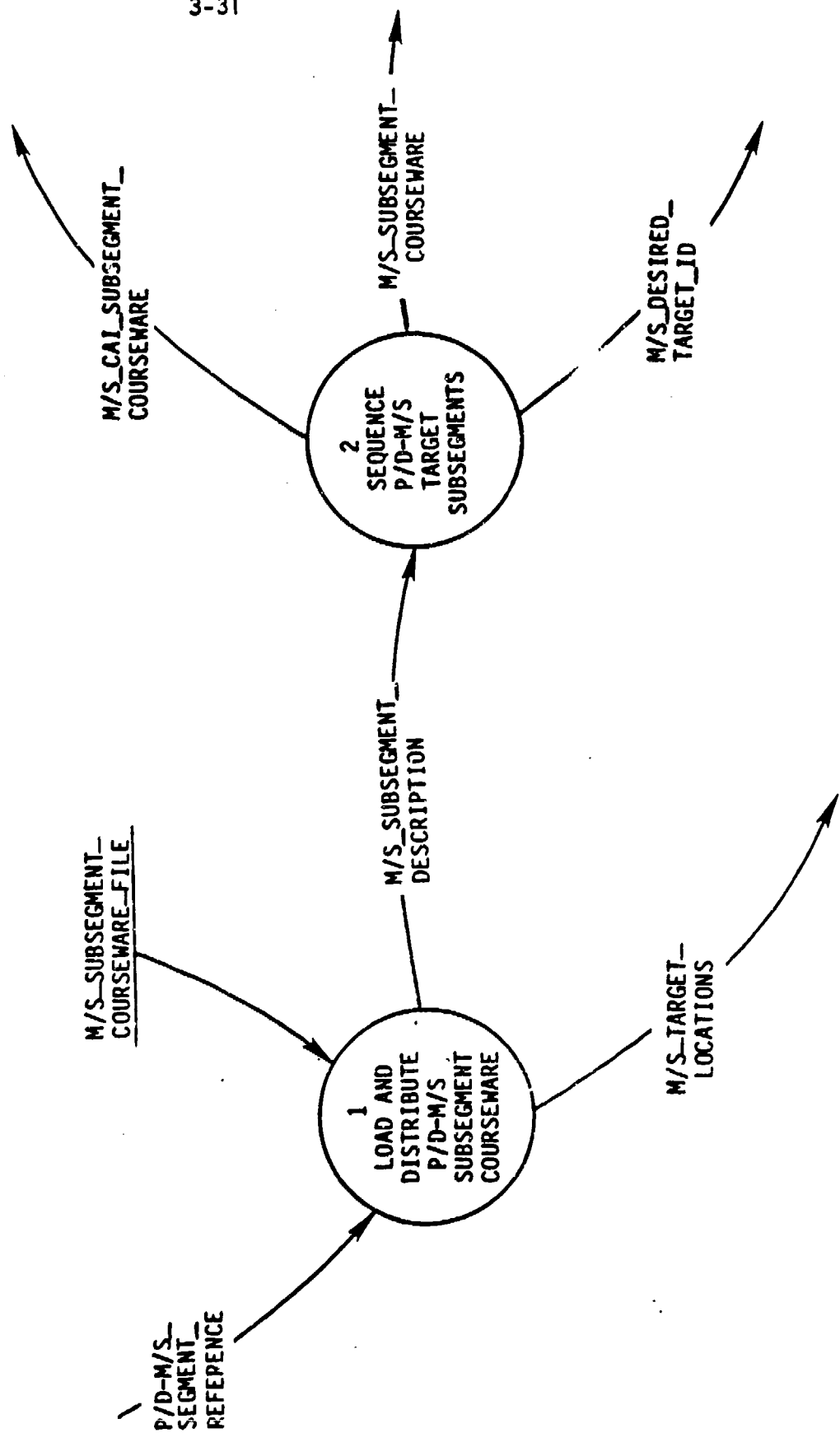
3-27



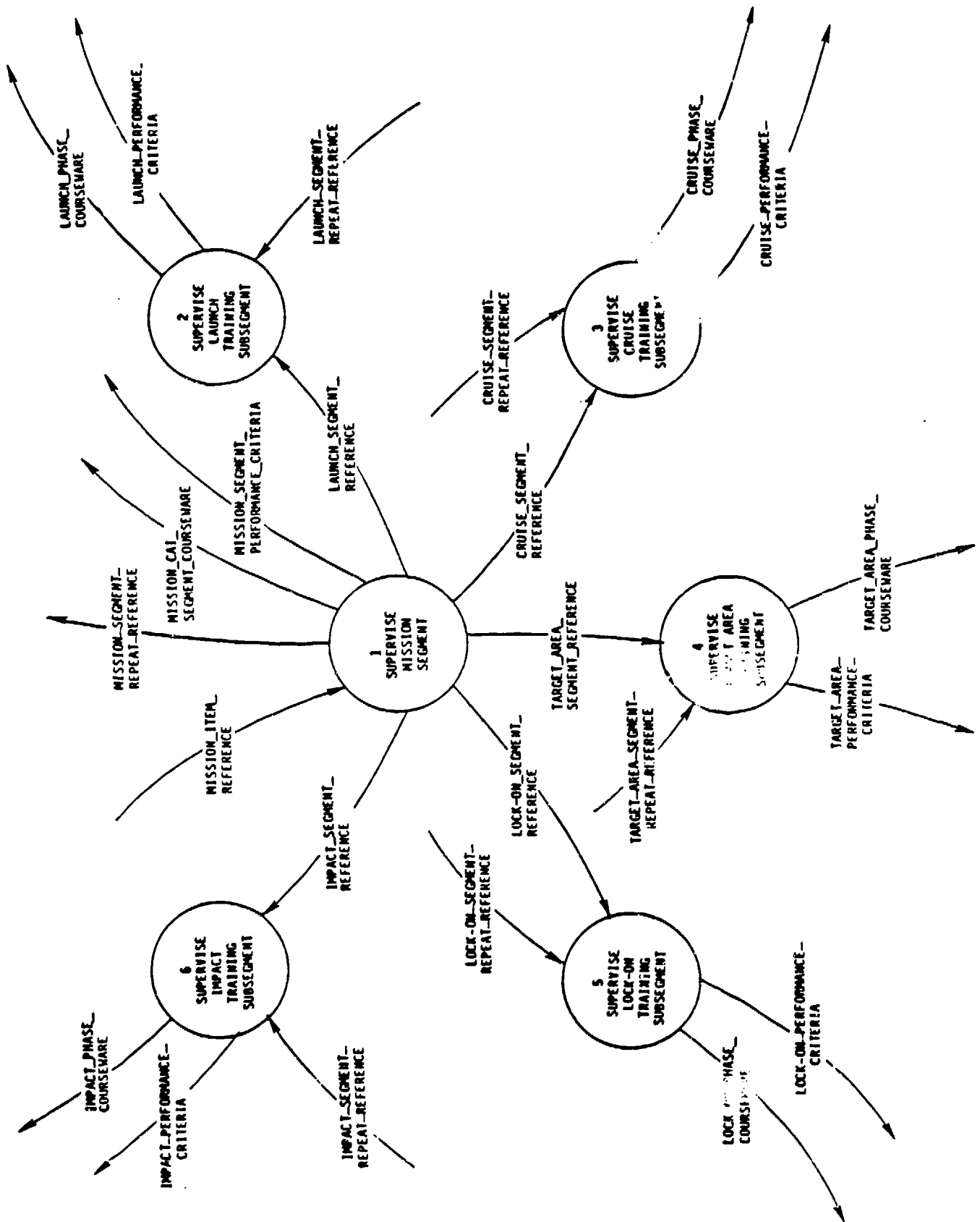
PROCESS 2.3.2: SUPERVISE P/D-S/D TARGET SUBSEGMENT



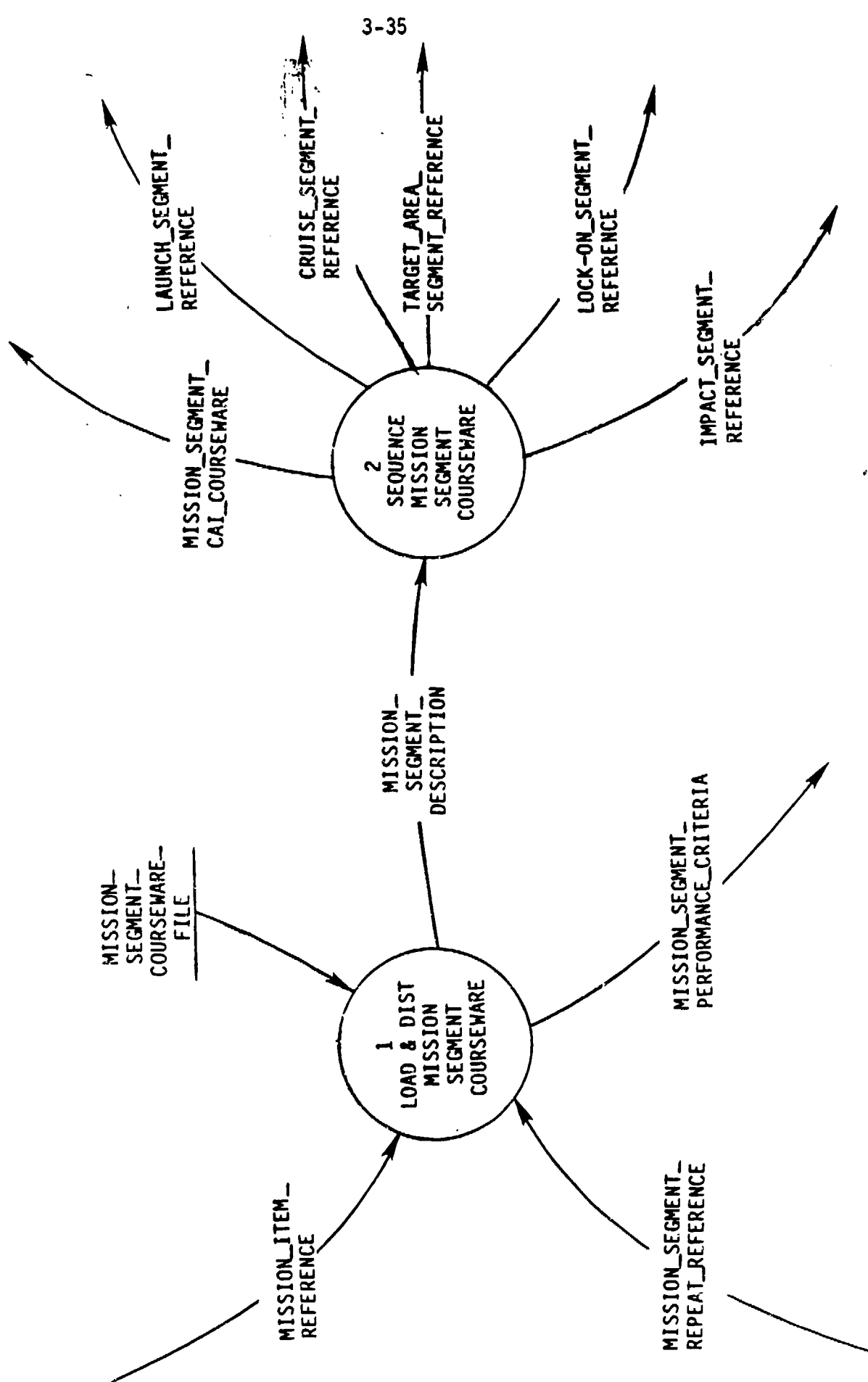
PROCESS 2.3.3: SUPERVISE P/D-M/S TARGET SUBSEGMENT



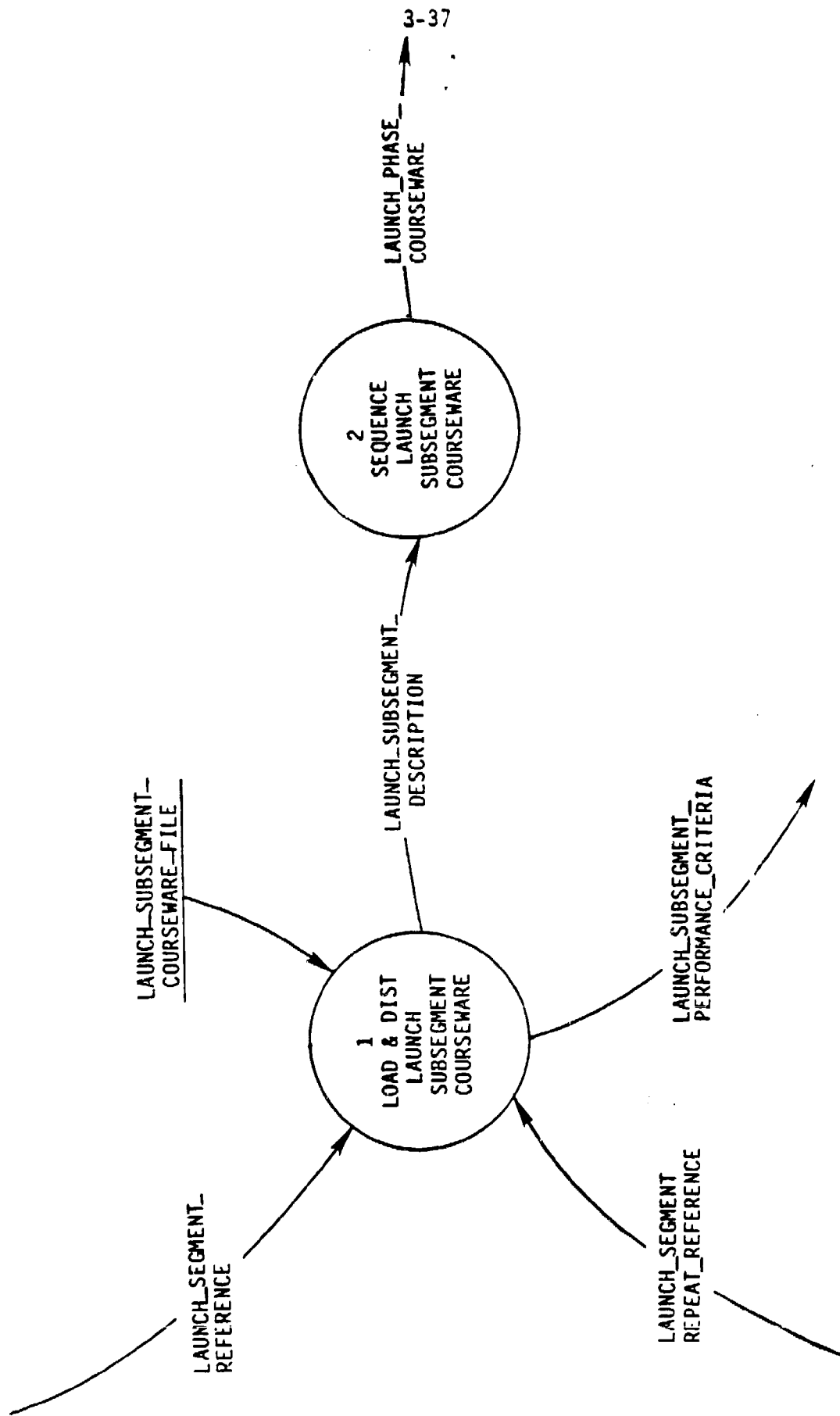
PROCESS 2.4: SUPERVISE MISSION TRAINING

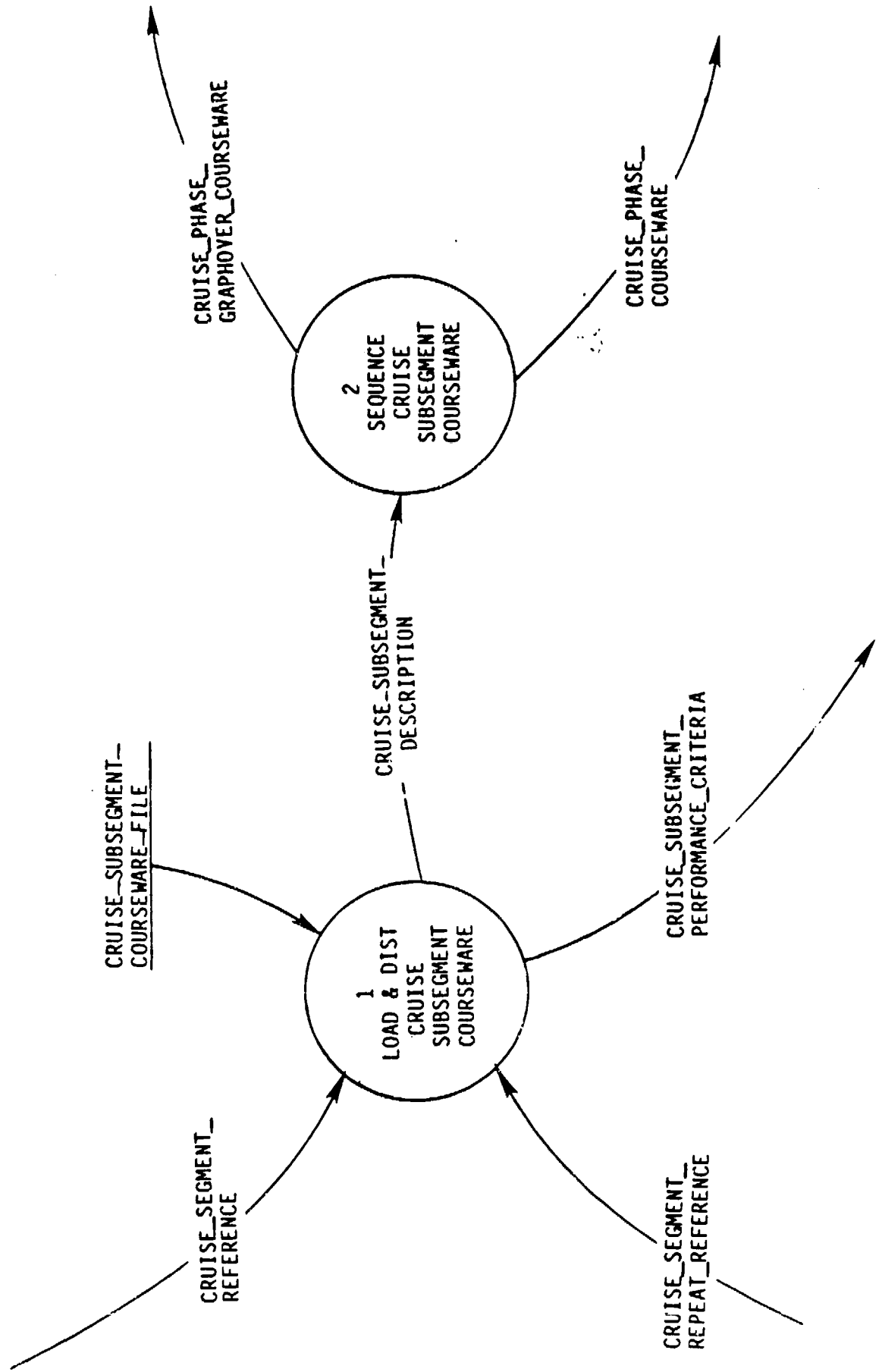


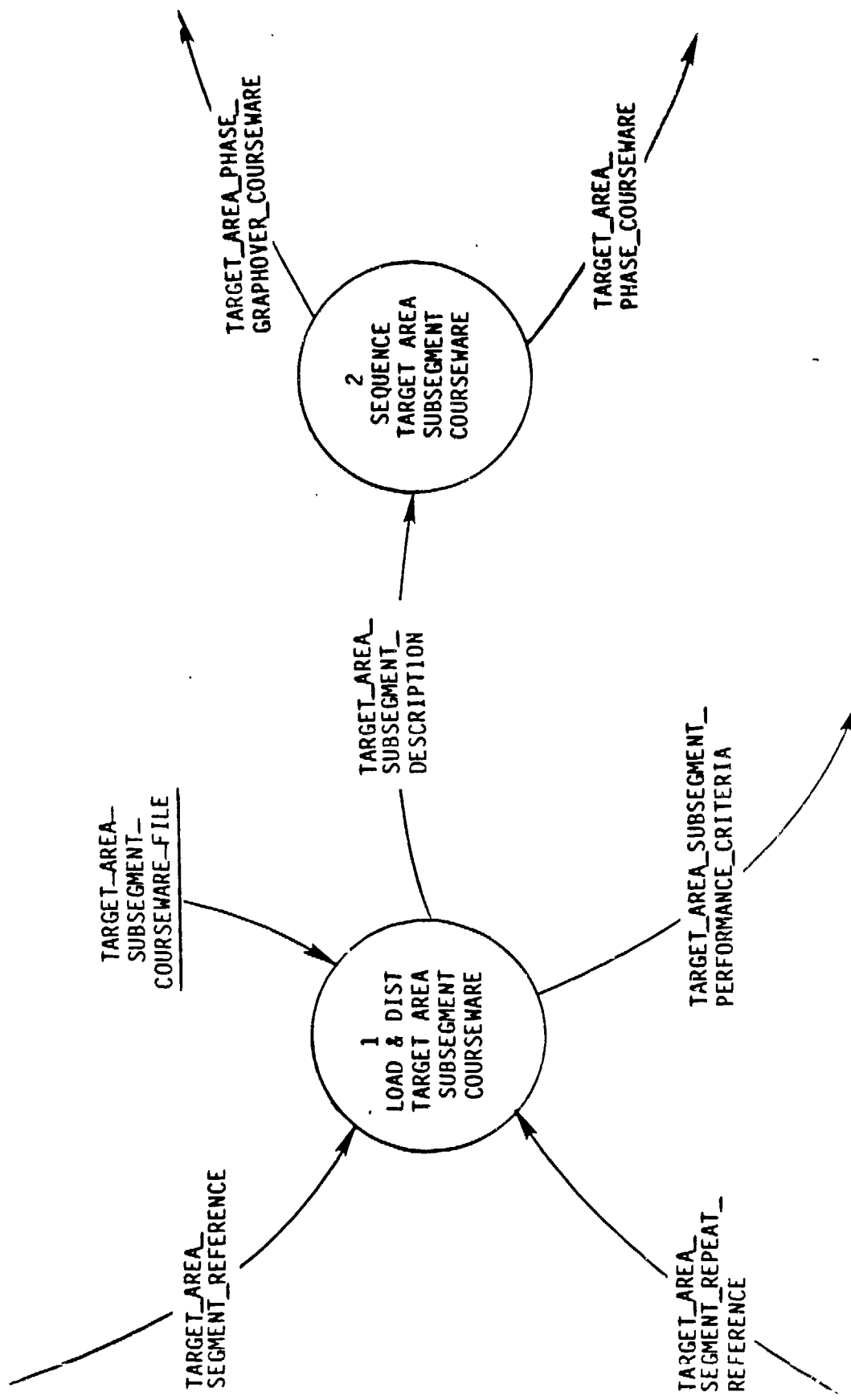
PROCESS 2.4.1: SUPERVISE MISSION SEGMENTS



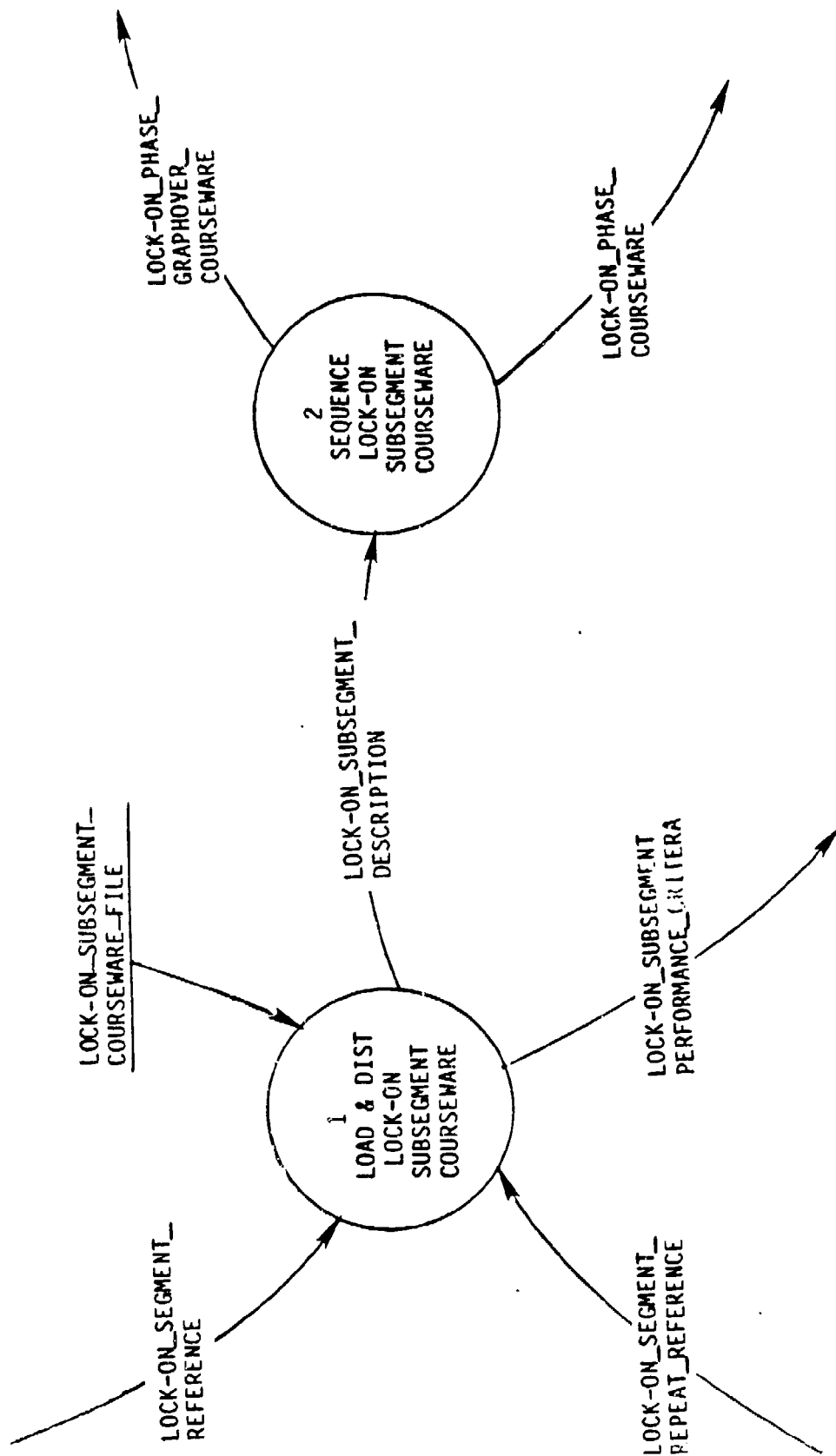
PROCESS 2.4.2: SUPERVISE LAUNCH TRAINING SEGMENT



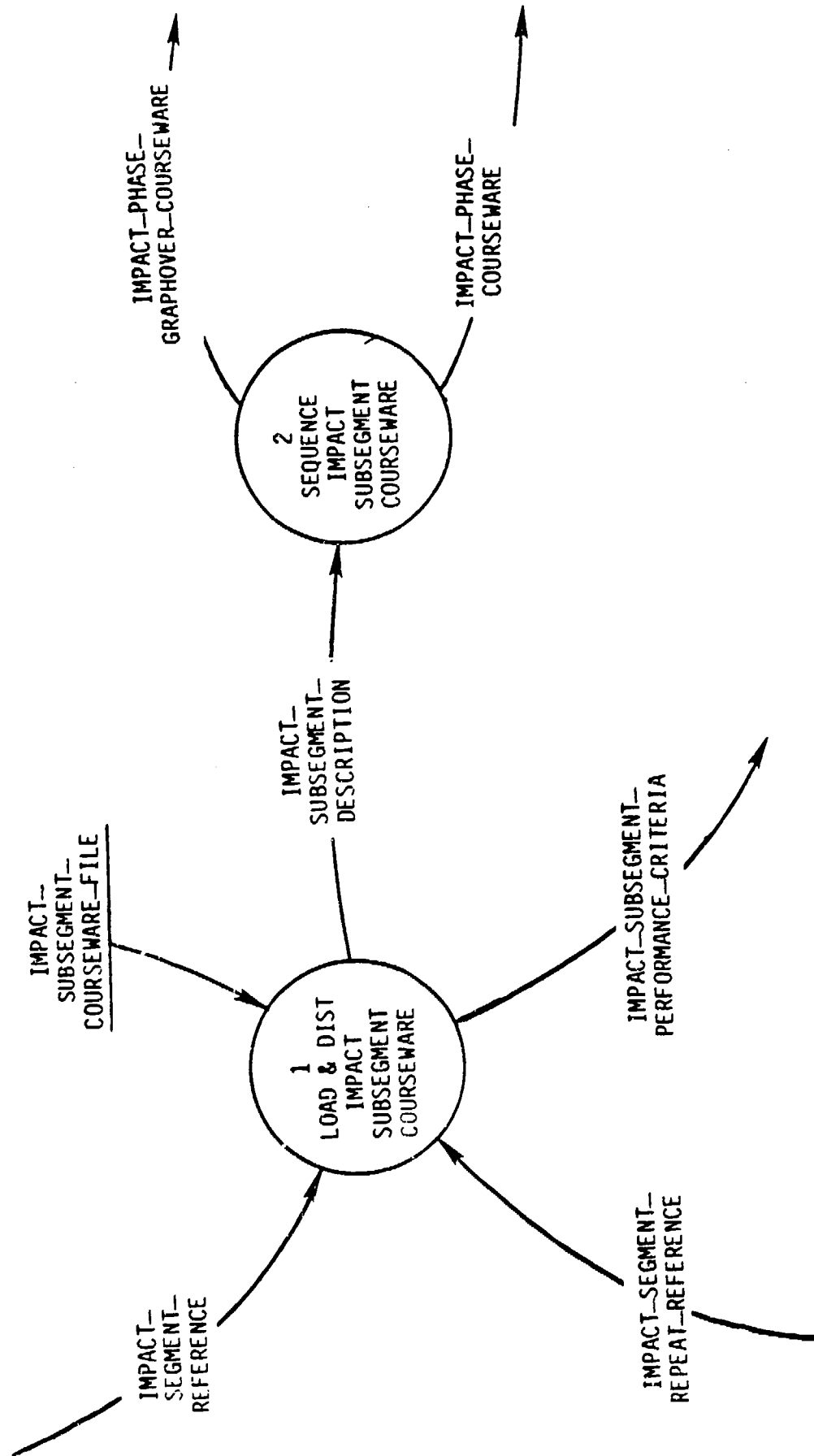
PROCESS 2.4.3: SUPERVISE CRUISE TRAINING SEGMENT



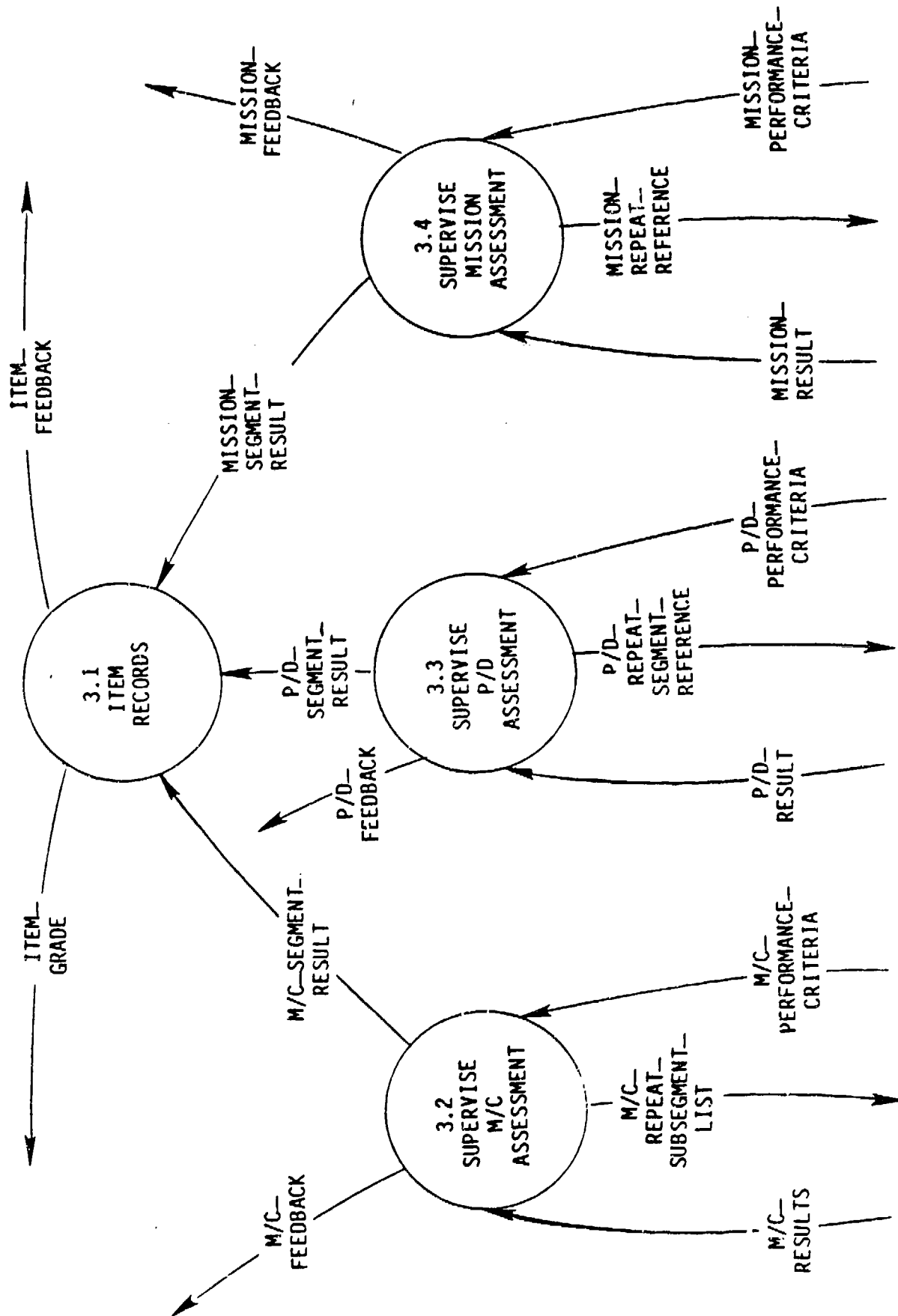
PROCESS 2.4.5: SUPERVISE LOCK-ON TRAINING SEGMENT

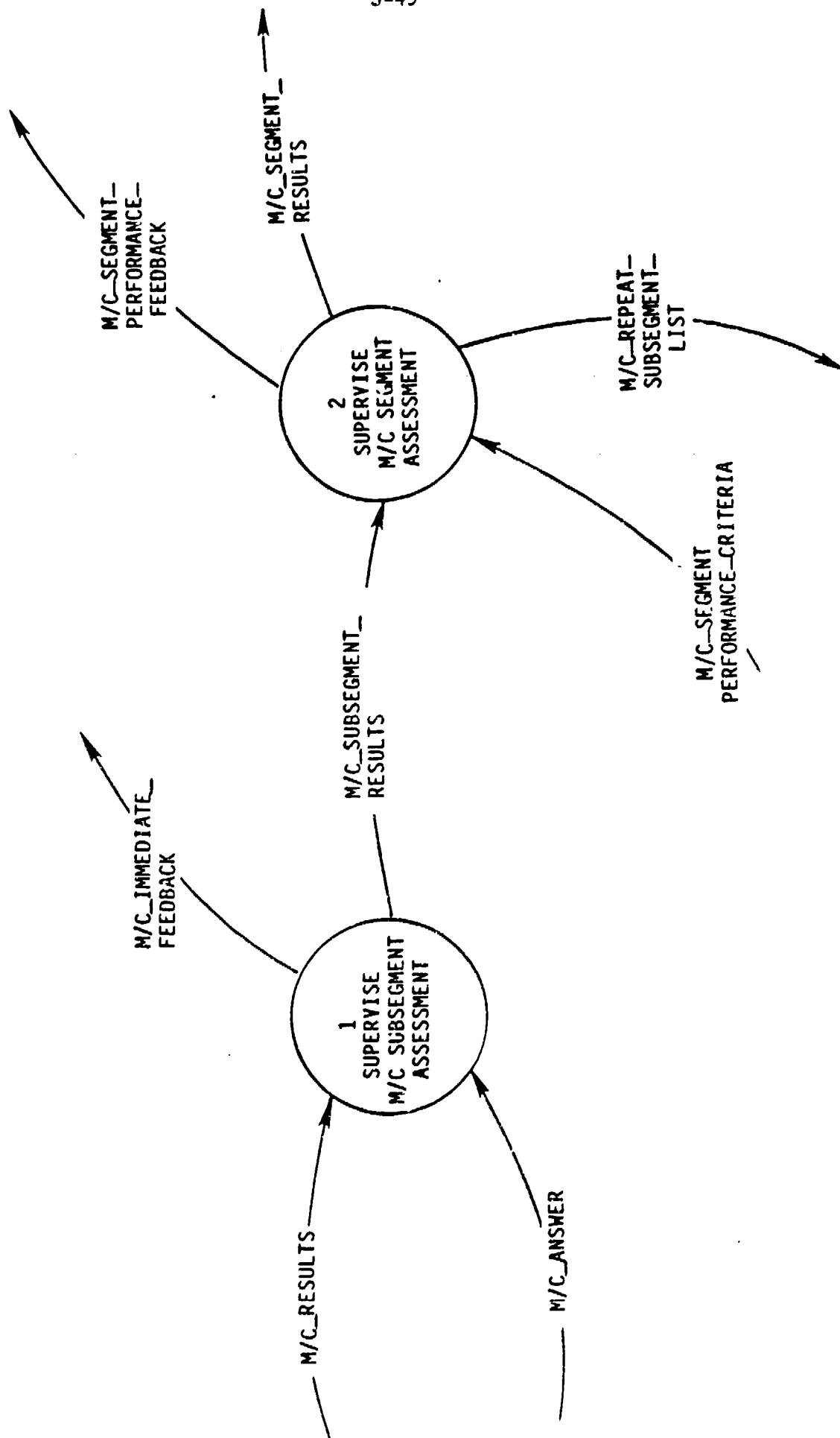


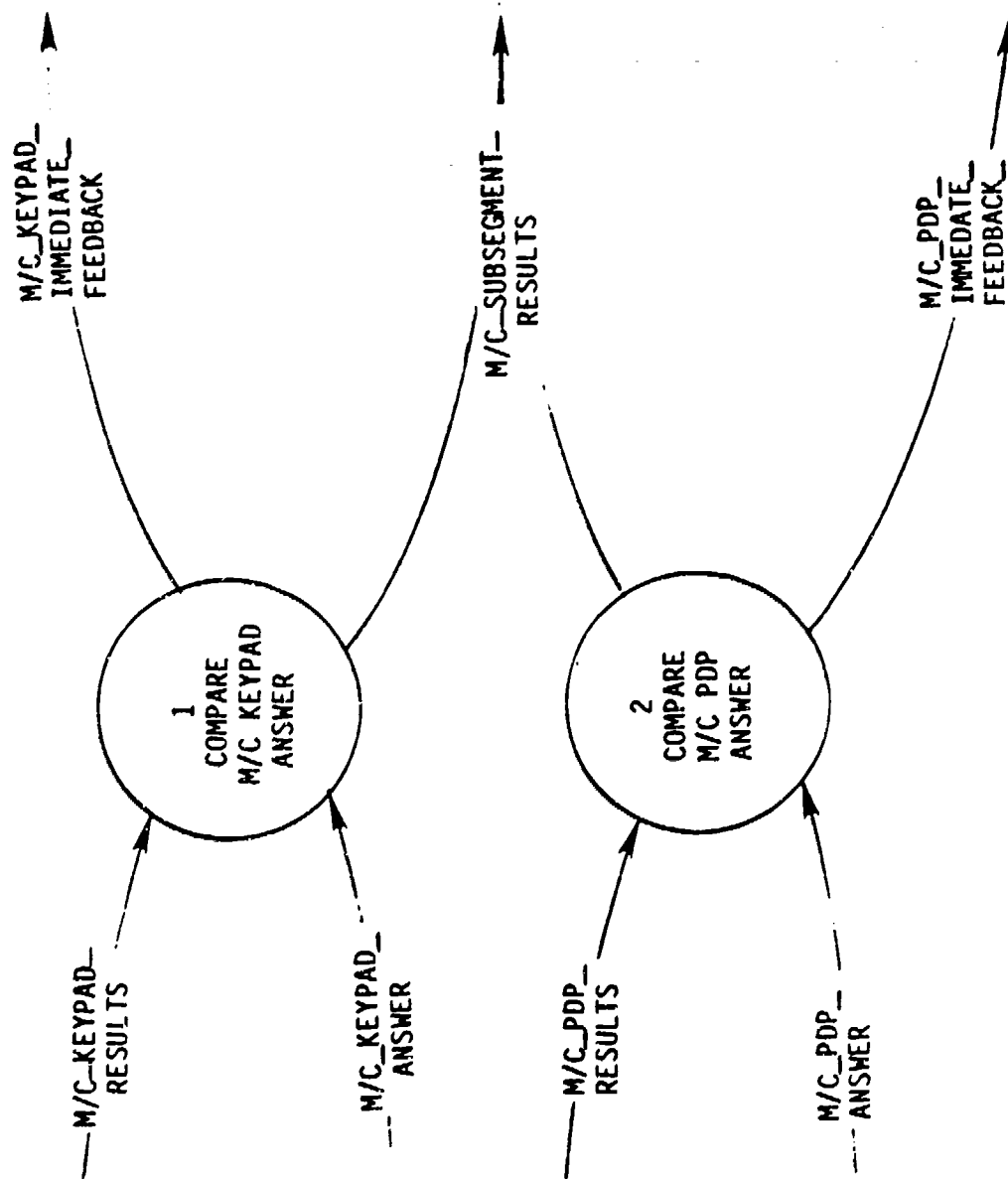
PROCESS 2.4.6: SUPERVISE IMPACT TRAINING SEGMENT



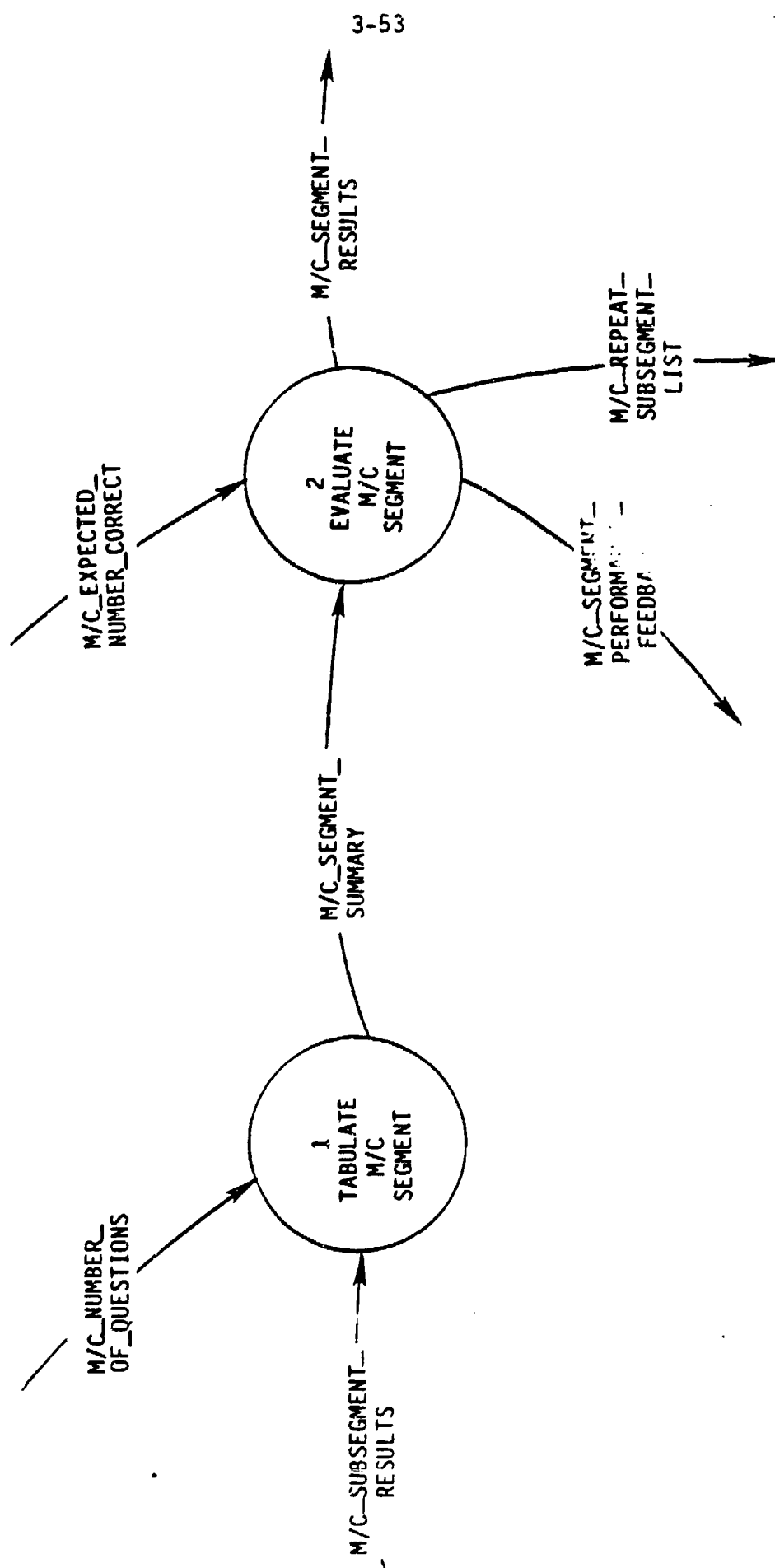
PROCESS 3: SUPERVISE ASSESSMENT



PROCESS 3.2: SUPERVISE M/C ASSESSMENT

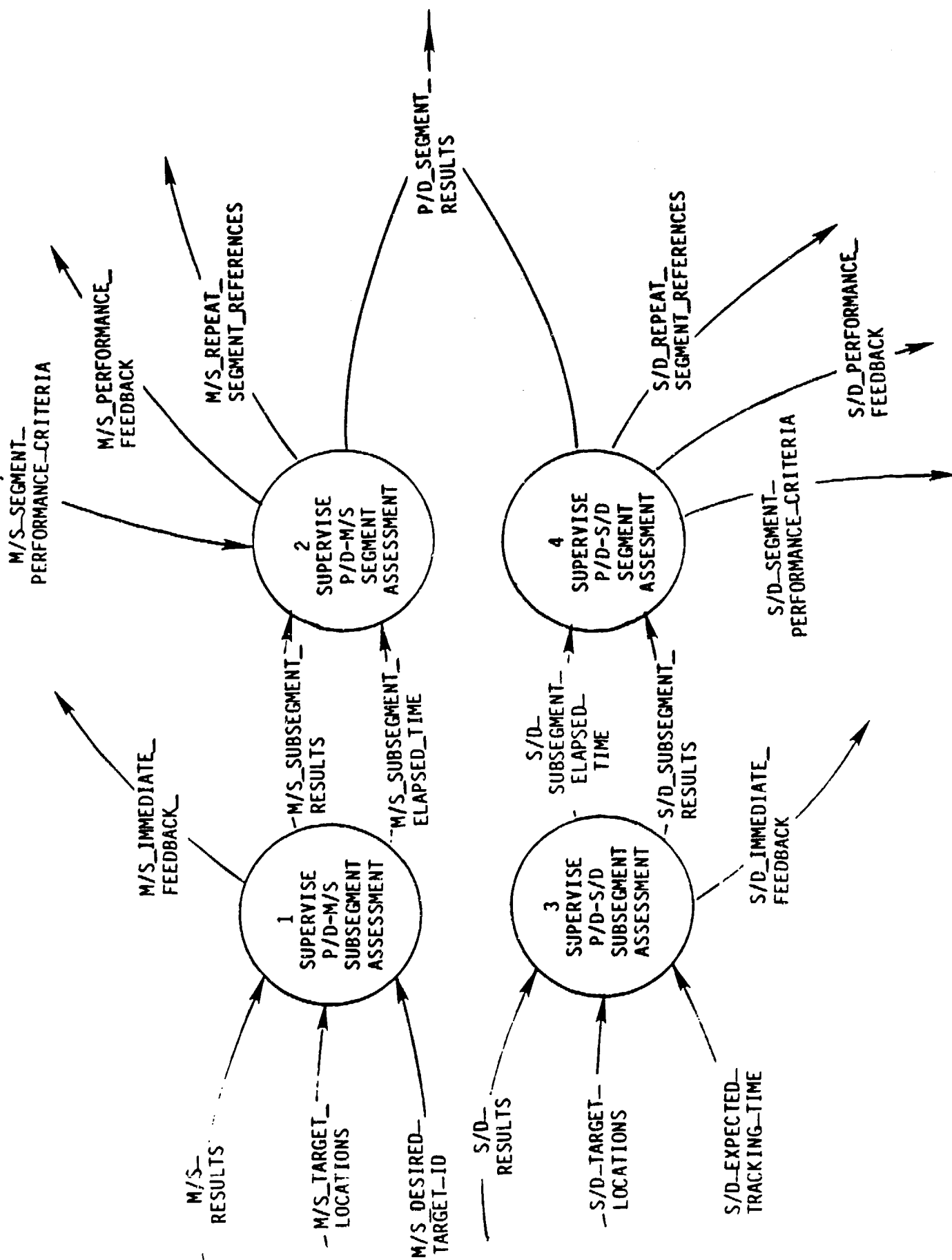
PROCESS 3.2.1: SUPERVISE M/C SUBSEGMENT ASSESSMENT

PROCESS 3.2.2: SUPERVISE M/C SEGMENT ASSESSMENT

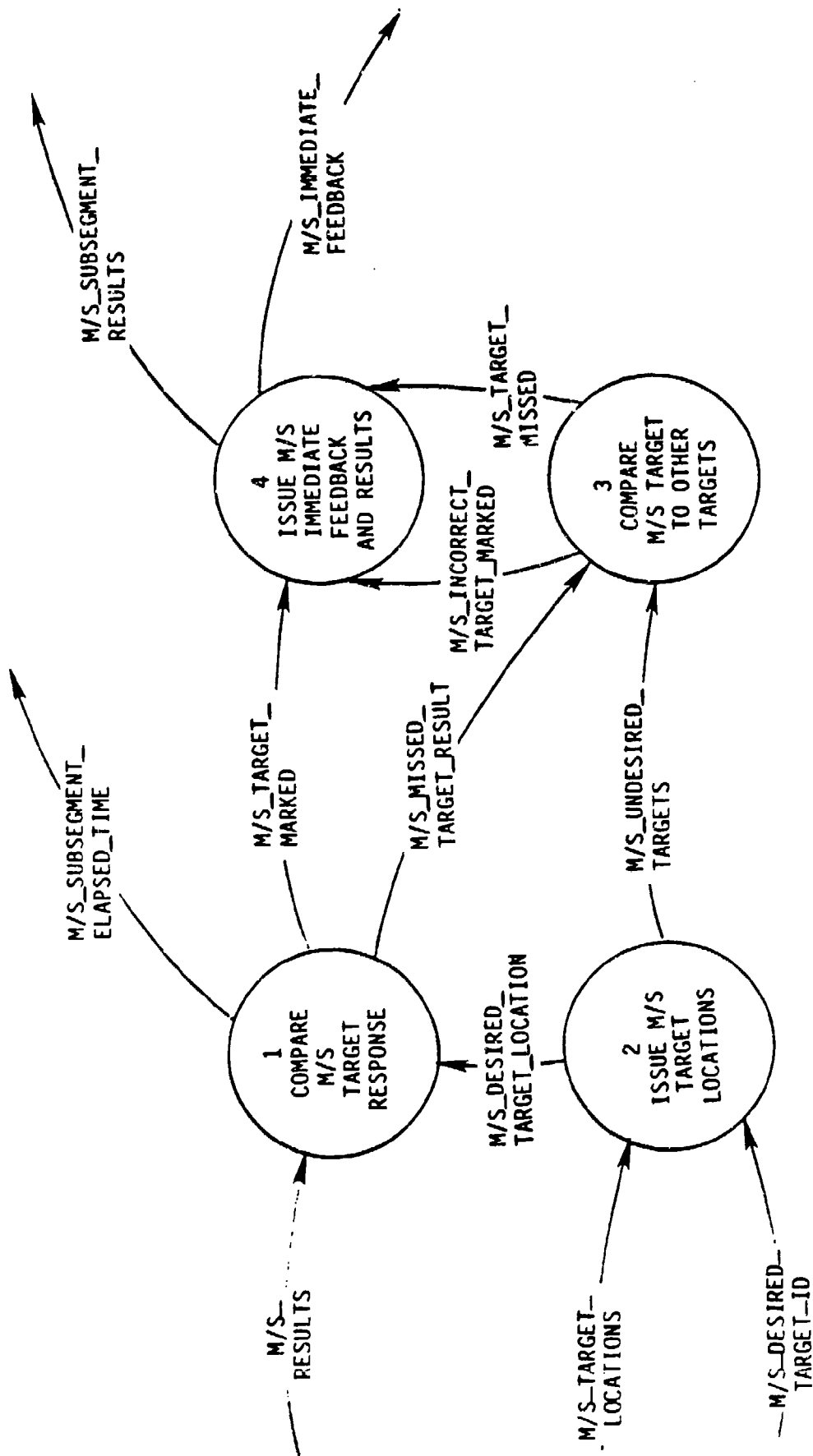


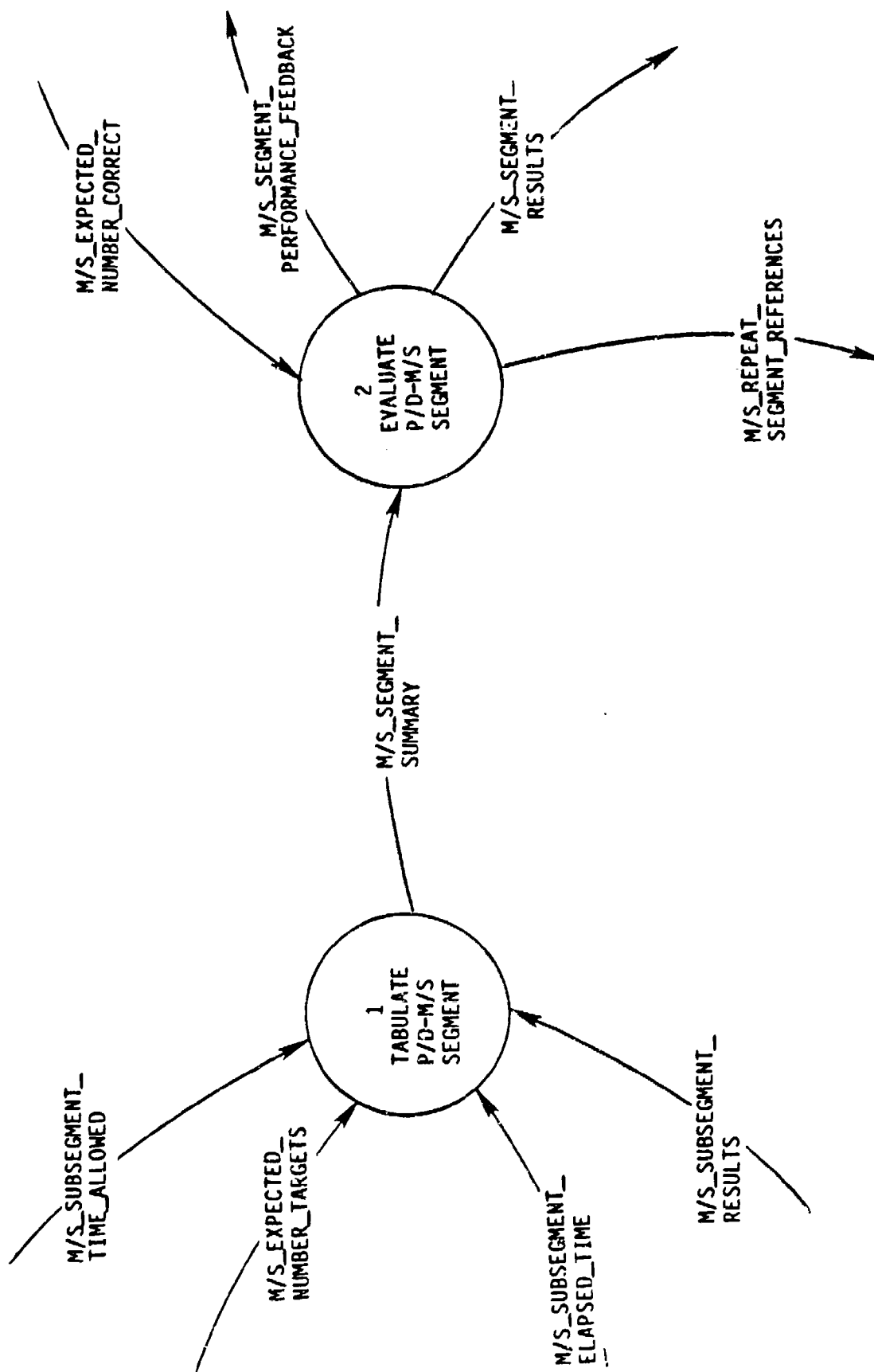
PROCESS 3.3: SUPERVISE P/D ASSESSMENT

3-55

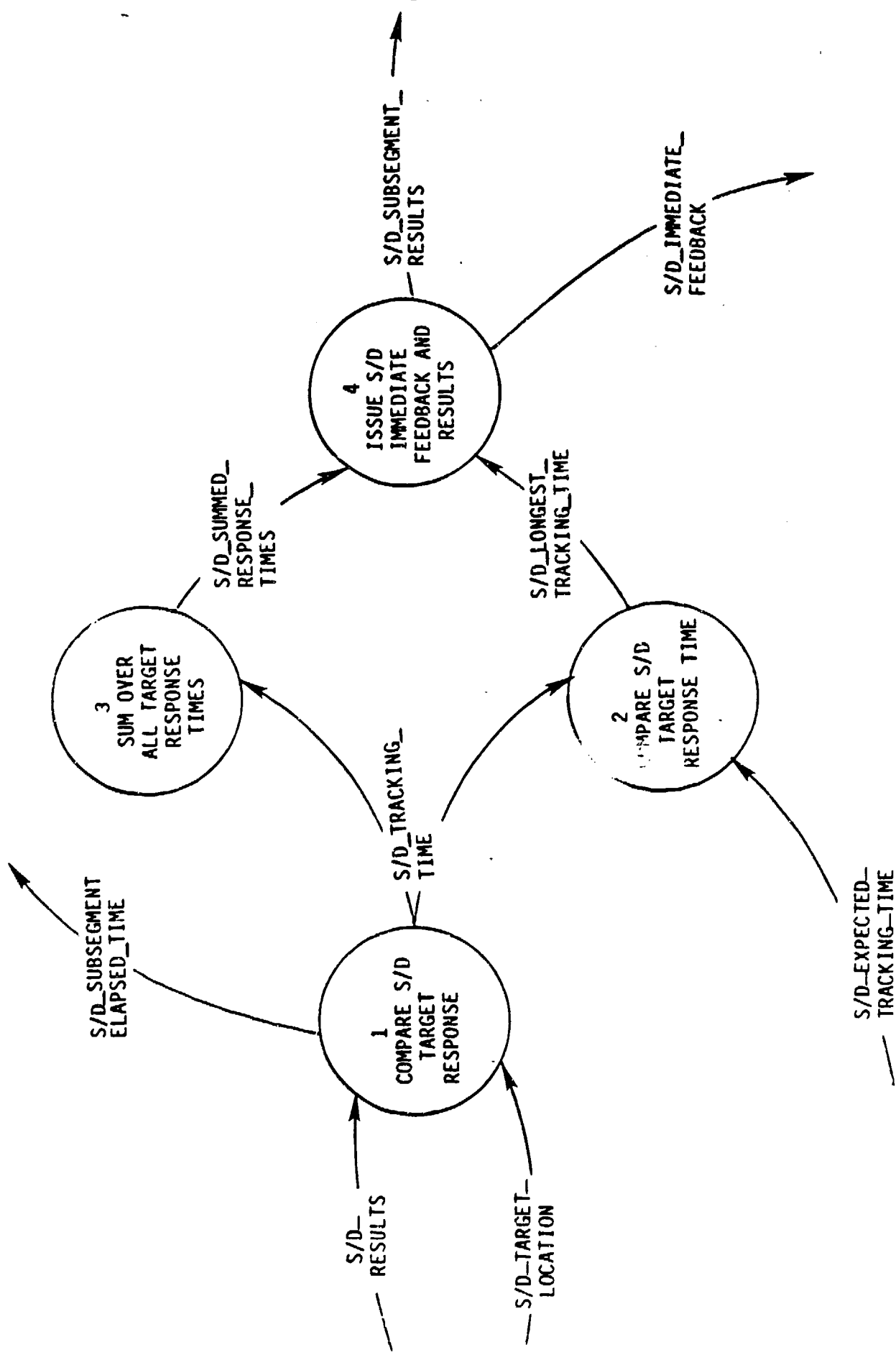


PROCESS 3.3.1: P/D-M/S SUBSEGMENT ASSESSMENT

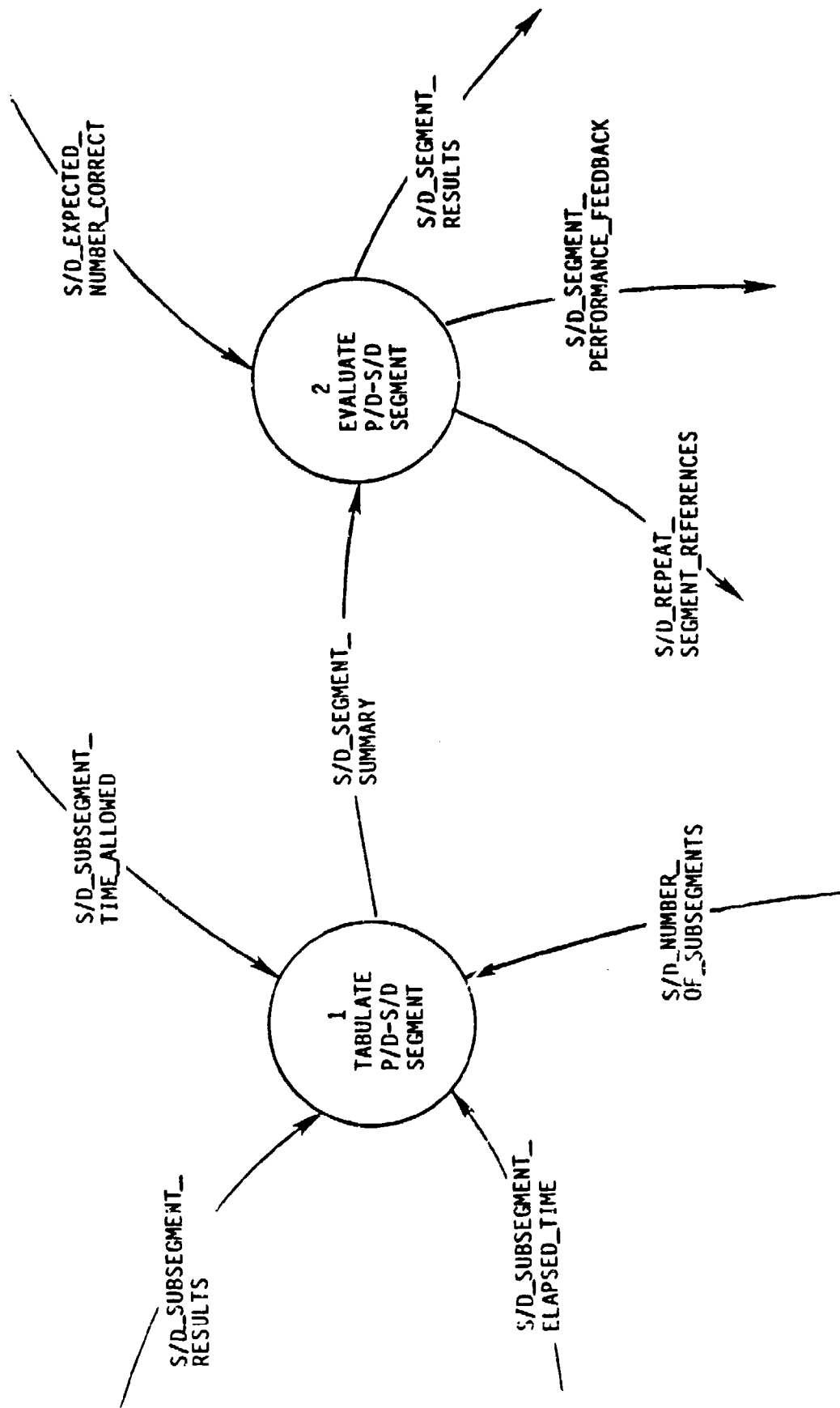


PROCESS 3.3.2: SUPERVISE P/D-M/S SEGMENT ASSESSMENT

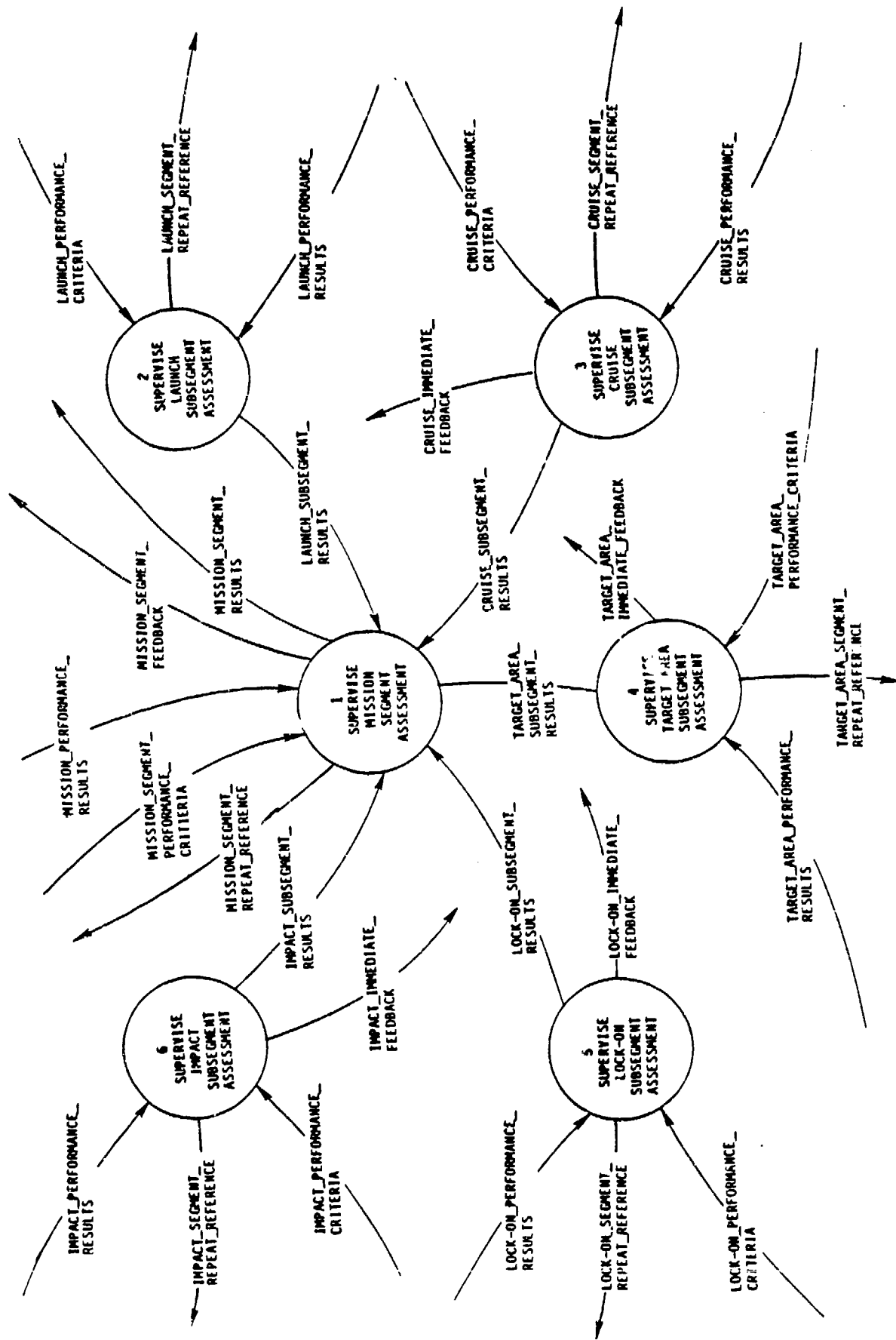
PROCESS 3.3.3: SUPERVISE P/D-S/D SUBSEGMENT ASSESSMENT



PROCESS 3.3.4: SUPERVISE P/D-S/D SEGMENT ASSESSMENT

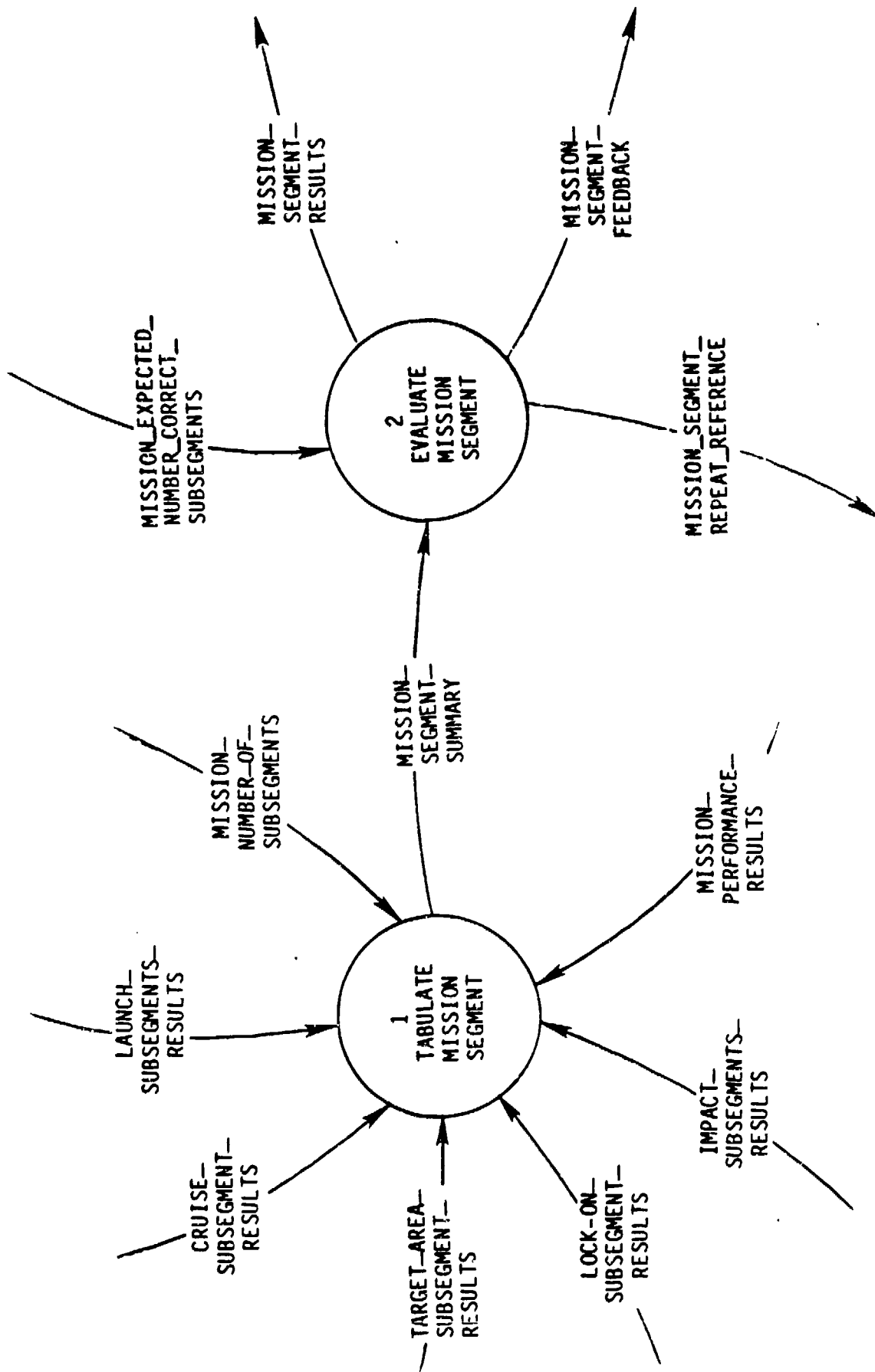


PROCESS 3.4: SUPERVISE MISSION ASSESSMENT

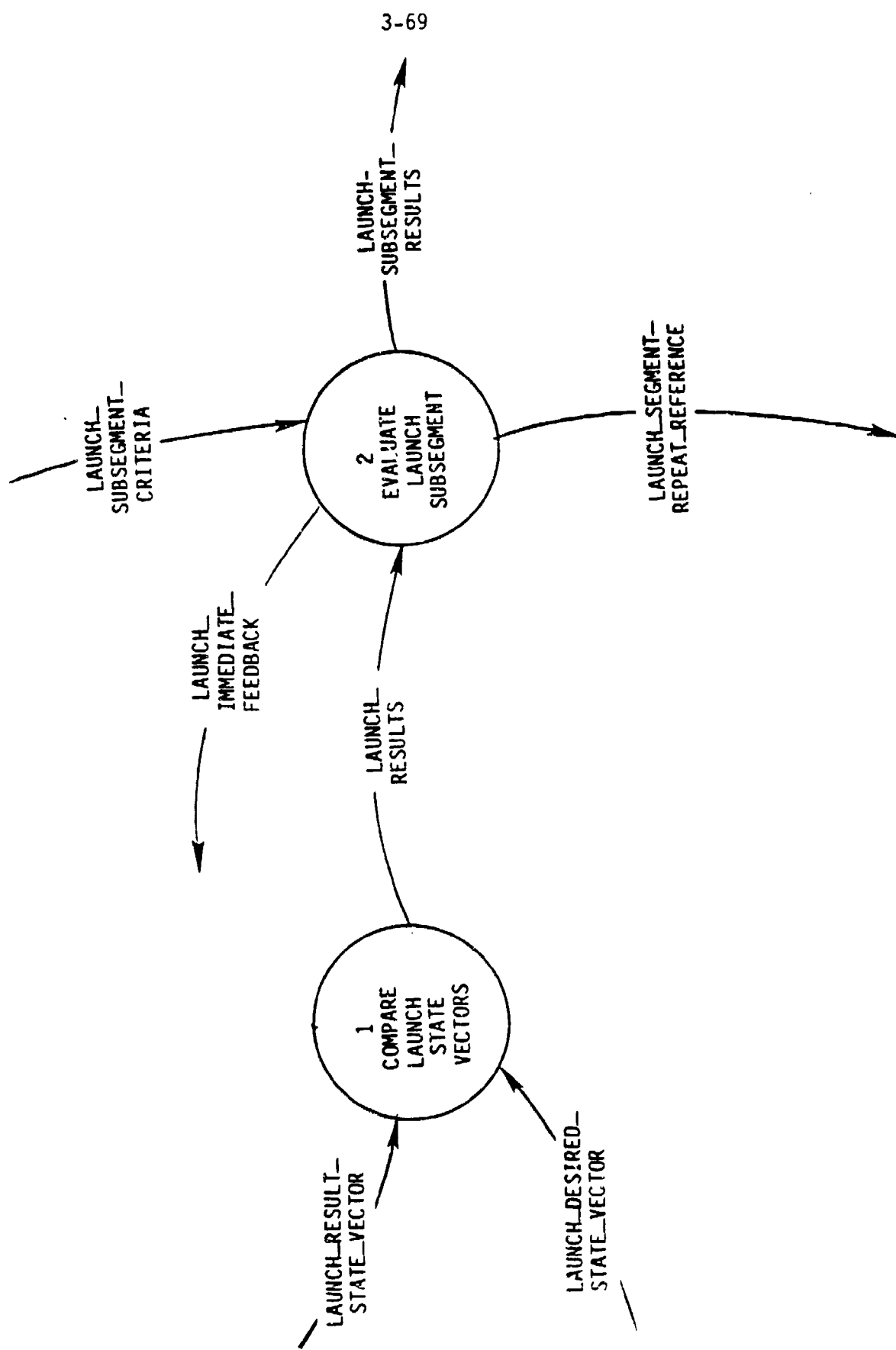


PROCESS 3.4.1: SUPERVISE MISSION SEGMENT ASSESSMENT

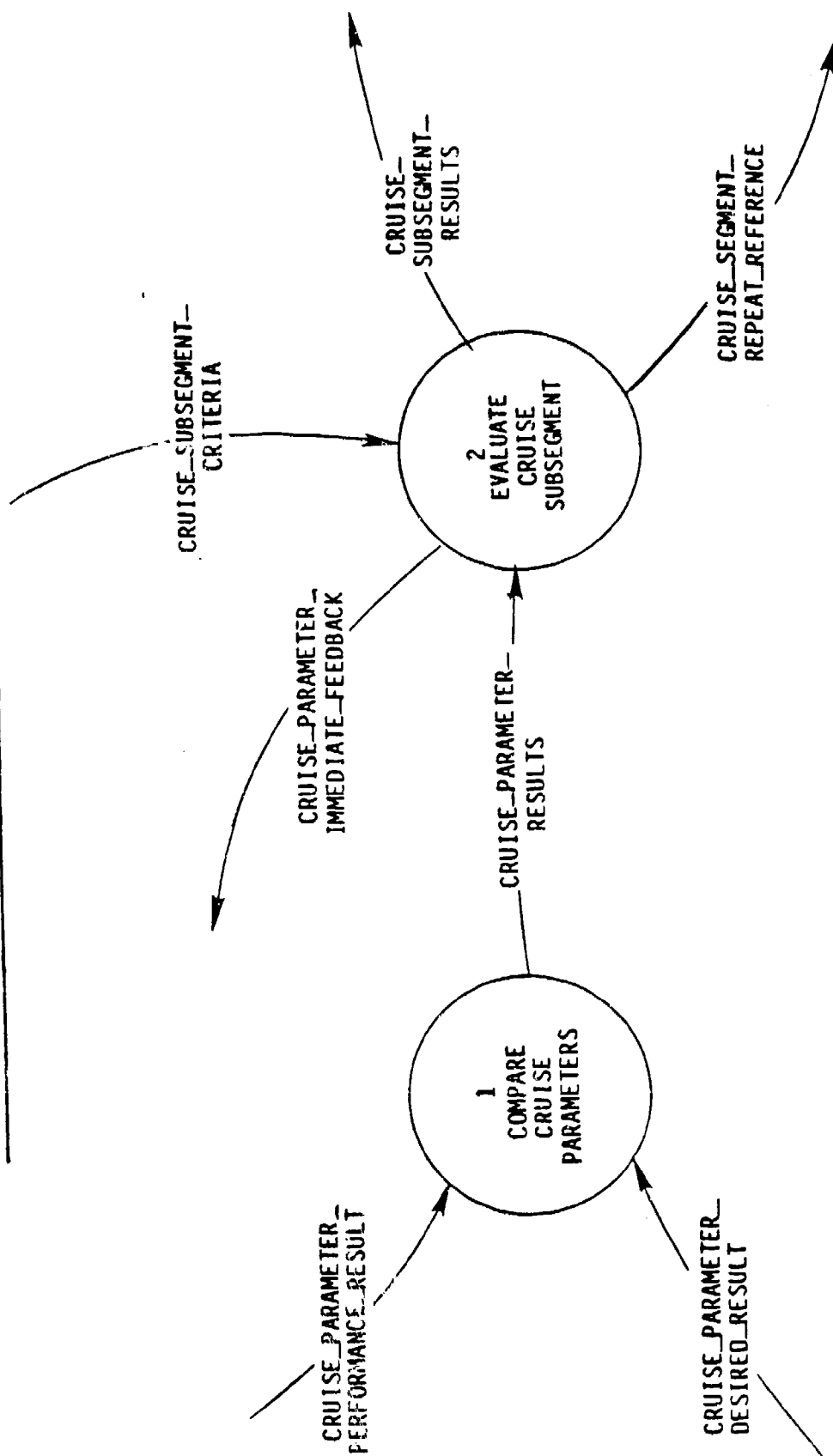
3-67



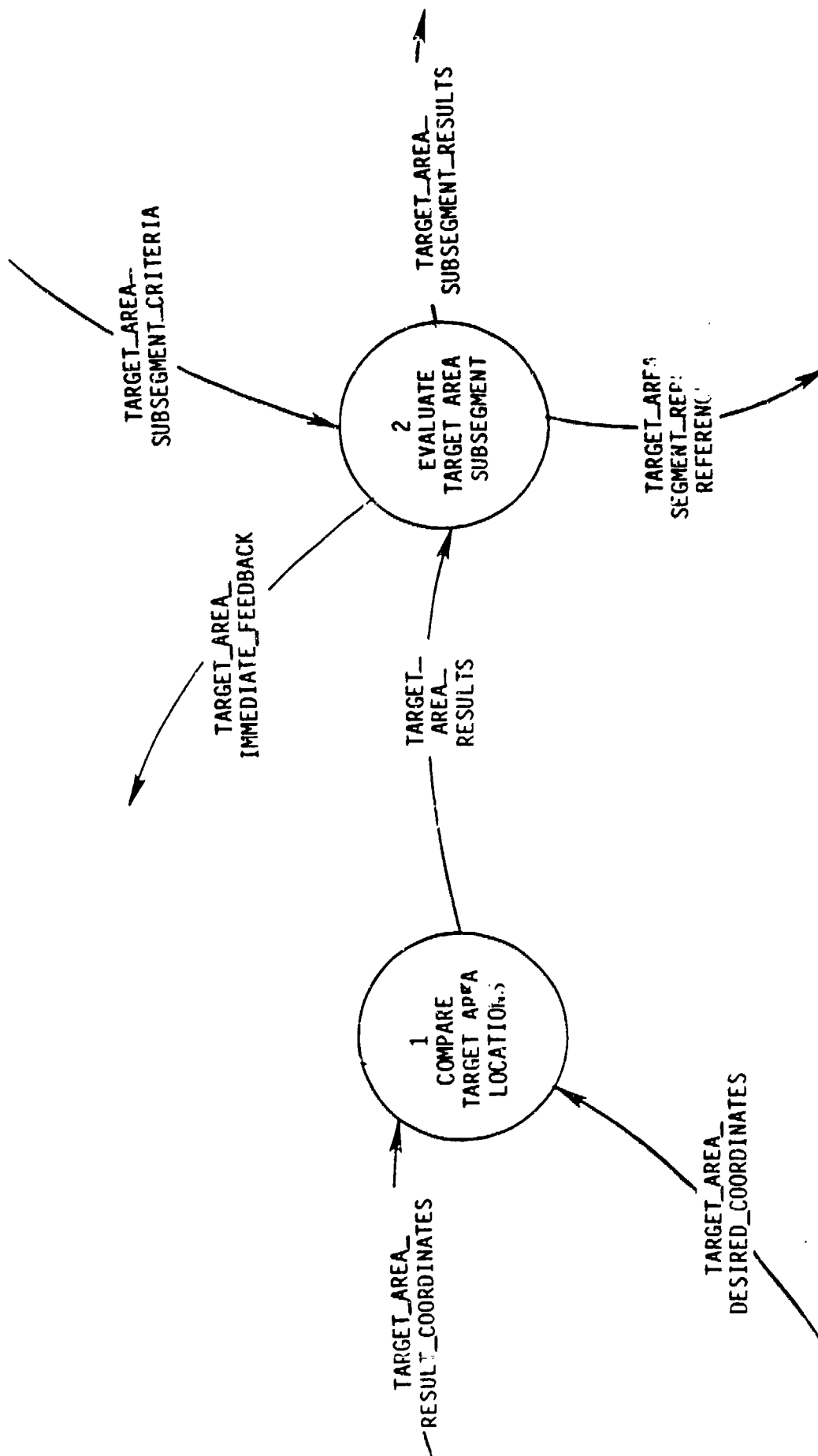
PROCESS 3.4.2: SUPERVISE LAUNCH SUBSEGMENT ASSESSMENT



PROCESS 3.4.3: SUPERVISE CRUISE SUBSEGMENT ASSESSMENT



PROCESS 3.4.4: SUPERVISE TARGET AREA SUBSEGMENT ASSESSMENT



Altitude_Update	* Specifies DPG display orientation *
	Roll_Angle + Pitch_Angle + Heading_Angle + Ground_Speed + Altitude + Sun_Angle
DPG_Commands	* Specifies map being loaded and * * configuration of DPG map video *
DPG_Navigation_Update	* Specifies which portion of loaded * * map is displayed *
DPG_Target_Update	* Specifies type and position of target*
DPG_Updates	* Updates and Commands issued to DPG *
	DPG_Navigation_Update + DPG_Attitude_Update + DPG_Target_Update + DPG_Commands
Encoded_Video_Database	* NTSC version of a video frame with * * encoded transform and position data *
	Analog_Video_Frame + {Frame_Transforms} + Present_Frame_GCS_Position
Environmental_State_Vector	* Wind_Velocity_Reference and * * lighting level/angles *
Fiber_Turns	* Indicates total length of flight *
Fin_Commands	* Originated in the operational system * * and communicated to the simulators * * via the FOL Uplink *
	Fin 1 (deg) + Fin 2 (deg) + Fin 3 (deg) + Fin 4 (deg)

Fin_Positions	* Calculated by the missile simulator * * and communicated to the operational * * system via the FOL Downlink *
	Fin 1 (deg) + Fin 2 (deg) + Fin 3 (deg) + Fin 4 (deg)
Final_Phase_Conditions	* Simulator terminating conditions *
	[Trigger_pull Fiber_Turns Altitude]
FOL_Uplink/Downlink	* Fiber optic link *
	Body_Rates Fin_Commands Fin_Positions Joystick_Movements Seeker_System_Downlink Initial_Uplink_Information
Frame_Information_Set	* Information encoded on each video * * frame *
	Possible_Projected_Frame_#'s + Frame_Transform_Set + Present_Frame_GCS_Position
Frame_Number	* Video disk frame references *
Frame_Numbers	* Unique video disk frame numbers *
	{ Frame_Number }
Frame_Transform	* Frame transform coefficients *
Frame_Transform_Set	* Needed to calculate the manipulation * * polynomial *
	{ Frame_Transform }

Function_Keys	* Function keys on the gunner station *
	[Alternate_Display Change_Map Iris_Auto Iris_Manual Iris_Open Iris_Close Track_B/W Track_W/B Record_Auto Record_Manual Spares]
Ground_Reference	* Voltage reference *
Historical_Best_Fit_Frame_Number	* Historical information containing * * the best fit frame number for a * * seeker simulation start-up *
	Best_Fit_Frame_Number
Historical_Data	* Historical data recorded for * * simulation startup *
	Historical_GCS_Seeker_Position + Historical_Best_Fit_Frame_Number + Historical_Zoom_Position
Historical_GCS_Missile_Angles	* Recorded deflections from previous * * missile flights *
	Phi + Psi + Theta
Historical_GCS_Missile_Coords	* Recorded coordinates from previous * * missile flights *
	X + Y + Z
Historical_GCS_Missile_Positions	* Recorded path from previous * * missile flights *
	Historical_GCS_Missile_Angles + Historical_GCS_Missile_Coords

Historical_GCS_Positions	* Data recorded for subsequent * use by flight simulators	* *
Historical_GCS_Missile_Positions + Best_Fit_Frame_Number + Zoom_Factor + Current_GCS_Seeker_Position		
Historical_GCS_Seeker_Position	* Historical information containing a * seeker position in the gunner * coordinate system	* * *
Current_GCS_Seeker_Position		
Historical_Position_Reference	* File reference to Historical_GCS_ * Missile_Position	* *
Historical_Zoom_Position	* Historical information on the zoom * position for the seeker simulator	* *
Zoom_Position		
H/M/R_Indicator	* Indication of an M/S target hit, * missed or hit on the second attempt	* *
[Hit_indicator Miss_Indicator Retry_Hit_Indicator]		
Impact_Desired_Coordinates	* Coordinates of desired impact	*
Impact_Immediate_Feedback	* Immediate feedback of impact time * and miss distance	* *
Impact_Performance_Criteria	* All criteria used to assess impact * performance during impact training	* *
Impact_Subsegment_Criteria + Impact_Desired_Result +		
Impact_Performance_Results	* Results of cruise training assessment*	
Impact_Result_Location		

Impact_Phase_Courseware	* Optional impact simulation start-up, *	*
	* control, and target IDs	*
Simulation_State_Vector		
Impact_Phase_Graphover_CW	* Impact training prompts	*
Impact_Phase_Time	* Length of time of impact training	*
Impact_Result_Coordinates	* Coords of simulated missile impact	*
Impact_Results	* Results of impact location comparison*	
Impact_Miss_Distance +		
Impact_Elapsed_Time		
Impact_Segment_Reference	* Pointer to impact segment	*
	* courseware	*
Impact_Segment_Repeat_Reference	* Pointer to impact segment to	*
	* be repeated	*
Impact_Subsegment_Criteria	* Criteria used to evaluate impact	*
Impact_Error_Radius +		
Impact_Time_Allowed +		
Impact_Segment_Repeat_Reference		
Impact_Subsegment_CW_File	* File containing mission impact	*
	* training subsegment courseware	*
Impact_Subsegment_Description +		
Impact_Subsegment_Performance_Crit		
Impact_Subsegment_Description	* Description of cruise courseware	*
	* including prompts, timing, and	*
	* optionally, simulation state vectors	*
Impact_Phase_Graphover_CW +		
Impact_Phase_Time +		
(Impact_Phase_Courseware)		
Impact_Subsegment_Results	* Status of impact subseg indicating	*
	* if missile impacted within desired	*
	* distance from target and time allowed*	*
Initial_GCS_Missile_Position	* Initial missile position in gunner	*
	* station coordinates	*

X +

Y +

Z

Initial_GCS_Missile_Position	* Initial speed and heading of missile *
	dx/dt + dy/dt + dz/dt + dPhi/dt + dPsi/dt + dTheta/dt
Initial_GCS_Thrust	* Initial missile thrust in gunner * * station coordinates *
	Force X + Force Y + Force Z
Initial_GCS_Missile_Velocity	* Initial speed and heading of missile *
	dx/dt + dy/dt + dz/dt + dPhi/dt + dPsi/dt + dTheta/dt
Initial_Simulation_State_Vector	* Start up information for simulators *
	Historical_Position_Reference + Environmental_Reference + Simulation_State_Vector
Initial_Uplink_Information	* Initial output from FOG_M computer * * to missile simulator *
	Ground_Reference + 28v_Supply_Reference + Actuator_Bottle_Pressure + Initial_GCS_Thrust
Input_Device_Type	* Gunner input devices *
	[Push_Buttons Function_Keys Keypad Joystick_Controls Altitude_Increment/Roll]

Item_Courseware_File	* File containing item courseware	*
	Item_Description + Item_Performance_Criteria	
Item_Description	* Description of item sequence	*
	Item_Type + Item_Reference	
Item_Grade	* Post-item grade (pass/fail)	*
	P/F_Indicator	
Item_Performance_Criteria	* Item level performance criteria	*
	* used to determine whether or not	*
	* the item performance record is to	*
	* be updated	*
Item_Reference	* Courseware item types	*
	[M/C_Item_Reference P/D_Item_Reference Mission_Item CAI_Item]	
Joystick_Controls	* Joystick input devices	*
	[Manual/Auto_Switch Pitch/Yaw_Control Inc/Dec_Field_of_Vision Trigger]	
Joystick_Movement	* Joystick position from operational	*
	* system	*
Keypad	* Keypad input device	*
	Alphanumeric_Keys	
Launch_Desired_Response_Time	* Desired launch training response time*	
Launch_Desired_State_Vector	* Desired state of launch parameters	*
Launch_Immediate_Feedback	* Feedback of launch training status	*
	* including which params deviated from	*
	* desired values & amount of deviation	*
Launch_Graphover_Courseware	* Any training related launch prompts	*

Launch_Performance_Criteria	* All criteria used to assess launch	*
	Launch_Subsegment_Criteria + Launch_Desired_State_Vector	
Launch_Phase_Courseware	* Initial launch state vector	*
Launch_Result_State_Vector	* Post-launch training state vector	*
Launch_Segment_Reference	* Pointer to launch segment	*
	* courseware	*
Launch_Segment_Repeat_Ref	* Pointer to launch segment to	*
	* be repeated	*
Launch_Subsegment_Criteria	* Criteria used to evaluate launch	*
	Launch_State_Vector_Tolerance + Launch_Desired_Response_Time + Launch_Segment_Repeat_Reference	
Launch_Subsegment_CW_File	* File containing mission launch	*
	* training subsegment courseware	*
	{ Launch_Subsegment_Description + Launch_Subseg_Performance_Crit }	
Launch_Subsegment_Description	* Subsegment courseware description	*
	Launch_Phase_Graphover_CW + Launch_Phase_Courseware	
Launch_Subsegment_Results	* Status of launch subseg indicating	*
	* if launch carried out within the	*
	* allowed time and launch parameter	*
	* error tolerances	*
Lock-on_Desired_Coordinates	* Location of desired target lock-on	*
Lock-on_Desired_Response_Time	* Length of time desired to attain	*
	* training lock-on	*
Lock-on_Immediate_Feedback	* Immediate feedback of lock-on time	*
	* and distance from target	*
Lock-on_Miss_Distance	* Allowable distance to miss lock-on	*

Lock-on_Performance_Criteria	* All lock-on related performance crit.*	
	Lock-on_Subsegment_Criteria +	
	Lock-on_Desired_Target_Coordinates +	
Lock-on_Phase_Courseware	* Optional lock-on courseware for	*
	* simulation start-up, control states	*
	* and target locations	*
	Simulation_State_Vector	
Lock-on_Phase_Graphover_CW	* Training prompt overlay	*
Lock-on_Phase_Time	* Length of time of lock-on training	*
Lock-on_Projected_Seeker_Coords	* Ground coordinates as projected from	*
	* the missile seeker cross-hairs	*
Lock-on_Results	* Results of lock-on location	*
	* comparison	*
	Lock-on_Miss_Distance +	
	Lock-on_Elapsed_Time +	
Lock-on_Segment_Reference	* Pointer to lock-on segment	*
	* courseware	*
Lock-on_Segment_Repeat_Reference	* Pointer to lock-on segment to	*
	* be repeated	*
Lock-on_Subsegment_CW_File	* File containing mission lock-on	*
	* training subsegment courseware	*
	Lock-on_Subsegment_Description +	
	Lock-on_Subseg_Performance_Crit	
Lock-on_Subsegment_Description	* Subsegment courseware description	*
	Lock-on_Phase_Graphover_Courseware +	
	Lock-on_Phase_Time +	
	(Lock-on_Phase_Courseware)	
Lock-on_Subsegment_Results	* Status of lock-on subseg indicating	*
	* if lock-on attained within error	*
	* radius and time allowed	*

Lock-on_Subsegment_Criteria	* Criteria used to evaluate lock-on *
	Lock-on_Error_Radius + Lock-on_Desired_Response_Time + Lock-on_Segment_Repeat_Reference
Lock-on_Target_Coordinates	* Locations of all targets to be * * encountered during lock-on phase *
	Target_Vector
Manipulation_Polynomial	* polynomial coefficients used by * * warper *
M/C_Answer	* key or pushbutton answer to M/C * * question *
	[M/C_Keypad_Answer M/C_PDP_Answer]
M/C_CAI_Segment_Courseware	* Courseware needed to run CAI * * instructional or explanatory material* * before or after an M/C segment *
M/C_CAI_Subsegment_Courseware	* Courseware needed to run CAI * * instructional or explanatory material* * before or after an M/C question *
M/C_Courseware	* Reference to a multiple choice item * * level courseware in the M/C_Segment_ * * Courseware_File *
	{ M/C_Segment_Reference + M/C_Segment_Performance }
M/C_Expected_Number_Correct	* number of correct answers needed to * * pass the segment *
M/C_Feedback	* Multiple choice gunner performance * * feedback *
	M/C_Immediate_Feedback + M/C_Segment_Performance_Feedback
M/C_Frame_Reference	* Video frame containing text and * * graphics of a multiple choice * * question *
	frame_number

M/C_Immediate_Feedback	* Feedback message indicating	*
	* correctness of gunners response,	*
	* with a retryt prompt if response was	*
	* incorrect	*
	[M/C_Keypad_Immediate_Feedback	
	M/C_PDP_Immediate_Feedback]	
M/C_Item_Reference	* Pointer to M/C courseware in the	*
	* M/C_Segment_Courseware_File	*
M/C_Incorrect_Subsegment	* Subsegment reference to an	*
	* incorrectly answered M/C question	*
	{ M/C_Subsegment_Reference }	
M/C_Input_Rules	* Device enables for M/C questions	*
	[M/C_Keypad_Input_Rules	
	M/C_PDP_Input_Rules]	
M/C_Keypad_Description	* Multiple choice keypad question	*
	* courseware and answer	*
	M/C_Keypad_Subsegment_Courseware +	
	M/C_Keypad_Answer	
M/C_Keypad_Immediate_Feedback	* Feedback message informing gunner of	*
	* correctness of response, plus	*
	* optionally a retry prompt	*
	[Correct_Indicator	
	Incorrect_Indicator +	
	(Retry_Prompt)	
M/C_Keypad_Input_Rules	* Key enables for M/C_Responses	*
	Keypad	
M/C_Keypad_Output	* Output from the Perform M/C Keypad	*
	* process of a keypad-type M/C question*	
M/C_Keypad_Results	* Gunner keypad responses to the	*
	* M/C_Question	*
M/C_Keypad_Subsegment_Courseware	* Courseware sent to the perform	*
	* process for a keypad subsegment	*
	M/C_Question +	
	M/C_Input_Rules	

69

M/C_Keypad_Answer	* Answer to the M/C question	*
Alphanumeric_Key		
M/C_Number_Correct	* Number of questions answered	*
	* correctly	*
M/C_Number_Incorrect	* Number of M/C_Questions answered	*
	* incorrectly	*
M/C_Number_Of_Questions	* Number of questions in the M/C	*
	* segment, i.e., the number of	*
	* subsegments	*
M/C_PDP_Answer	* Set of devices corresponding to a	*
	* PDP-type segment	*
	[PDP	
	Advance_Subfunction_Key	
	Fire_Switch]	
M/C_PDP_Description	* Multiple choice PDP question course-	*
	* ware and PDP answer	*
	M/C_PDP_Subsegment_Courseware +	
	M/C_Answer	
M/C_PDP_Immediate_Feedback	* Feedback message informing gunner of	*
	* correctness of response, plus	*
	* optionally a retry prompt and	*
	* PDP_Feedback	*
	[Correct_Message	
	Incorrect_Message +	
	(Retry_Prompt)] +	
	(PDP_Feedback)	
M/C_PDP_Input_Rules	* PDP enables for M/C PDP responses	*
	PDP_Input	
M/C_PDP_Output	* Output from the Perform M/C process	*
	* of a PDP type M/C question	*
M/C_PDP_Results	* Gunner PDP responses to M/C_Questions*	
M/C_Performance_Criteria	* Scoring and evaluation criteria for	*
	* a multiple choice segment	*
	M/C_Segment_Performance_Criteria +	
	M/C_Answer	

M/C_PDP_Subsegment_Courseware	* Courseware needed to run a PDP-type * question. Includes the lighting or * flashing of PDP's, as well as text * or frames displayed on the CRT	* * * *
	M/C_Question +	
	M/C_PDP_Input_Rules	
M/C_Question	* The page of text or video frame * reference of a multiple choice * question	* * *
	[M/C_Frame_Reference	
	M/C_Text_Reference]	
M/C_Repeat_Subsegment_List	* List of M/C subsegment references * to questions to be repeated	* *
	{ M/C_Subsegment_References }	
M/C_Result	* Gunner answers to M/C questions	*
	[M/C_Keypad_Results	
	M/C_PDP_Results]	
M/C_Segment_Courseware_File	* File containing multiple choice * segment courseware descriptions	* *
	{ M/C_Segment_Descriptions }	
M/C_Segment_Description	* List of segments to be sequenced * and the courseware or references * needed for each	* * *
	(M/C_CAI_Segment_Courseware) +	
	M/C_Segment_Reference +	
	(M/C_CAI_Segment_Courseware)	
M/C_Segment_Performance_Criteria	* Segment performance parameters	*
	M/C_Number_Of_Questions +	
	M/C_Expected_Number_Correct	
M/C_Segment_Performance_Feedback	* Textual information informing the * gunner as to how many questions were * answered correctly, and whether or * not the segment was passed	* * * *
	{ Alphanumeric_String }	

M/C_Segment_Reference	* Pointer to multiple choice segment * level courseware description	* *
M/C_Segment_Results	* M/C segment pass or fail indicator	*
	P/F_Indicator	
M/C_Segment_Summary	* Tabulated results of the multiple * choice segment	* *
	M/C_Number_Correct + M/C_Number_Incorrect + { M/C_Incorrect_Subsegment }	
M/C_Subsegment_Courseware_File	* File containing multiple choice * subsegment courseware descriptions	* *
	{ M/C_Subsegment_Description }	
M/C_Subsegment_Description	* Multiple choice subsegment * descriptions	* *
	(M/C_CAI_Subsegment_Description) + [M/C_Keypad_Description M/C_PDP_Description] (M/C_CAI_Subsegment_Description)	
M/C_Subsegment_Reference	* References to multiple choice * subsegment level courseware * descriptions	* * *
M/C_Subsegment_Results	* Correctness of gunner response	*
	[Correct_Indicator Incorrect_Indicator]	
M/C_Text_Reference	* Pointer to a page of text containing * a multiple choice question	* *
	{ Alphanumeric_String }	
MCS_To_GCS_Transform	* Matrix used to rotate and translate * coordinates from the missile * coordinate system (MCS) to the * gunner coordinate system (GCS) or * vice versa	* * * * *
Menu_Hierarchy	* Order of appearance of menu items	*
	{ Menu_Items }	

Menu_ID	* Unique ID of menu (items) in hierarchy	*
Menu_Input_Rules	* Set of valid menu responses	*
Menu_I/O	* I/O specific to menus	*
	[Lesson_Menu_I/O Topic_Menu_I/O Item_Menu_I/O]	
Menu_Item	* Gunner selectable items	*
	Menu_Item_ID + Menu_Item_Type	
Menu_Item_Choice	* Menu item identifier	*
	Alphanumeric_Character	
Menu_Item_Class	* Categories of menu items	*
	[Parent_Menu_Item Next_Menu_Item Subordinate_Menu_Item_List Current_Menu_Item]	
Menu_Item_Description	* File containing menu item scripts * associated with menu item ID's	*
Menu_Item_ID	* Unique ID of menu item in hierarchy	*
Menu_Item_List_Output	* Menu item output components	*
	{ Menu_Item_Choice + Menu_Item_Script + Menu_Item_Performance_Record }	
Menu_Item_Performance_History	* Record of pass/fail on corresponding menu items	*
	{ Menu_Item_Performance_Record }	
Menu_Item_Performance_Record	* Gunner status of menu item	*
	[Menu_Item_Completed Menu_Item_Not_Completed]	
Menu_Item_Script	* Description of menu item	*
	{ Alphanumeric_String }	

Menu_Item_Type	* Type can be another menu or a * training segment	*
	[Item Menu]	
Menu_Output	* Name of menu and list of belonging * menu items	*
	Menu_Heading_Output + Menu_Item_List_Output	
Menu_Response	* Gunner menu selections	*
	Alphanumeric_Character	
Menu_Selection_Rules	* Set of possible gunner selections	*
	{ Menu_Item_Choice + Menu_Item }	
Menu_Template	* File of menu type structures	*
	{ Menu_Type_Template }	
Menu_Type	* Basic menu types	*
	[Subordinate_Item_Menu Passed_Item Failed_Item_Menu]	
Menu_Type_Template	* Set of menu item classifications	*
	{ Menu_Item_Class }	
Mission_Courseware	* Mission courseware types	*
	[Mission_Segment_CAI_Courseware Launch_Phase_Courseware Cruise_Phase_Courseware Target_Area_Phase_Courseware Lock-on_Phase_Courseware Impact_Phase_Courseware]	
Mission_Expected_#_Correct_Subsegs	* Number of correct subsegments * needed to pass mission segment	*

Mission_Feedback	* Mission feedback type	*
	[Mission_Immediate_Feedback Launch_Immediate_Feedback Cruise_Immediate_Feedback Target_Area_Immediate_Feedback Lock-on_Immediate_Feedback Impact_Immediate_Feedback]	
Mission_Immediate_Feedback	* Indication of crash	*
Mission_Item_Reference	* Pointer to item level mission * courseware	*
Mission_Number_of_Subsegments	* Number of subsegments in a mission	*
Mission_Ordered_Segment_Reference	* Mission segment references along * with their cardinality	*
	[Launch_Segment_Reference + Card1 Cruise_Segment_Reference + Card2 Target_Area_Segment_Reference + Card3 Lock-on_Segment_Reference + Card4 Impact_Segment_Reference + Card5n]	
Mission_Performance_Criteria	* Performance criteria of mission * segment types	*
	[Mission_Segment_Performance_Criteria Launch_Subseg_Performance_Criteria Cruise_Subseg_Performance_Criteria Target_Area_Subseg_Performance_Crit. Lock-on_Subseg_Performance_Criteria Impact_Subseg_Performance_Criteria]	
Mission_Performance_Results	* Final state vector from flight/video * simulation -- this indicates missile * crash max fiber turns attained	*
	Mission_Final_State_Vector	
Mission_Repeat_Reference	* Mission repeat reference types	*
	[Mission_Segment_Repeat_Reference Launch_Segment_Repeat_Reference Cruise_Segment_Repeat_Reference Target_Area_Segment_Repeat_Reference Lock-on_Segment_Repeat_Reference Impact_Segment_Repeat_Reference]	

Mission_Results	* Mission result types *
	[Mission_Performance_Results Launch_Performance_Results Cruise_Performance_Results Target_Area_Performance_Results Lock-on_Performance_Results Impact_Performance_Results]
Mission_Segment_CAI_Courseware	* Pre-mission CAI courseware *
	CAI_Courseware
Mission_Segment_Courseware_File	* Segment level mission courseware *
	Mission_Segment_Description + Mission_Segment_Performance_Criteria
Mission_Segment_Description	* Ordered list of mission segment * * references; note that the cardinality* * of any segment in the list must not * * exceed that of any subsequent * * segment in the list *
	(Mission_Segment_CAI_Courseware) + { Mission_Ordered_Segment_Reference }
Mission_Segment_Feedback	* Post-mission results feedback *
Mission_Segment_Performance_Crit.	* Performance Criteria used to assess * * entire mission segment *
	Mission_Expected_Number_Correct_Subsegs + Mission_Number_of_Subsegments + Mission_Segment_Repeat_Reference
Mission_Segment_Repeat_Reference	* Pointer to repeated mission segment *
Mission_Segment_Result	* Pass/Fail status of mission segment *
Mission_Segment_Summary	* Summary of tabulated subsegments *
	Mission_Subsegments_Correct + Mission_Crash_Status
M/S_CAI_Subsegment_Courseware	* Courseware needed to run * * instructional or explanatory material* * before or after an M/S subsegment *

M/S_Desired_Target_ID	* Reference to a particular M/S target *
	* (map symbol) to be found by gunner *
M/S_Desired_Target_Location	* Screen coordinates of prompted target*
	Target_Coordinate +
	Target_Radius
M/S_Expected_Number_Correct	* Number of target hits needed to *
	* pass the segment *
M/S_Expected_Number_Targets	* Total number of targets in the *
	* segment *
M/S_Feedback	* Gunners performance feedback for a *
	* multiple static target problem *
	M/S_Immediate_Feedback +
	M/S_Segment_Performance_Feedback
M/S_Immediate_Feedback	* Textual feedback message indicating *
	* whether the prompted target was hit *
	* or missed, with a possible retry *
	* prompt and graphic indicating correct*
	* target *
	{ Alphanumeric_String } +
	(Retry_Prompt) +
	(Desired_Target_Graphic)
M/S_Incorrect_Target_Marked	* Indication that an incorrect target *
	* was hit instead of prompted target *
	Wrong_Target_Indicator +
	Desired_Target_Coordinate
M/S_Input_Rules	* Joystick enables for M/S gunner *
	* responses *
M/S_Item_Reference	* Reference to an M/S segment in the *
	* M/S_Segment_Courseware_File *
M/S_Missed_Target_Result	* M/S_Result not matching prompted *
	* target *
	Desired_Target_Coordinate
M/S_Number_Correct	* Number of M/S targets (map symbols) *
	* marked correctly *

M/S_Number_Incorrect	* Number of M/S targets (map symbols) * marked incorrectly	* *
M/S_Number_Retries_Correct	* Number of M/S targets (map symbols) * corrected on the second attempt	* *
M/S_Number_Time_Exceeded	* Number of times the subsegment time * limit has been exceeded	* *
M/S_Performance_Criteria	* Scoring and evaluation criteria for * a multiple static target problem	* *
	M/S_Target_Locations + M/S_Desired_Target_ID + M/S_Segment_Performance_Criteria	
M/S_Repeat_Segment_Reference	* A complete M/S segment training * reference	* *
	M/S_Item_Reference	
M/S_Results	* Cross hair screen coordinates at * trigger pull	* *
	Cross_Hair_Coordinates	
M/S_Segment_Courseware_File	* File of multiple static target * point disk courseware descriptions	* *
	{ M/S_Segment_Description }	
M/S_Segment_Description	* Consists of the CAI courseware and * M/S subsegment references that make * up the M/S segment	* * *
	P/D_CAI_Segment_Courseware + { M/S_Subsegment_Reference }	
M/S_Segment_Performance_Criteria	* M/S performance parameters needed to * calculate pass or fail of segment	* *
	M/S_Subsegment_Time_Allowed + M/S_Expected_Number_Targets + M/S_Expected_Number_Correct	
M/S_Segment_Performance_Feedback	* Textual/graphic feedback message * informing gunner of segment per- * formance and pass or fail	* * *
	{ Alphanumeric_String }	

M/S_Segment_Results	* Results of the M/S segment evaluation*
	P/F_Indicator
M/S_Segment_Summary	* Profile of segment performance *
	* for feedback and evaluation purposes *
	M/S_Number_Correct +
	M/S_Number_Incorrect +
	M/S_Number_Time_Exceeded +
	M/S_Number_Retrieves_Corrected
M/S_Subsegment_Courseware	* Courseware sent to the perform *
	* process to run the M/S subsegment *
	M/S_Target_Locations +
	Target_Radius +
	M/S_Input_Rules +
	M/S_Subsegment_Time_Allowed +
	DMG_Reference
M/S_Subsegment_Courseware_File	* File containing multiple static *
	* target courseware descriptions *
	{ M/S_Subsegment_Description }
M/S_Subsegment_Description	* M/S Subsegments to be sequenced and *
	* the courseware or references needed *
	* for each *
	(M/S_CAI_Subsegment_Courseware) +
	M/S_Subsegment_Courseware +
	M/S_Desired_Target_ID +
	(M/S_CAI_Subsegment_Courseware)
M/S_Subsegment_Elapsed_Time	* Time from moment gunner is prompted *
	* for a target to moment gunner pulls *
	* joystick trigger *
M/S_Subsegment_Results	* Indication of prompted target hit *
	* or missed, and subsegment time *
	* exceeded (in seconds) *
	H/M/R_Indicator +
	Time_Exceeded
M/S_Subsegment_Time_Allowed	* time allowed to mark a target (in *
	* seconds) -- does not terminate *
	* subsegment if exceeded *

M/S_Target_Locations	* List of target screen coordinates * * corresponding to the target location * * on the DMG *
	{ Target_Coordinates + Target_Radius }
M/S_Target_Marked	* Prompted target marked by gunner *
	Hit_Indicator
M/S_Target_Missed	* Indication that no targets were hit *
	Miss_Indicator + Desired_Target_Coordinate
M/S_Undesired_Targets	* List of all targets other than * * prompted target *
	{ Target_Coordinates }
P/D_CAI_Segment_Courseware	* Courseware needed to run * * instructional or explanatory * * material before or after each * * segment *
P/D_Courseware	* Information needed to run a P/D * * segment *
	P/D_CAI_Segment_Courseware + [P/D-M/S_Segment_References P/D-S/D_Segment_References]
P/D_Item_Reference	* Reference to point disk item * * level courseware in the P/D_Segment_ * * Courseware_File *
	[M/S_Item_Reference S/D_Item_Reference]
P/D-M/S_Segment_Reference	* Reference to a multiple static * * target (DMG) point disk problem *
P/D_Feedback	* Gunner feedback of a point disk * * problem *
	[M/S_Feedback S/D_Feedback]

P/D_Performance_Criteria	* Scoring and evaluation criteria for * point disk problems	* *
	[M/S_Performance_Criteria S/D_Performance_Criteria]	
PDP_Feedback	* Altering the state of the PDP(s) to * indicate the correct answer to an * M/C PDP question	* * *
PDP_Input	* PDP input devices	*
	PDPs + Advance_Subfunction_Key + PDP_Fire_Switch	
P/D_Repeat_Segment_Reference	* List of subsegments from the P/D * segment to be repeated	* *
	[M/S_Repeat_Segment_Reference S/D_Repeat_Segment_Reference]	
P/D_Result	* Gunners response to a point disk * subsegment	* *
	[M/S_Result S/D_Result]	
P/D-S/D_Segment_Reference	* Reference to a single dynamic * target (CGI) point disk problem	* *
P/D_Segment_Courseware_File	* File of point disk segment * courseware descriptions	* *
	[M/S_Segment_Courseware_Files S/D_Segment_Courseware_Files]	
P/D_Segment_Description	* Contains the segment descriptions * for M/S or S/D segments	* *
	[M/S_Segment_Description S/D_Segment_Description]	
P/D_Segment_Performance_Criteria	* Number of subsegments and number * needed to pass of each of the * subsegment types	* * *
	[M/S_Segment_Performance_Criteria S/D_Segment_Performance_Criteria]	

P/D_Segment_Results	* Pass or fail of segment	*
	[M/S_Segment_Results S/D_Segment_Results]	
P/D_Segment_Courseware_File	* File containing descriptions of * the point-disk subsegment sequence	*
	{ P/D_Segment_Descriptions }	
Performance_Criteria	* Segment performance criteria types	*
	[M/C_Performance_Criteria P/D_Performance_Criteria Mission_Performance_Criteria Item_Performance_Criteria]	
Performance_Records	* File of gunner training results	*
Performance_Results	* Gunner training results	*
	[M/C_Result P/D_Result Mission_Result]	
P/F_Indicator	* Indicates pass or fail of segments * or items	*
Possible_Projected_Frame_Numbers	* Frame numbers with highest * probability of being displayed next	*
	{ Frame_Numbers }	
Present_GCS_Frames_Position	* Unique coordinates of frame	*
Projected_GCS_Missile_Position	* Projected missile position in the * gunner coordinate system	*
	X + Y + Z + Phi + Psi + Theta	

Projected_GCS_Seeker_Position	* Projected seeker position in the * gunner coordinate system	* *
	X + Y + Z + Phi + Psi + Theta	
Projected_MCS_Seeker_Angles	* Projected seeker angles in the * missile coordinate system	* *
	Phi + Psi + Theta	
Relative_GCS_Frame_Vector	* Frame vector after execution of pan * instructions in gunner coordinates	* *
	Delta X + Delta Y + Delta Z	
Relative_GCS_Projected_Vector	* Vector from Current_GCS_Seeker_ * Position to Projected_GCS_Seeker_ * Position	* * *
	Delta X + Delta Y + Delta Z + Delta Phi + Delta Psi + Delta Theta	
Repeat_Training	* Segment repeat references	*
	[M/C_Repeat_Subsegment_List P/D_Repeat_Segment_Reference Mission_Repeat_Reference]	
Retry_Prompt	* A textual feedback message that * tells the gunner he is incorrect * and to try again	* * *
	{ Alphanumeric_String }	
S/D_CAI_Subsegment_Courseware	* Courseware needed to run * instructional or explanatory * material before or after an S/D	* * *

S/D_CGI_Function_Reference	* Reference to a particular CGI * geometric figure-generating * function. The function controls * the motion, speed and size of the * geometric figure -- parameters * provided by the S/D Subsegment * Courseware. (Level-of-difficulty is * implicit in the reference)	* * * * * * * *
S/D_Initialization_Parameters	* Data needed to prepare the CGI * generator for an S/D subsegment	* *
S/D_CGI_Function_reference + S/D_Initial_Target_Location		
S/D_Expected_Number_Correct	* Number of successful subsegments * needed to pass the segment and move * up to the next level of difficulty	* * *
S/D_Expected_Tracking_Time	* Amount of time gunner is expected * to maintain cross hairs on S/D * target (in seconds)	* * *
S/D_Immediate_Feedback	* Subsegment feedback consisting of the * longest time gunner maintained cross * hairs on the figure, and total time * maintained on the figure	* * * *
S/D_Longest_Time_Feedback + S/D_Sum_Of_Times_Feedback		
S/D_Initial_Target_Location	* Initial location of CGI function * that generates geometric figures * on the CRT	* * *
S/D_Input_Rules	* Set of joystick enables for the * gunner target tracking practice	* *
Joystick_Controls		
S/D_Item_Reference	* Reference to an S/D segment within * the S/D_Segment_Courseware_File	* *
S/D_Feedback	* Gunner feedback to a single dynamic * target point disk problem	* *
S/D_Immediate_Feedback + S/D_Segment_Performance_Feedback		

S/D_Longest_Time_Feedback	* Feedback message indicating the * success/failure of a subsegment and * the longest time the cross hairs were * centered on an S/D target	* * * *
S/D_Longest_Tracking_Time	* Longest continuous period in which * the cross hairs were centered on the * S/D Target (in seconds)	* * *
S/D_Number_Of_Subsegments	* The number of subsegments in an S/D * segment	* *
S/D_Performance_Criteria	* Scoring and evaluation criteria for * a single dynamic target problem	* *
	S/D_Expected_Tracking_Time + S/D_Segment_Performance_Criteria	
S/D_Repeat_Segment_Reference	* Repeat reference for a segment of * the same level of difficulty if * the segment just completed was * not passed	* * * *
	S/D_Item_Reference	
S/D_Result	* Screen coordinates of gunner * cross hairs	* *
	{ Cross_Hair_Coordinates }	
S/D_Segment_Courseware_File	* File containing single dynamic * point disk segment descriptions	* *
	{ S/D_Segment_Description }	
S/D_Segment_Description	* CAI segment courseware and S/D * subsegment references needed for an * S/D segment	* * *
	P/D_CAI_Segment_Courseware + { S/D_Subsegment_Reference }	
S/D_Segment_Performance_Criteria	* Performance parameters needed to * determine pass or fail of segment	* *
	S/D_Number_Of_Segments + S/D_Subsegment_Time_Allowed + S/D_Expected_Number_Correct	

S/D_Segment_Results	* Pass or fail indication of an S/D segment *
	P/F_Indicator
S/D_Segment_Summary	* Summary of all subsegment results for* * evaluation and performance feedback * * purposes *
	{ S/D_Longest_Tracking_Time + S/D_Summed_Response_Time }
S/D_Subsegment_Courseware	* Courseware sent to the perform * * process to run the S/D subsegment *
S/D_Subsegment_Courseware_File	* File containing single dynamic * * point disk subsegment descriptions *
	{ S/D_Subsegment_Description }
S/D_Subsegment_Description	* S/D subsegments to be sequenced * * and the courseware or references * * needed for each *
	(S/D_CAI_Subsegment_Courseware) + S/D_Subsegment_Courseware + Expected_Tracking_Time + Initial_Target_Location + Action_Reference + Subsegment_Courseware)
S/D_Subsegment_Elapsed_Time	* time elapsed since start of * * segment (in seconds) *
S/D_Subsegment_Results	* Results of target tracking practice. * * released at the end of the subsegment *
	S/D_Longest_Tracking_Time + S/D_Summed_Response_Times
S/D_Subsegment_Time_Allowed	* maximum time allowed to complete * * the subsegment tracking requirement * * (in seconds) *
S/D_Summed_Response_Times	* Sum of tracking times occurring in * * the course of a subsegment - released* * upon completion of the subsegment *
S/D_Sum_Of_Times_Feedback	* Feedback message indication the total* * length of time the S/D target was * * tracked by the gunner *

S/D_Target_Locations * Screen coordinates of S/D target *

 * (CGI geometric figure) *

{ Target_Coordinates }

S/D_Tracking_Time * Length of time the cross hair was *

 * centered on a S/D target during the *

 * course of a subsegment *

Seeker_Commands * Seeker yaw and pitch rate commands *

 * to keep seeker image stationary, and *

 * magnification change commands *

Seeker_Yaw_Rate +
Seeker_Pitch_Rate
Zoom_Command

Seeker_Pitch_Gimbal * Rate of change of pitch gimbal angles*

dPhi/dt
dPsi/dt
dTheta/dt

Seeker_Pitch_Rate * Rate of change of seeker pitch *

dPhi/dt
dPsi/dt
dTheta/dt

Seeker_System_Downlink * Seeker data to operational system *

Zoom_Position +
Seeker_Pitch_Rate +
Seeker_Yaw_Rate +
Seeker_Pitch_Gimbal +
Seeker_Yaw_Gimbal

Seeker_MCS_Position * Matrix of sines and cosines of *

 * seeker angles transforming *

 * seeker coordinates to missile coords *

Seeker_Video NTSC Signal

Seeker_Yaw_Gimbal * Rate of change of yaw gimbal angles *

dPhi/dt +
dPsi/dt +
dTheta/dt

Seeker_Yaw_Rate	* Rate of change of seeker yaw *
	dPhi/dt + dPsi/dt + dTheta/dt
Simulation_State_Vector	* Imulation state vector types *
	[Initial_Simulation_State_Vector Simulation_Control_State_Vector Target_Vector]
Slide_Paging	* CAI instructional or explanatory *
	* material displayed through a series *
	* of video disk slides *
State_Vector_Type	* Indicates the phase of a simulation *
	[Cruise_Type Target_Area_Type Lock-on_Type Impact_Type]
Supply_28V_Reference	* voltage reference *
Switched_Frame_Buffer	* digitized video frame *
Target_Coordinate	* Screen coordinates of a P/D target. *
	* a DMG map symbol or a CGI *
	* geometric figure *
Target_Radius	* Radius of acceptable target zone *
	* (in pixels) *
Target_Area_Coordinates	* Coordinates of center of target area *
Target_Area_Desired_Coordinates	* Location of desired target area *
Target_Area_Distance	* Defined distance(radius) of the *
	* desired target area *
Target_Area_Elapsed_Time	* Actual time taken during training to *
	* locate and mark target area *
Target_Area_Result_Coordinates	* Missile coordinates at time of *
	* target area training response *
Target_Area_Immediate_Feedback	* Immediate feedback of target area *
	* time and distance from target area *

Target_Area_Performance_Criteria	* All criteria used to assess target	*
	* area training performance	*
	Target_Area_Subsegment_Criteria +	
	Target_Area_Desired_Location +	
Target_Area_Performance_Results	* Training results used for assessment	*
	Target_Area_Missile_Coordinates	
Target_Area_Phase_Courseware	* Optional target area courseware used	*
	* to start simulation, alter a control	*
	* state or initialize targets	*
	Simulation_State_Vector	
Target_Area_Phase_Graphover_CW	* Target area training prompt overlay	*
Target_Area_Phase_Time	* Length of time of target area subseg	*
Target_Area_Radius	* The target area is defined as a	*
	* cylinder having a radius of Target	*
	* Area_Radius and projecting 'upward'	*
	* from the ground.	*
Target_Area_Results	* Results of target area comparison	*
	Target_Area_Distance +	
	Target_Area_Elapsed_Time	
Target_Area_Segment_Reference	* Pointer to target area segment	*
	* courseware	*
Target_Area_Segment_Repeat_Ref	* Pointer to target area segment to	*
	* be repeated	*
Target_Area_Subsegment_CW_File	* File containing mission training	*
	* target area courseware	*
	Target_Area_Subsegment_Description +	
	Target_Area_Performance_Criteria	
Target_Area_Subseg_Criteria	* Criteria used to evaluate the	*
	* target area subsegment	
	Target_Area_Error_Radius +	
	Target_Area_Desired_Response_Time +	
	Target_Area_Repeat_Reference	

Target_Area_Subsegment_Description	* Description of the target area	*
	* subseg. including time, prompts and	*
	* optionally, simulation start-up info	*
	Target_Area_Phase_Graphover_CW +	
	Target_Area_Phase_Time +	
	(Target_Area_Phase_Courseware)	
Target_Area_Subsegment_Results	* Status of target area subsegment	*
	* indicating if the missile was within	*
	* the allowed time and target distance	*
Target_Coordinates	* Screen coordinates for target in	*
	* M/S and S/D P/D problems	*
	Target_X_Coordinate +	
	Target_Y_Coordinate +	
	(Target_Z_Coordinate)	
Target_Reference	* Reference to an M/S target	*
	* (map symbol)	*
Target_Vector	* Target positions and type for DPG	*
Target_X_Coordinate	* Expected P/D answer X-screen coord	*
Target_Y_Coordinate	* Expected P/D answer Y-screen coord	*
Target_Z_Coordinate	* Expected Lock-on answer Z-screen	*
	* coordinate	*
Text_Paging	* CAI instructional or explanatory	*
	* material displayed through pages of	*
	* text on the CRT	*
Training_Courseware	* Training courseware types	*
	[CAI_Courseware	
	M/C_Courseware	
	P/D_Courseware	
	Mission_Courseware]	
Training_Results	* Initial scored results, i.e., prior	*
	* to feedback messages	*
	[P/D_Training_Results	
	M/C_Training_Results	
	Mission_Training_Results]	

Topic_Reference	* Pointer to topic level's item	*
	* courseware	*
Video_Disk_CAI_Files	* Source for video frames references	*
	* displayed on the CRT during a CAI	*
	* segment	*
	{ Frame_Number }	
Video_Disk_Images	NTSC Signal	
Video_Disk_M/C_Keypad_Files	* Source for video frame references	*
	* displayed on the CRT during an M/C	*
	* keypad subsegment	*
	{ Frame_Number }	
Video_Disk_M/C_PDP_Files	* Source for video frame references	*
	* displayed on the CRT during an M/C	*
	* PDP subsegment	*
	{ Frame_Number }	
Warped_Subframe	* Recalculated subframe	*
Winchester_CAI_Files	* Textual material displayed on the CRT*	
	* during a CAI segment or subsegment	*
Winchester_M/C_Keypad_File	* Textual material displayed on the	*
	* CRT during an M/C keypad subsegment	*
Winchester_M/C_PDP_File	* Textual material displayed on the	*
	* CRT during an M/C PDP subsegment	*
Wrong_Target_Indicator	* Indication of wrong target hit	*
Zoom_Command	* Command from gunner station to	*
	* missile system	*
Zoom_Position	* Zoom Frame	*

SECTION 5

PROCESS DESCRIPTIONS

This chapter contains the process descriptions (also referred to as "mini-specs") of primitives in the set of data flow diagrams. There is one mini-spec for each primitive process, i.e., a process not subdivided into subordinate processes in the data flow diagrams. The purpose of each mini-spec is to define the transformation of data flow arriving at its process into the data flows departing from the process. To state these transformation rules clearly, without giving a method of implementation, a language called structured English is used in most cases. Structured English, which resembles pseudo-code, uses sequences of statements (executed in order) that consist of computational descriptions, if-then decision structures, and repetition structures. In other cases, structured English is too restrictive (particularly in describing the mission simulation), and concise prose paragraphs are used instead.

1.1.1 SELECT MENU ITEM

DESCRIPTION: Match the menu response with the corresponding selection rule's menu item choice and set the selected menu item equal to that selection rule's menu time.

INPUTS: Menu_Response
Menu_Selection_Rules

OUTPUTS: Selected_Menu_Item

PSEUDOCODE: Repeat the following:
 get next Menu_Selection_Rule
 Until Menu_Item_Choice = Menu_Response
 Selected_Menu_Item ← Menu_Item

1.1.2 SELECT MENU TYPE

DESCRIPTION: If the menu item is of type item, initialize item sequencing, set next menu ID to the item ID and either set next menu type to passed item or failed segment, depending upon the item grade. Otherwise, the menu item type is menu; next menu is set to the menu item ID choice and next menu type becomes the menu of subordinate items.

INPUTS: Selected Menu_Item
Item_Grade

OUTPUTS: Topic Reference
Menu_ID
Menu_Type

PSEUDOCODE: Menu_ID ← Menu_Item_ID

If Menu_Item_Type = Item then
Item_ID ← Menu_Item_ID

Issue the Topic_Reference to begin Item sequencing.

Upon receiving the Item_Grade (item completed):

If Item_Grade = Item_Passed then
Menu_Type ← Passed_Item_Menu
Else Item_Grade = Item_Failed
Menu_Type ← Failed_Item_Menu

Else Menu_Item_Type = Menu
Next_Menu_Type ← Menu_Of_Subordinate_Items

1.2.1 SEQUENCE MENU TYPE'S ITEMS

DESCRIPTION: Get the menu form template associated with the next menu type.

INPUTS: Menu_Type
Menu_Template

OUTPUTS: Menu_Item_Class

PSEUDOCODE: Retrieve the Menu_Template associated with Menu_Type from the Template file

Repeat the following:

Get the next Menu_Item_Class from the Menu_Template

Until there are no more Menu_Item_Classes in the Menu_Template file.

1.2.2 GET MENU ITEMS

DESCRIPTION: Using the next menu item class and the menu's ID, access the menu hierarchy structure file and retrieve the menu item(s).

INPUTS: Menu_Item_Class
Menu_ID
Menu_Hierarchy

OUTPUTS: Menu_Item_Type
Menu_Item_ID

PSEUDOCODE: Retrieve the Menu_Hierarchy file record associated with the Menu_Item_Class and Menu_ID.

Assign the contents of the Menu_Hierarchy record to the Menu_Item_Type and the Menu_Item_ID.

1.2.3 BUILD MENU SELECTION RULES

DESCRIPTION: Assemble the menu selection rules by first joining each menu item ID and type to a unique menu item choice character to form a menu selection rule.

INPUTS: Menu Item ID
Menu_Item_Type

OUTPUTS: Menu Selection Rules
Menu_Item_Choice

PSEUDOCODE: Repeat the following:

Sequence to the next ordered Menu_Item_Choice
Issue the Menu_Item_Choice

Menu_Selection Rules ← Menu Item ID +
Menu_Item_Type +
Menu_Item_Choice

Until there are no more Menu_Item_IDs (Types)

1.3.1 LOAD MENU ITEM DESCRIPTION

DESCRIPTION: Using the Menu item ID, access the menu item description file and distribute the menu item script.

INPUTS: Menu_Item_ID
Menu_Item_Description

OUTPUTS: Menu_Item_Script

PSEUDOCODE: Retrieve the Menu_Script associated with the Menu_Item_ID from the Menu_Item_Description file.

1.3.2 LOAD MENU-ITEM PERFORMANCE HISTORY

DESCRIPTION: Using the menu item ID, access the menu item performance history file and distribute the menu item's performance record to the menu output process.

INPUTS: Menu_Item_ID
Menu_Item_Performance_History

OUTPUTS: Menu_Item_Performance_Record

PSEUDOCODE: Retrieve the Menu_Item_Performance_Record from the
Menu_Item_Performance_History file based upon the
Menu_Item_ID

1.4.1 BUILD MENU OUTPUT

DESCRIPTION: Build the menu output by first issuing the menu heading associated with the menu type and then collecting the menu items (script and choice char) into a list and issuing it.

INPUTS: Menu_Item_Choice
Menu_Item_Performance_Record
Menu_Type
Menu_Item_Script

OUTPUTS: Menu_Output

PSEUDOCODE: Issue the menu heading output associated with the
Next_Menu_Type

Repeat the following:

Match each Menu_Item_Script and Choice_Char with it's
associated Menu_Item_Performance_Record

Menu_Item_List_Output ? Menu_Item_Script +
Menu_Item_Choice +
Menu_Item_Performance_Record

Until there are no more Menu_Items

1.4.2 BUILD MENU INPUT RULES

DESCRIPTION: Build a set of valid input (G.S. response) characters by collecting all of the menu's menu item choice characters.

INPUTS: Menu_Item_Choice

OUTPUTS: Menu_Input_Rules

PSEUDOCODE: Repeat the following:

Accept each Menu_Item_Choice and add it to the
Menu_Input_Rules

Until there are not more Menu_Item_Choices

Enable the keypad keys contained in the Menu_Input_Rules

2.1.1 LOAD AND DISTRIBUTE ITEM COURSEWARE

DESCRIPTION: Load the item level courseware corresponding to the lesson item(reference from topic menu) chosen by the gunner. Distribute the item level description to the item sequencer and the performance criteria to the update item performance-record process.

INPUTS: Topic_Reference
Item_Courseware_File .

OUTPUTS: Item_Description
Item_Performance_Criteria

PSEUDOCODE: Load the Item_Description and Item_Performance_Criteria, corresponding to the Topic_Reference, from the Item_Courseware_File.

Issue the Item_Description for item type selection

Issue the Item_Performance_Criteria for performance record updating

2.1.2 SEQUENCE ITEM TYPE

DESCRIPTION: Sequence the training supervisor process corresponding to the type of lesson item chosen from the topic menu.

INPUTS: Item_Description

OUTPUTS: CAI_Item_Courseware
M/C_Item_Reference
P/D_Item_Reference
Mission_Item_Reference

PSEUDOCODE: Sequence the applicable Item_Reference

- Case 1 (CAI_Item)
Issue the CAI_Item_Reference to
begin the applicable CAI_Type training.
- Case 2 (M/C_Item)
Issue the M/C_Item_Reference
begin M/C training.
- Case 3 (P/D_Item)
Issue the P/D_Item_Reference to
begin P/D training.
- Case 4 (Mission_Item)
Issue the Mission_Item_Reference to
begin Mission training.

2.2.1.1 LOAD AND DISTRIBUTE M/C SEGMENT COURSEWARE

DESCRIPTION: Distribute the segment level courseware descriptions for multiple choice problems. The problems may be preceded or followed by CAI subsegments as specified in the segment description.

INPUT: M/C_Item_Reference
M/C_Segment_Courseware_File

OUTPUT: M/C_Segment_Description
M/C_Number_Of_Questions
M/C_Expected_Number_Correct

PSEUDOCODE: Upon receiving an M/C_Item_Reference

Repeat

Load and Issue from the M/C_Segment_Courseware_File

the M/C_Segment_Description
the M/C_Number_Of_Questions
the M/C_Expected_Number_Correct

Until no more segment references.

2.2.1.2 SEQUENCE M/C SEGMENT PARTS

DESCRIPTION: Recieves an M/C segment description and sequences the segment according to the description. M/C_CAI_Segment_Courseware may precede or follow the multiple choice segment as desired.

INPUT: M/C_CAI_Segment_Description

OUTPUT: M/C_Segment_Courseware
M/C_Segment_Reference

PSEUSOCODE: Upon recieving an M/C_Segment_Description

Repeat

Sequence M/C_CAI_Segment_Courseware

Issue an M/C_Segment_Reference

Sequence M/C_CAI_Segment_Courseware

Until no more segment descriptions.

2.2.2.1 LOAD AND DISTRIBUTE M/C SUBSEGMENT COURSEWARE

DESCRIPTION: Distribute the M/C_Segment_Descriptions corresponding to an M/C_Segment_Reference. For M/C, each subsegment is a question that may be enveloped by some CAI material.

INPUT: M/C_Segment_Reference
M/C_Subsegment_Courseware
M/C_Repeat_Subsegment_List

OUTPUT: M/C_Subsegment_Description

PSEUDOCODE: Upon receiving an M/C_Segment_Reference,

Load the M/C_Subsegment_Courseware_File
Issue the M/C_Subsegment_Description

Upon receiving an M/C_Subsegment_List,

Issue an M/C_Subsegment_Description corresponding to
the repeated subsegments

2.2.2.2 SEQUENCE M/C SUBSEGMENTS

DESCRIPTION: Sequence multiple choice question of two types: those involving PDP displays and answers, and those involving only the keypad. Both types are enveloped by CAI material when necessary.

INPUT: M/C_Segment_Description

OUTPUT: M/C_PDP_Subsegment_Courseware
M/C_PDP_Answer
M/C_CAI_Subsegment_Courseware
M/C_Keypad_Subsegment_Courseware
M/C_Keypad_Answer

PSEUDOCODE: Upon receiving an M/C_Segment_Description,

Repeat

Sequence M/C_CAI_Subsegment_Courseware

Sequence either

M/C_PDP_Subsegment_Courseware +
M/C_PDP_Answer

Or

M/C_Keypad_Subsegment_Courseware +
M/C_Keypad_Answer

Sequence M/C_CAI_Subsegment_Courseware

Until all subsegments have been sequenced.

2.3.1.1 LOAD AND DISTRIBUTE P/D SEGMENT COURSEWARE

DESCRIPTION: Distributes segment-level courseware for the two dimensional point-disk problem. Courseware will either be a multiple stationary targets (M/S) segment or a single dynamic target (S/D) segment. M/S segments correspond to map symbology (DMG) problems and S/D segments correspond to target tracking practice using CGI geometric figures.

INPUT: P/D_Item_Reference
 P/D_Segment_Courseware_File
 M/S_Repeat_Segment_Reference
 S/D_Repeat_Segment_Reference

OUTPUT: M/S_Segment_Performance_Criteria
 S/D_Segment_Performance_Criteria
 P/D_Segment_Description

PSEUDOCODE: Upon receiving a P/D_Item_Reference,

Repeat

Load the P/D_Segment_Description
 Issue the P/D_Segment_Description

If the segment is an M/S segment, then
 Load and Issue the M/S_Performance_Criteria
 Else
 Load and Issue the S/D_Performance_Criteria

Until no more segment references.

Upon receiving an M/S_Repeat_Segment_Reference

Load and Issue a M/S_Segment_Description
 Load and Issue an M/S_Segment_Performance_Criteria

Upon receiving an S/D_Repeat_Segment_Reference

Load and Issue an S/D_Segment_Description
 Load and Issue an S/D_Segment_Performance_Criteria

2.3.1.2 SEQUENCE P/D SEGMENT PARTS

DESCRIPTION: Recieves a segment description and sequences the various segment parts. CAI segments optionally precede or follow the M/S or S/D segments. M/S and S/D descriptions are references to subsegment courseware that is loaded and sequenced by the subsegment supervisors.

INPUT: P/D_Segment_Description

OUTPUT: P/D_CAI_Segment_Courseware
P/D-M/S_Segment_References
P/D-S/D_Segment_References

PSEUDOCODE: Upon recieving the P/D_Segment_Description,
Repeat
 Sequence P/D_CAI_Segment_Courseware
 If an M/S segment
 Sequence P/D_M/S_Segment_References
 Else
 Sequence P/D_S/D_Segment_References.
 Sequence P/D_CAI_Segment_Courseware
Until there are no more segments.

2.3.2.1 LOAD AND DISTRIBUTE P/D - S/D SUBSEGMENT COURSEWARE

DESCRIPTION: Loads the courseware for each subsegment of an S/D segment. Each subsegment consists of one geometric figure at one of four levels of difficulty and shown on the screen for a specified length of time.

INPUT: P/D-S/D_Segment_Reference
S/D_Subsegment_Courseware_File

OUTPUT: S/D_Subsegment_Description

PSEUDOCODE: Load the S/D_Subsegment_Courseware corresponding to the P/D-S/D_Segment_References.

Repeat

Issue an S/D_Subsegment_Description

Until there are no more subsegments.

2.3.2.2 SEQUENCE P/D - S/D TARGET SUBSEGMENTS

DESCRIPTION: Sequence the subsegments corresponding to the S/D segment descriptions. CAI subsegments precede or follow the S/D subsegment training and are issued when necessary. The S/D_CGI_Function_Reference is issued to the device or process responsible for generating the CGI image on the CRT. The S/D_Initial_Target_Location acts as a "seed" for the CGI function. The S/D_Expected_Tracking_Time is the length of time the gunner is expected to center the cross hairs on the CGI-generated figure. The subsegment courseware contains all other information needed to run the subsegment (such as device enables and timing requirements).

INPUT: S/D_Subsegment_Description

OUTPUT: S/D_CAI_Subsegment_Courseware
 S/D_Expected_Tracking_Time
 S/D_Initial_Target_Location
 S/D_Subsegment_Courseware
 S/D_CGI_Function_Reference

PSEUDOCODE: Upon receiving an S/D_Subsegment_Description:

Sequence S/D_CAI_Subsegment_Courseware

Issue the S/D_CGI_Function_Reference
 Issue the S/D_Initial_Target_Location
 Issue the S/D_Expected_Tracking_Time

Sequence the S/D_Subsegment_Courseware

Sequence S/D_CAI_Subsegment_Courseware

2.3.3.1 LOAD AND DISTRIBUTE P/D - M/S TARGET SUBSEGMENT COURSEWARE

DESCRIPTION: Loads the courseware for each subsegment of an M/S segment. Each subsegment consists of one of the targets on the CRT, so that a segment of ten targets has ten subsegments.

INPUT: P/D-M/S_Segment_Reference
M/S_Segment_Courseware_File

OUTPUT: M/S_Subsegment_Description

PSEUDOCODE: Load the M/S_Subsegment_Courseware corresponding to the P/D_Segment_References.

Repeat

Issue M/S_Subsegment_Description

Until there are no more subsegments.

2.3.3.2 SEQUENCE P/D - M/S TARGET SUBSEGMENTS

DESCRIPTION: Sequence the subsegments corresponding to the S/D segment descriptions. CAI subsegments precede or follow the S/D subsegment training and are issued when necessary. The S/D_Desired_Target_ID is the target the gunner is prompted to find and mark.

INPUT: M/S_Subsegment_Description

OUTPUT: M/S_Desired_Target_ID
M/S_Subsegment_Courseware
M/S_CAI_Subsegment_Courseware

PSEUDOCODE: Upon receiving an M/S_Subsegment_Description:

Sequence M/S_CAI_Courseware

Sequence M/S_Subsegment_Courseware
Issue the M/S_Desired_Target_ID

Sequence M/S_CAI_Courseware

When the M/S_Subsegment_Time_Allowed has expired,
then terminate the segment.

2.4.1.1 LOAD AND DISTRIBUTE MISSION SEGMENT COURSEWARE

DESCRIPTION: Load the mission segment courseware. Distribute the mission segment description to the mission sequencer and the performance criteria to the assess mission segment performance process.

INPUTS: Mission_Item_Reference
Mission_Segment_Courseware_File
Mission_Segment_Repeat_Reference

OUTPUTS: Mission_Segment_Description
Mission_Segment_Performance_Criteria

PSEUDOCODE: Upon receiving either a Mission_Item_Reference
or a Mission_Segment_Repeat_Reference,

Load the associated Mission_Segment_Description
and Mission_Segment_Performance_Criteria
from the Mission_Segment_Courseware_File.

Issue the Mission_Description for mission segment sequencing.

Issue the Mission_Performance_Criteria for mission assessment.

2.4.1.2 SEQUENCE MISSION SEGMENT COURSEWARE

DESCRIPTION: Control the mission segment sequence. Note that a segment sequence can consist of a repeated segment, or a sequence of segments of increasing cardinality.

INPUTS: Mission_Segment_Description

OUTPUTS: Mission_Segment_CAI_Courseware
Launch_Segment_Reference
Cruise_Segment_Reference
Target_Area_Segment_Reference
Lock-on_Segment_Reference
Impact_Segment_Reference

PSEUDOCODE: Issue the Mission_Segment_CAI_Reference to begin the applicable CAI_Type (pre-mission CAI).

Beginning with the first segment in the Mission_Segment_Description and ending with the last segment in the list, proceed sequentially through the following segment cases making sure that the cardinality of any segment does not that of it's predecessor:

- CASE 1 (Launch_Segment),
Issue the Launch_Segment_Reference to begin the Launch phase of the mission.
- CASE 2 (Cruise_Segment),
Issue the Cruise_Segment_Reference to begin the Cruise phase of the mission.
- CASE 3 (Target_Segment),
Issue the Target_Area_Segment_Reference to begin the Target_Area phase of the mission.
- CASE 4 (Lock-on_Segment),
Issue the Lock-on_Segment_Reference to begin the Lock-on phase of the mission.
- CASE 5 (Impact_Segment),
Issue the Impact_Segment_Reference to begin the Impact phase of the mission.

2.4.2.1 LOAD AND DISTRIBUTE LAUNCH SUBSEGMENT COURSEWARE

DESCRIPTION: Load the launch segment courseware. Distribute the launch segment description to the launch sequencer and the performance criteria to the assess launch segment performance process.

INPUTS: Launch_Segment_Reference
Launch_Subsegment_Courseware_File
Launch_Segment_Repeat_Reference

OUTPUTS: Launch_Subsegment_Description
Launch_Subsegment_Performance_Criteria

PSEUDOCODE: Upon receiving either a Launch_Segment_Reference or a Launch_Segment_Repeat_Reference,

Load the associated Launch_Subsegment_Courseware from the Launch_Subsegment_Courseware_File.

Issue the Launch_Description for launch subsegment sequencing.

Issue the Launch_Subsegment_Performance_Criteria for launch subsegment performance assessment.

2.4.2.2 SEQUENCE LAUNCH SUBSEGMENT COURSEWARE

DESCRIPTION: Control the launch segment sequence.

INPUTS: Launch_Subsegment_Description

OUTPUTS: Launch_Phase_Courseware
Launch_Phase_Graphover_Courseware

PSEUDOCODE: Issue the Launch_Phase_Courseware to begin
launch training(as run by operational system).

Issue the Launch_Phase_Graphover Courseware to the
graphics controller for any specialized launch related
training screen overlays.

2.4.3.1 LOAD AND DISTRIBUTE CRUISE SUBSEGMENT COURSEWARE

DESCRIPTION: Load the cruise segment courseware. Distribute the cruise segment description to the cruise sequencer and the performance criteria to the assess cruise segment performance process.

INPUTS: Cruise_Segment_Reference
Cruise_Subsegment_Courseware_File
Cruise_Segment_Repeat_Reference

OUTPUTS: Cruise_Subsegment_Description
Cruise_Subsegment_Performance_Criteria

PSEUDOCODE: Upon receiving either a Cruise_Segment_Reference
or a Cruise_Segment_Repeat_Reference,

Load the associated Cruise_Subsegment_Description
and Cruise_Performance_Criteria
from the Cruise_Subsegment_Courseware_File.

Issue the Cruise_Description for cruise subsegment
sequencing.

Issue the Cruise_Subsegment_Performance_Criteria for
cruise subsegment performance assessment.

2.4.3.2 SEQUENCE CRUISE SUBSEGMENT COURSEWARE

DESCRIPTION: Control the cruise subsegment sequence.

INPUTS: Cruise_Subsegment_Description

OUTPUTS: Cruise_Phase_Courseware
Cruise_Phase_Graphover_Courseware

PSEUDOCODE: If the Cruise_Phase_Courseware contains a Simulation_State_Vector, issue it to the Simulator to either begin a cruise phase missile/seeker simulation or to alter the missile/seeker control state.

Issue the Cruise_Parameter_Update to change a flight or navigational parameter state.

Issue the Cruise_Phase_Graphover_Courseware to prompt the gunner to restore the altered parameter to its initial state.

When the time of the cruise segment > Cruise_Phase_Time
relenquish control of the simulation to the mission
segment supervisor.

2.4.4.1 LOAD AND DISTRIBUTE TARGET_AREA SUBSEGMENT COURSEWARE

DESCRIPTION: Load the target area segment courseware. Distribute the target_area segment description to the target_area sequencer and the performance criteria to the assess target_area segment performance process.

INPUTS: Target_Area_Segment_Reference
Target_Area_Subsegment_Courseware_File
Target_Area_Segment_Repeat_Reference

OUTPUTS: Target_Area_Subsegment_Description
Target_Area_Subsegment_Performance_Criteria

PSEUDOCODE: Upon receiving either a Target_Area_Segment_Reference or a Target_Area_Segment_Repeat_Reference,

Load the associated Target_Area_Subsegment_Description and the Target_Area_Performance_Criteria from the Target_Area_Subsegment_Courseware_File.

Issue the Target_Area_Description for Target_Area subsegment sequencing.

Issue the Target_Area_Subsegment_Performance_Criteria for cruise subsegment performance assessment.

2.4.4.2 SEQUENCE TARGET_AREA SEGMENT COURSEWARE

DESCRIPTION: Control the target_area segment sequence.

INPUTS: Target_Area_Subsegment_Description

OUTPUTS: Target_Area_Phase_Courseware
Target_Area_Phase_Graphover_Courseware

PSEUDOCODE: If the Target_Area_Phase_Courseware contains a Simulation_State_Vector, issue it to the Simulator to either begin a target area phase missile/seeker simulation or to alter the missile/seeker control state.

Issue the Target_Area_Phase_Graphover_Courseware to prompt the gunner to restore the altered parameter to its initial state.

When the time of target area subsegment > Target_Area_Phase_Time, relenquish control of the simulation to the mission segment supervisor.

2.4.5.1 LOAD AND DISTRIBUTE LOCK-ON SUBSEGMENT COURSEWARE

DESCRIPTION: Load the lock-on segment courseware. Distribute the lock-on segment description to the lock-on sequencer and the performance criteria to the assess lock-on segment performance process.

INPUTS: Lock-on_Segment_Reference
Lock-on_Subsegment_Courseware_File
Lock-on_Segment_Repeat_Reference

OUTPUTS: Lock-on_Subsegment_Description
Lock-on_Subsegment_Performance_Criteria

PSEUDOCODE: Upon receiving either a Lock-on_Segment_Reference or a Lock-on_Segment_Repeat_Reference,

Load the associated Lock-on_Subsegment_Description and the Lock-on_Performance_Criteria from the Lock-on_Subsegment_Courseware_File.

Issue the Lock-on_Description for Lock-on subsegment sequencing.

Issue the Lock-on_Subsegment_Performance_Criteria for cruise subsegment performance assessment.

2.4.5.2 SEQUENCE LOCK-ON SUBSEGMENT COURSEWARE

DESCRIPTION: Control the lock-on segment sequence.

INPUTS: Lock-on_Subsegment_Description

OUTPUTS: Lock-on_Phase_Courseware
Lock-on_Phase_Graphover_Courseware

PSEUDOCODE: If the Lock-on_Phase_Courseware contains a Simulation_State_Vector, issue it to the Simulator to either begin a lock-on phase missile/seeker simulation or to alter the missile/seeker control state.

Issue the Lock-on_Phase_Graphover_Courseware to prompt the gunner to restore the altered parameter to its initial state.

When the time of the lock-on segment > Lock-on_Phase_Time
relenquish control of the simulation to the mission
segment supervisor.

2.4.6.1 LOAD AND DISTRIBUTE IMPACT SUBSEGMENT COURSEWARE

DESCRIPTION: Load the impact segment courseware. Distribute the impact segment description to the impact sequencer and the performance criteria to the assess impact segment performance process.

INPUTS: Impact_Segment_Reference
Impact_Subsegment_Courseware_File
Impact_Segment_Repeat_Reference

OUTPUTS: Impact_Subsegment_Description
Impact_Subsegment_Performance_Criteria

PSEUDOCODE: Upon receiving either a Impact_Segment_Reference
or a Impact_Segment_Repeat_Reference,

Load the associated Impact_Subsegment_Description
and the Impact_Performance_Criteria
from the Impact_Subsegment_Courseware_File.

Issue the Impact_Description for Impact subsegment
sequencing.

Issue the Impact_Subsegment_Performance_Criteria for
cruise subsegment performance assessment.

2.4.6.2 SEQUENCE IMPACT SEGMENT COURSEWARE

DESCRIPTION: Control the impact segment sequence.

INPUTS: Impact_Subsegment_Description

OUTPUTS: Impact_Phase_Courseware
Impact_Phase_Graphover_Courseware

PSEUDOCODE: If the Impact_Phase_Courseware contains a Simulation_State_Vector, issue it to the Simulator to either begin a impact phase missile/seeker simulation or to alter the missile/seeker control state.

Issue the Impact_Phase_Graphover_Courseware to prompt the gunner to restore the altered parameter to its initial state.

When the time of the impact segment > Impact_Phase_Time
relenquish control of the simulation to the mission
segment supervisor.

3.1 RECORD ITEM PERFORMANCE

DESCRIPTION: Update item performance record, if required.

INPUTS: Item_Performance_Criteria
M/C_Segment_Result
P/D_Segment_Result
Mission_Segment_Result

OUTPUTS: Item_Grade
Performance_Records

PSEUDOCODE: Update performance records only if the
Item_Performance_Criteria contains the Item_ID.

If the Segment_Result = passed then
Mark the Performance_Record as passed.

Otherwise, Segment_Result = failed,
Mark the Item_Performance_Record as failed.

Item_Grade = Segment_Result.

3.2.1.1 COMPARE M/C KEYPAD ANSWER

DESCRIPTION: Compare the gunner's keypad response with the correct M/C_Keypad_Answer to determine if the gunner answered correctly. Immediate feedback consist of a correct message, a retry prompt, an incorrect message or a time expired message.

INPUT: M/C_Keypad_Results
M/C_Keypad_Answer

OUTPUT: M/C_Keypad_Immediate_Feedback
M/C_Subsegment_Results

PSEUDOCODE: Repeat
 If the time allowed has not expired, then
 Upon recieving an M/C_Keypad_Result
 Compare it to the corresponding M/C_Keypad_Answer.
 CASE 1: Gunner's first attempt
 If correct,
 Issue M/C_Keypad_Immediate_Feedback correct message,
 Issue M/C_Subsegment_Result correct indicator
 Else
 Issue M/C_Keypad_Immediate_Feedback retry prompt
 CASE 2: Gunner's second attempt
 If correct,
 Issue M/C_Subsegment_Results correct indicator,
 Issue M/C_Keypad_Immediate_Feedback correct message
 Else
 Issue M/C_Subsegment_Result incorrect indicator,
 Issue M/C_Keypad_Immediate_Feedback incorrect message
 Else
 Issue M/C_Keypad_Immediate_Feedback time expired message,
 Issue M/C_Subsegment_Results incorrect indicator
 Until there are no more M/C_Keypad_Results

3.2.1.2 COMPARE M/C PDP ANSWER

DESCRIPTION: Compare the gunner's PDP response with the correct M/C_PDP_Answer to determine if the gunner answered correctly. M/C_PDP_Immediate_Feedback consists of a correct message, a retry prompt, an incorrect message, a time expired message, or optionally PDP_Feedback (such as lighting or flashing of PDP's).

INPUT: M/C_PDP_Results
M/C_PDP_Answer

OUTPUT: M/C_PDP_Immediate_Feedback
M/C_Subsegment_Results

PSEUDOCODE: Repeat
 If the time allowed has not expired, then
 Upon receiving an M/C_PDP_Result
 Compare it to the corresponding M/C_PDP_Answer.

 CASE 1: Gunner's first attempt
 If correct,
 Issue M/C_PDP_Immediate_Feedback correct message,
 Issue M/C_Subsegment_Result correct indicator
 Else
 Issue M/C_PDP_Immediate_Feedback retry prompt

 CASE 2: Gunner's second attempt
 If correct,
 Issue M/C_Subsegment_Results correct indicator,
 Issue M/C_PDP_Immediate_Feedback correct message
 Else
 Issue M/C_Subsegment_Result incorrect indicator,
 Issue M/C_PDP_Immediate_Feedback incorrect message

 Else
 Issue M/C_PDP_Immediate_Feedback time expired message,
 Issue M/C_Subsegment_Results incorrect indicator
 Until there are no more M/C_PDP_Results

3.2.2.1 TABULATE M/C SEGMENT

DESCRIPTION: Determine the number of multiple choice questions answered correctly. If a number was answered incorrectly on a retry attempt, then record a question identifier.

INPUT: M/C_Number_Of_Questions
M/C_Segment_Results

OUTPUT: M/C_Subsegment_Summary

PSEUDOCODE: Upon receiving the number of questions

Repeat

If correct:

Increment a counter of the number correct

Else

Save an identifier of the subsegment incorrectly answered.

Decrement the number of questions

Until M/C_Number_Of_Questions = 0.

Issue M/C_Segment_Summary

3.2.2.2 EVALUATE M/C SEGMENT

DESCRIPTION: Determine if the segment is passed or failed and take the appropriate action. If the lesson was failed, a list of subsegment references are released to indicate questions to be repeated.

INPUT: M/C_Expected_Number_Correct
M/C_Segment_Summary

OUTPUT: M/C_Segment_Results
M/C_Repeat_Subsegment_List
M/C_Segment_Performance_Feedback

PSEUDOCODE: Upon receiving the M/C_Expected_Number_Correct
and the M/C_Segment_Summary,

 If M/C_Number_Correct > M/C_Expected_Number_Correct

 Issue M/C_Segment_Performance_Feedback pass message
 Issue M/C_Segment_Results pass indicator

 Else

 Issue M/C_Segment_Performance_Feedback fail message
 Issue M/C_Segment_Results fail indicator
 Issue M/C_Repeat_Subsegment_List

3.3.1.1. COMPARE M/S TARGET RESPONSE

DESCRIPTION: Determine if the target marked was the target desired.

INPUT: M/S_Results
M/S_Desired_Target_Location

OUTPUT: M/S_Subsegment_Elapsed_Time
M/S_Target_Missed_Target_Result
M/S_Missed_Target_Results

PSEUDOCODE: Upon receiving the M/S_Desired_Target_Location,
Reset and start the M/S_Subsegment_Elapsed_Time
Upon receiving an M/S_Result
Stop and reset the Subsegment_Elapsed_Time
If M/S_Result is within the Target_Radius of the
M/S_Desired_Target_Location,
Issue M/S_Target_Marked
Else
Issue M/S_Missed_Target_Result

3.3.1.2 ISSUE M/S TARGET LOCATIONS

DESCRIPTION: Issue the location of the prompted target for initial comparison to the gunner's choice, and issue a list of undesired targets for comparison if the initial check fails.

INPUT: M/S_Target_Locations
M/S_Desired_Target_ID

OUTPUT: M/S_Desired_Target_Location
M/S_Undesired_Targets

PSEUDOCODE: Upon receiving a list of M/S_Target_Locations and an M/S_Desired_Target_ID,

Issue the M/S_Desired_Target_Location corresponding to the M/S_Desired_Target_ID.

Issue a list of the remaining targets in case a secondary comparison of results is necessary.

3.3.1.3 COMPARE M/S TARGET TO OTHER TARGETS

DESCRIPTION: If the desired target has not been marked, the check to see if other possible targets were marked.

INPUT: M/S_Missed_Target_Result
M/S_Undesired_Targets

OUTPUT: M/S_Incorrect_Target_Marked
M/S_Target_Missed

PSEUDOCODE: Upon receiving an M/S_Missed_Target_Result

Repeat

Compare it to each of the M/S_Undesired_Targets

Until a hit occurs or all targets have been checked.

If the M/S_Missed_Target_Result is within the target
radius of an M/S_Undesired_Target

Issue an M/S_Incorrect_Target_Marked

Else

Issue an M/S_Target_Missed

3.3.1.4 ISSUE M/S IMMEDIATE FEEDBACK AND RESULTS

DESCRIPTION: Immediate feedback is issued according to whether a desired target was hit, an undesired target was hit, or all targets were missed.

INPUT: M/S_Target_Marked
M/S_Incorrect_Target_Marked
M/S_Target_Missed

OUTPUT: M/S_Subsegment_Results
M/S_Immediate_Feedback

PSEUDOCODE: Repeat
 Upon receiving an M/S_Target_Marked
 CASE 1: First attempt
 Issue an M/S_Immediate_Feedback hit message
 Issue an M/S_Subsegment_Result hit indicator
 CASE 2: Second attempt
 Issue an M/S_Immediate_Feedback hit message
 Issue an M/S_Subsegment_Result retry hit indicator
 Upon receiving an M/S_Target_Missed
 CASE 1: First attempt
 Issue an M/S_Immediate_Feedback retry prompt
 CASE 2: Second attempt
 Issue an M/S_Immediate_Feedback miss message
 Issue an M/S_Subsegment_Result missed indicator
 Upon receiving an M/S_Incorrect_Target_Marked
 CASE 1: First attempt
 Issue an M/S_Immediate_Feedback retry prompt
 CASE 2: Second attempt
 Issue an M/S_Immediate_Feedback miss message
 Issue an M/S_Subsegment_Result missed indicator

 Until no more subsegments (targets)

3.3.2.1 TABULATE P/D - M/S SEGMENT

DESCRIPTION: Compile a segment summary consisting of the number of correct responses and a list of specific responses that were wrong. The segment summary is issued after the last M/S_Subsegment_Result has been recieved.

INPUT: M/S_Subsegment_Time_Allowed
M/S_Expected_Number_Targets
M/S_Subsegment_Results
M.S_Subsegment_Elapsed_Time

OUTPUT: M/S_Segment_Summary

PSEUDOCODE: Upon receiving an M/S_Subsegment_Result,

 If the target was hit, increment a hit-target counter

 If the target was missed, increment a missed-target counter

 If the target was hit on the second attempt, increment a retry-hit counter

 if the M/S_Elapsed_Time > M/S_Time_Allowed, increment a time-limit-exceeded counter

When the segment is complete, issue an M/S_Segment_Summary containing the results of each of the counters.

3.3.2.2 EVALUATE P/D - M/S SEGMENT

DESCRIPTION: Determine if the segment was passed or failed, and take the appropriate action. If the lesson was failed, issue a segment repeat reference so that the same segment or a similar segment may be sequenced. The information in the segment summary is displayed in the feedback messages.

INPUT: M/S_Segment_Summary
M/S_Expected_Number_Correct

OUTPUT: M/S_Segment_Performance_Feedback
M/S_Segment_Results
M/S_Repeat_Segment_References

PSEUDOCODE: If M/S_Number_Correct < M/S_Expected_Number_Correct

Issue an M/S_Segment_Performance_Feedback failed message
Issue an M/S_Repeat_Segment_Reference
Issue an M/S_Segment_Result failed indicator

Else

Issue an M/S_Segment_Performance_Feedback passed message
Issue an M/S_Segment_Result passed indicator

3.3.3.1 COMPARE S/D TARGET RESPONSE

DESCRIPTION: When the segment begins, start a clock to record the elapsed time of the subsegment. Keep track of the time in which the cross hairs were centered on the target. Stop timing and reset to zero whenever the cross hairs stray from the figure, and issue a tracking time.

INPUT: S/D_Results
S/D_Target_Location

OUTPUT: S/D_Subsegment_Elapsed_Time
S/D_Tracking_Time

PSEUDOCODE: Upon receiving an S/D_Target_Location,
 Initialize and start S/D_Subsegment_Elapsed_Time
 Repeat
 When the S/D_Result is centered within the S/D_Target_Location then start the S/D_Tracking_Time.
 If the S/D_Result strays from the geometric figure,
 Stop and issue the S/D_Tracking_Time
 Reset the S/D_Tracking_Time to zero.
 Until there are no more S/D_Target_Locations

3.3.3.2 COMPARE_S/D_TARGET_RESPONSE_TIME

DESCRIPTION: Compare each S/D_Target_Tracking_Time to the expected tracking time to see if greater. With each comparison, the longest period of tracking is saved.

INPUT: S/D_Tracking_Time
S/D_Expected_Tracking_Time

OUTPUT: S/D_Longest_Tracking_Time

PSEUDOCODE: Repeat

 Upon receiving an S/D_Tracking_Time,

 If it is the first, save it for comparison to other tracking times.

 If it is not, compare it to the existing saved S/D_Tracking_Time.

 If the new S/D_Tracking_Time > saved S/D_Tracking_Time

 Replace the saved S/D_Tracking_Time with the new S/D_Tracking_Time.

 Else

 Discard the new S/D_Tracking_Time

Until there are no more S/D_Tracking_Times

If the saved S/D_Tracking_Time > S/D_Expected_Tracking_Time

 Issue an S/D_Longest_Tracking_Time passed indicator

Else

 Issue an S/D_Longest_Tracking_Time failed indicator

3.3.3.3 SUM OVER ALL TARGET RESPONSE TIME

DESCRIPTION: Sum all tracking times to get a cumulative tracking time.

INPUT: S/D_Tracking_Time

OUTPUT: S/D_Summed_Response_Time

PSEUDOCODE: Initialize S/D_Summed_Response_Time to zero

Repeat

 Upon receiving an S/D_Tracking_Time, add it to the
 contents of S/D_Summed_Response_Time

Until there are no more S/D_Tracking_Times

Issue the S/D_Summed_Response_Time

3.3.3.4 ISSUE S/D IMMEDIATE FEEDBACK AND RESULTS

DESCRIPTION: Issue the appropriate immediate feedback consisting of the cumulative tracking time and the longest tracking time.
Issue the appropriate subsegment results.

INPUT: S/D_Summed_Response_Time
S/D_Longest_Tracking_Time

OUTPUT: S/D_Subsegment_Results
S/D_Immediate_Feedback

PSEUDOCODE: Upon receiving the S/D_Summed_Response_Time and the S/D_Immediate_Feedback,

 If the subsegment was passed,

 Issue an S/D_Immediate_Feedback passed message
 Issue an S/D_Subsegment_Result passed indicator

 Else

 Issue an S/D_Immediate_Feedback failed message
 Issue an S/D_Subsegment_Result failed indicator

3.3.4.1 TABULATE P/D SEGMENT

DESCRIPTION: Compile a segment summary consisting of the number of correct responses. The segment summary is issued after the last S/D_Subsegment_Result has been recieved.

INPUT: S/D_Subsegment_Results
S/D_Subsegment_Elapsed_Time
S/D_Number_Of_Subsegments
S/D_Subsegment_Time_Allowed

OUTPUT: S/D_Segment_Summary

PSEUDOCODE: If S/D_Subsegment_Elapsed_Time > S/D_Subsegment_Time_Allowed
 If S/D_Subsegment_Results passed,
 Increment a subsegment passed counter
 Else
 Increment a subsegment failed counter
 Increment a subsegment number counter
If the subsegment number counter = S/D_Number_Of_Subsegments
 Release the S/D_Subsegment_Counter

3.3.4.2 EVALUATE P/D SEGMENT

DESCRIPTION: Determine whether a segment has been passed or failed and take the appropriate action. If the segment has been passed, issue a segment passed performance feedback and a segment passed indicator. If the segment has been failed, issue a segment failed performance feedback, a segment failed indicator, and a segment repeat reference.

INPUT: S/D_Segment_Summary
S/D_Expected_Number_Correct

OUTPUT: S/D_Segment_Results
S/D_Segment_Performance_Feedback
S/D_Repeat_Segment_Reference

PSEUDOCODE: Upon receiving an S/D_Segment_Summary,

 If number passed < S/D_Expected_Number_Correct

 Issue an S/D_Segment_Results failed indicator
 Issue an S/D_Segment_Performance_Feedback failed message
 Issue an S/D_Repeat_Segment_Reference

 Else

 Issue an S/D_Segment_Results passed indicator
 Issue an S/D_Segment_Performance_Feedback passed message

3.4.1.1 TABULATE MISSION SEGMENT

DESCRIPTION: Tabulate results over all mission subsegments.

INPUTS: Mission_Number_Of_Subsegments
Mission_Performance_Results
Launch_Subsegment_Results
Cruise_Subsegment_Results
Target_Area_Subsegment_Results
Lock-on_Subsegment_Results
Impact_Subsegment_Results

OUTPUTS: Mission_Segment_Summary

PSEUDOCODE: Repeat the following:

 If the Subsegment_Result = Pass, increment the
 Mission_Subsegment_Correct counter.

Until either,
 the number of iterations > Mission_Number_Of_Segments,
 or, the Mission_Performance_Result indicates crash.

Issue the Mission_Segment_Summary consisting of
the Mission_Number_Of_Correct and Mission_Crash_Status.

3.4.1.2 EVALUATE MISSION SEGMENT COURSEWARE

DESCRIPTION: This process evaluates the mission results by comparing the expected number of correct subsegments to the actual number of correct subsegments.

INPUTS: Mission_Segment_Summary
Mission_Expected_Number_Correct_Subsegments

OUTPUTS: Mission_Segment_Feedback
Mission_Segment_Results
Mission_Repeat_Segment_Reference

PSEUDOCODE: If the Mission_Crash_Status is positive or
the Mission_Subsegments_Correct < Mission_Expected_Number
Correct then,

Issue the Mission_Segment_Feedback informing the gunner
of a failed mission.

If applicable, issue the Mission_Segment_Repeat_Reference,

Otherwise, issue Mission_Segment_Results = Fail.

Otherwise,

Issue the Mission_Segment_Feedback informing the gunner
of a successful mission.

Issue Mission_Segment_Results = Pass.

3.4.2.1 COMPARE LAUNCH STATE VECTORS

DESCRIPTION: Compare the desired and resultant launch state vectors

INPUTS: Launch_Result_State_Vector
Launch_Desired_State_Vector

OUTPUTS: Launch_Results

PSEUDOCODE: Upon receiving the Launch_Desired_State_Vector
begin the Launch_Elapsed_Time.

Upon receiving the Launch_Result_State_Vector
(due to launch or training timeout)
stop the Launch_Elapsed_Time.

Determine the Launch_State_Vector_Difference by comparing
the Launch_Result_State_Vector to the
Launch_Desired_State_Vector

Issue the Launch_State_Vector_Difference and the
Launch_Elapsed_Time for launch subsegment evaluation.

3.4.2.2 EVALUATE LAUNCH SUBSEGMENT

DESCRIPTION: Evaluate the launch state difference and determine if all parameter fields are within the allowable tolerance.

INPUTS: Launch_Subsegment_Criteria
Launch_Results

OUTPUTS: Launch_Subsegment_Results
Launch_Segment_Repeat_Reference

PSEUDOCODE: If the Launch_State_Vector_Difference < Launch_Error_Tolerance
and the Launch_Elapsed_Time < Launch_Desired_Response_Time:

Issue Launch_Immediate_Feedback informing the
gunner of Launch_Elapsed_Time, Launch_State_Vector_
Difference and that the segment was passed.

Otherwise,

Issue Launch_Parameter_Immediate_Feedback informing the
gunner that the segment was failed.

If applicable, issue the Launch_Segment_Repeat_Reference.

3.4.3.1 COMPARE CRUISE PARAMETERS

DESCRIPTION: Compare the resultant parameter value with the desired parameter value.

INPUTS: Cruise_Parameter_Performance_Result
Cruise_Parameter_Desired_Result

OUTPUTS: Cruise_Parameter_Immediate_Feedback
Cruise_Parameter_Results

PSEUDOCODE: Upon receiving the Cruise_Parameter_Desired_Result
begin the Cruise_Elapsed_Time.

Upon receiving the Cruise_Parameter_Performance_Result
(due to cruise training timeout)
stop the Cruise_Elapsed_Time.

Determine the Cruise_Parameter_Difference by comparing
the Cruise_Parameter_Performance_Result to the
Cruise_Parameter_Desired_Result

Issue the Cruise_Parameter_Difference and the
Cruise_Elapsed_Time for cruise subsegment evaluation.

3.4.3.2 EVALUATE CRUISE SUBSEGMENT

DESCRIPTION: Evaluate the cruise subsegment by determining if the cruise parameter state difference and the elapsed subsegment time are within the allowed tollerances.

INPUTS: Cruise_Subsegment_Criteria
Cruise_Parameter_Results

OUTPUTS: Cruise_Subsegment_Results
Cruise_Segment_Repeat_Reference

PSEUDOCODE: If the Cruise_Parameter_Difference < Cruise_Parameter_Error
and the Cruise_Elapsed_Time < Cruise_Desired_Response_Time:

Issue Cruise_Parameter_Immediate_Feedback informing the
gunner of Cruise_Elapsed_Time, Cruise_Parameter_
Difference and that the segment was passed.

Otherwise,

Issue Cruise_Parameter_Immediate_Feedback informing the
gunner that the segment was failed.

If applicable, issue the Cruise_Segment_Repeat_Reference.

3.4.4.1 COMPARE TARGET AREA SUBSEGMENT COURSEWARE

DESCRIPTION: Compare the coordinates of the missile to the coordinates of the center of the target area and issue the results for evaluation.

INPUTS: Target_Area_Result_Coordinates
Target_Area_Desired_Coordinates

OUTPUTS: Target_Area_Immediate_Feedback
Target_Area_Results

PSEUDOCODE: Upon receiving the Target_Area_Desired_Coordinates,
begin the Target_Area_Elapsed_Time.

Upon receiving the Target_Area_Result_Coordinates,
(due to gunner's target area training response)
stop the Target_Area_Elapsed_Time.

Calculate the Target_Area_Distance as the distance
between the Target_Area_Result_Coordinates and the
Target_Area_Desired_Coordinates.

Issue the Target_Area_Distance and Target_Area_Elapsed_Time
for target area subsegment evaluation.

3.4.4.2 EVALUATE TARGET AREA SUBSEGMENT

DESCRIPTION: Evaluate the target area subsegment by determining if the missile distance from the center of the target area and the elapsed time are within the allowed tolerances.

INPUTS: Target_Area_Performance_Criteria
Target_Area_Results

OUTPUTS: Target_Area_Subsegment_Results
Target_Area_Segment_Repeat_Reference

PSEUDOCODE: If the Target_Area_Distance < Target_Area_Radius and
the Target_Area_Elapsed_Time < Target_Area_Desired_
Response_Time

Issue Target_Area_Immediate_Feedback informing the
gunner that the segment was passed.

Issue the Target_Area_Subsegment_Result as passed.

Otherwise,

Issue the Target_Area_Subsegment_Result as failed

Issue the Target_Area_Immediate_Feedback informing
the gunner that the segment was failed.

If appropriate, issue the Target_Area_Segment_
Repeat_Reference.

3.4.5.1 COMPARE LOCK-ON COORDINATES

DESCRIPTION: Compare the resultant and desired lock-on target Coordinates and issue the results to be evaluated.

INPUTS: Lock-on_Projected_Seeker_Coordinates
Lock-on_Desired_Target_Coordinates

OUTPUTS: Lock-on_Immediate_Feedback
Lock-on_Results

PSEUDOCODE: Upon receiving the Lock-on_Desired_Coordinates,
begin the Lock-on_Elapsed_Time.

Upon receiving the Lock-on_Projected_Seeker_Coordinates,
(due to simulated missile Lock-on)
stop the Lock-on_Elapsed_Time.

Determine the Lock-on_Miss_Distance by comparing the
Lock-on_Projected_Seeker_Coordinates to the
Lock-on_Desired_Target_Coordinates

Issue the Lock-on_Miss_Distance ,Lock-on_Time , and the
for lock-on evaluation.

3.4.5.2 EVALUATE LOCK-ON

DESCRIPTION: Evaluate the lock-on subsegment with respect to its performance criteria.

INPUTS: Lock-on_Subsegment_Criteria
Lock-on_Results

OUTPUTS: Lock-on_Subsegment_Results
Lock-on_Immediate_Feedback
Lock-on_Error_Radius

PSEUDOCODE: If the Lock-on_Miss_Distance < Lock-on_Error_Radius and the Lock-on_Time < Lock-on_Max_Time, then

Issue the Lock-on_Marked_Result as passed.

Otherwise,

Issue the Lock-on_Subsegment_Result as failed

Issue the Lock-on_Immediate_Feedback informing the gunner that the segment was failed.

If appropriate, issue the Lock-on_Segment_Repeat_Reference.

Issue the Lock-on_Error_Radius so that the coordinates can be compared to other target locations.

3.4.5.3 COMPARE ALL COORDINATES

DESCRIPTION: Compare the resultant target location to all other target locations and issue the results to be evaluated.

INPUTS: Lock-on_Target_Coordinates
Lock-on_Desired_Coordinates
Lock-on_Error_Radius

OUTPUTS: Lock-on_Immediate_Feedback

PSEUDOCODE: Determine the Lock-on_Miss_Distance by comparing the Lock-on_Projected_Seeker_Location to the Lock-on_Target_Locations.

If the Lock-on_Miss_Distance is within the Lock-on_Error_Radius of any of the other targets,

Issue Lock-on_Immediate_Feedback informing the gunner that the incorrect target was locked onto.

3.4.6.1 COMPARE IMPACT LOCATIONS

DESCRIPTION: Compare the resultant and desired impact locations and issue the results to be evaluated.

INPUTS: Impact_Result_Coordinates
Impact_Desired_Coordinates

OUTPUTS: Impact_Results

PSEUDOCODE: Upon receiving the Impact_Desired_Coordinates,
begin the Impact_Elapsed_Time.

Upon receiving the Impact_Result_Coordinates,
(due to simulated missile impact)
stop the Impact_Elapsed_Time.

Calculate the Impact_Miss_Distance as the distance between
the Impact_Result_Coordinates and the Impact_Desired_Coordinates.

Issue the Impact_Miss_Distance and Impact_Elapsed_Time for
impact subsegment evaluation.

3.4.6.2 EVALUATE IMPACT SEGMENT

DESCRIPTION: Evaluate the impact subsegment by determining if the distance from impact to the target and the elapsed subsegment time are within the allowed tolerances.

INPUTS: Impact_Subsegment_Criteria
Impact_Results

OUTPUTS: Impact_Subsegment_Results
Impact_Segment_Repeat_Reference

PSEUDOCODE: If the Impact_Miss Distance < Impact_Error_Radius and
the Impact_Elapsed_Time < Impact_Desired_Response_Time then

Issue the Impact_Subsegment_Result as passed.

Issue the Impact_Immediate_Feedback informing
the gunner that the segment was passed.

Otherwise,

Issue the Impact_Subsegment_Result as failed

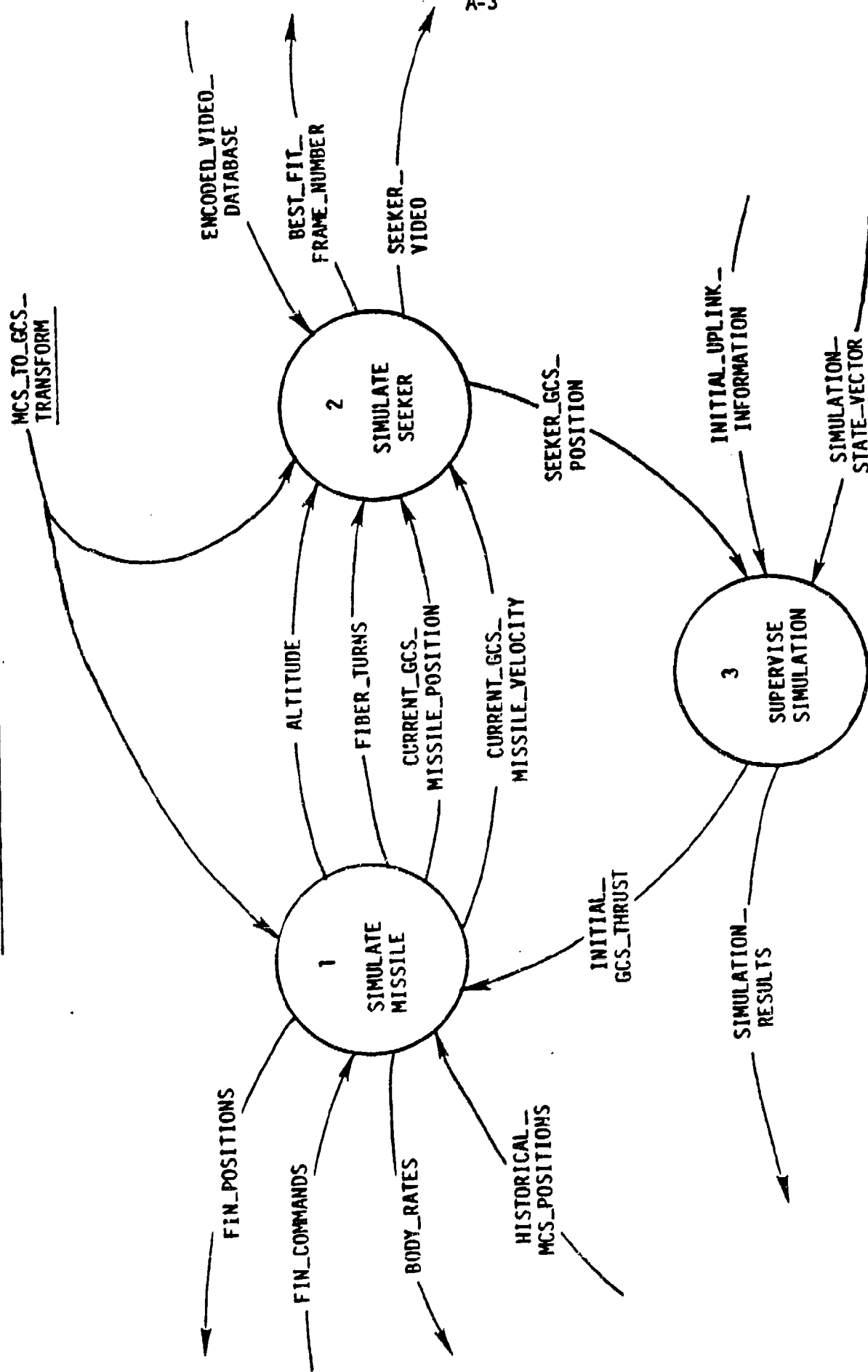
Issue the Impact_Immediate_Feedback informing
the gunner that the segment was failed.

If appropriate, issue the Impact_Segment_
Repeat_Reference.

APPENDIX A: DATA FLOW DIAGRAMS OF VIDEO DISK FLIGHTS SIMULATION

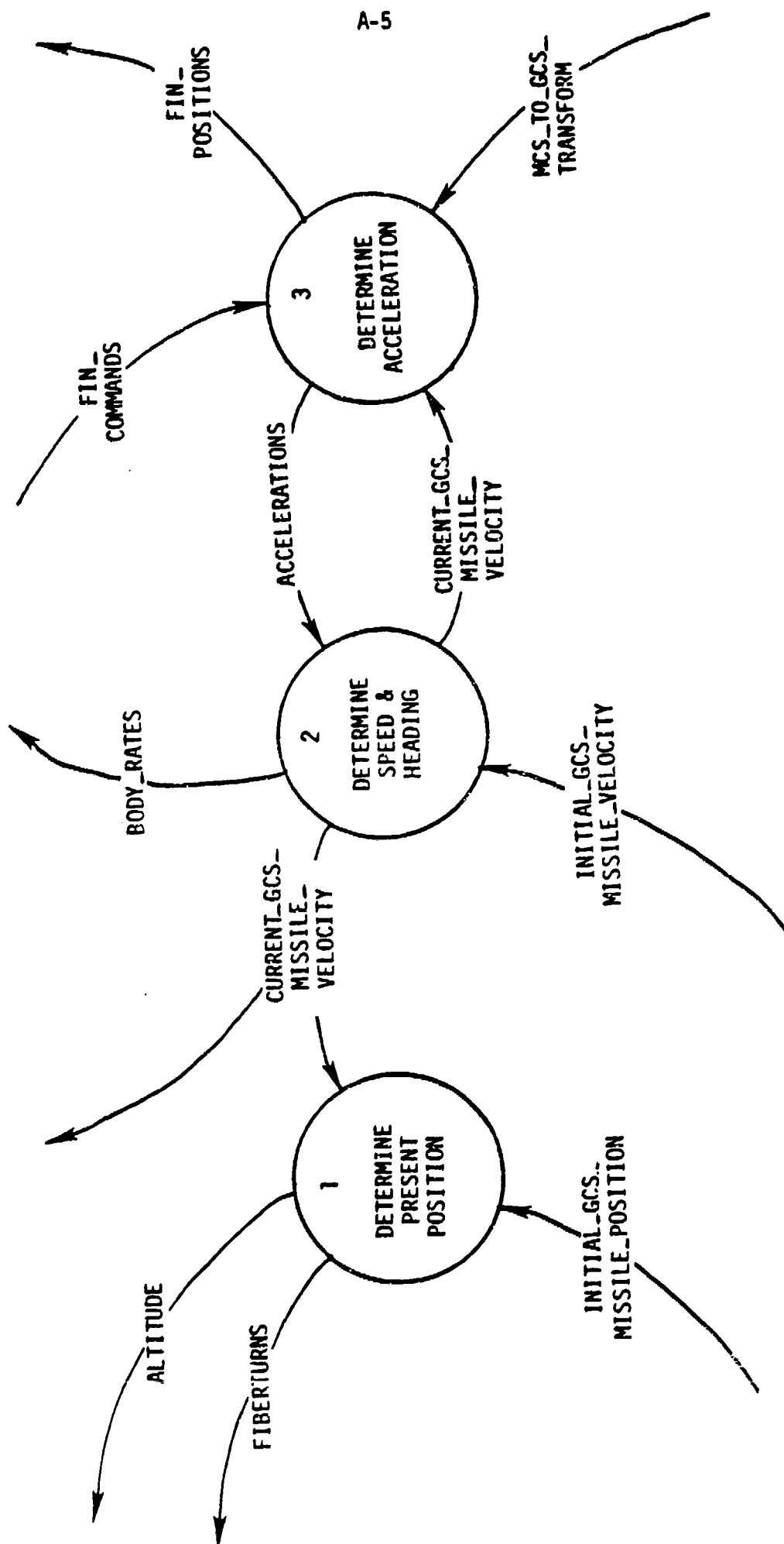
This appendix contains data flow diagrams of processes which directly interface with the FOG-M hardware. Appendix B contains descriptions (i.e., mini-specs) of the primitive processes that involve these data flows.

PROCESS 4.4a VIDEO DISK FLIGHT SIMULATION

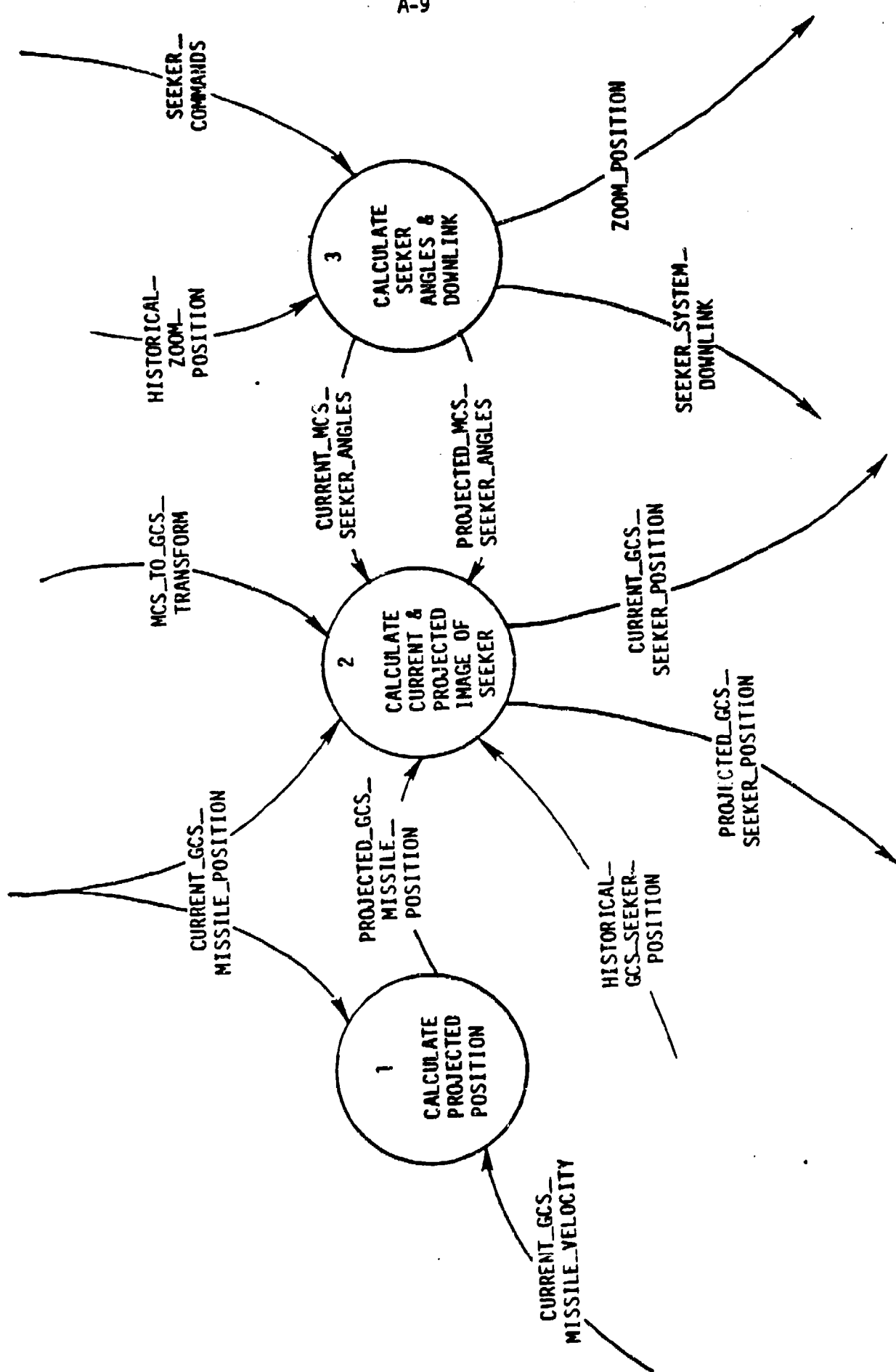


PROCESS 4.4.1: SIMULATE MISSILE

A-5

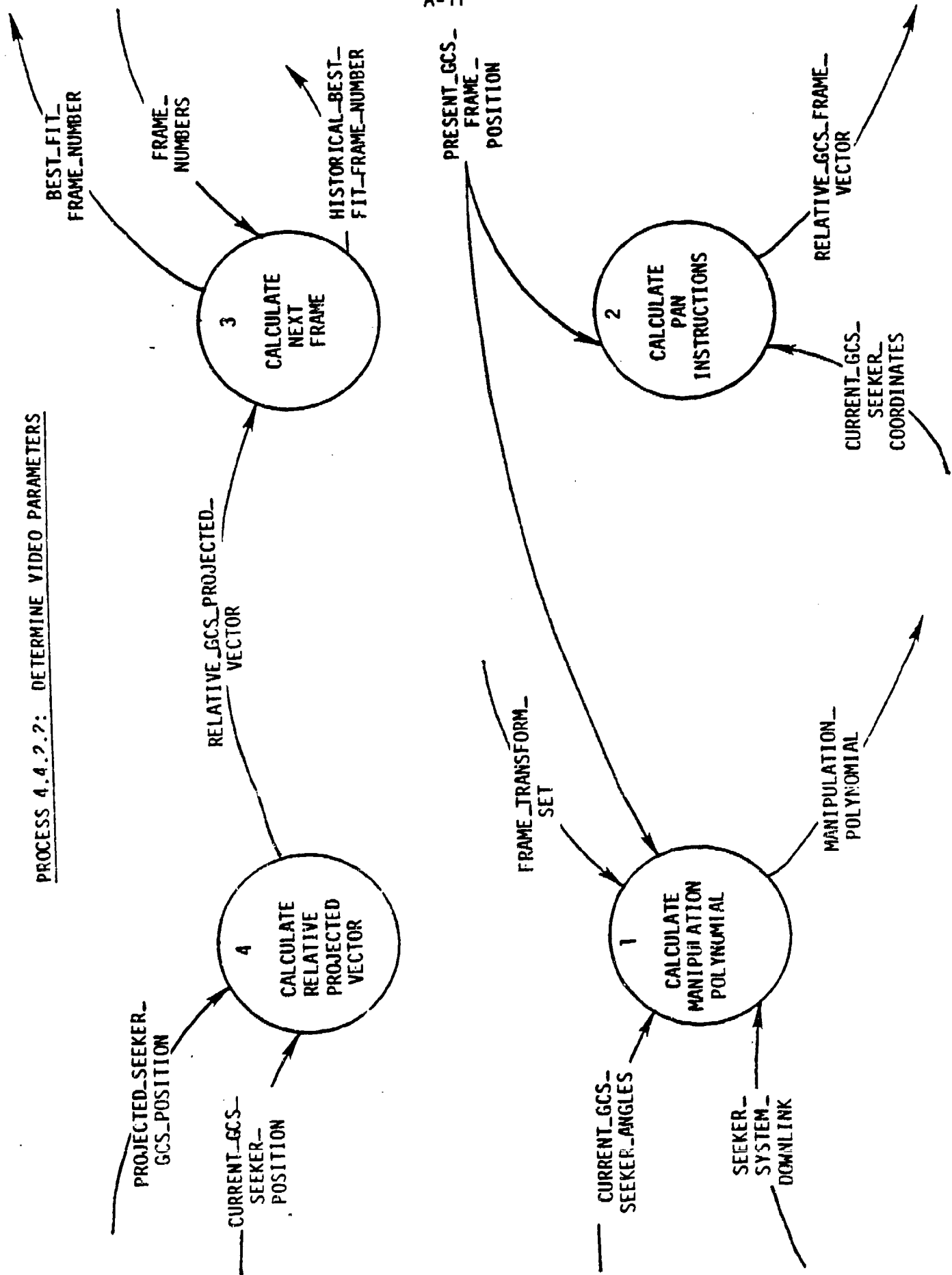


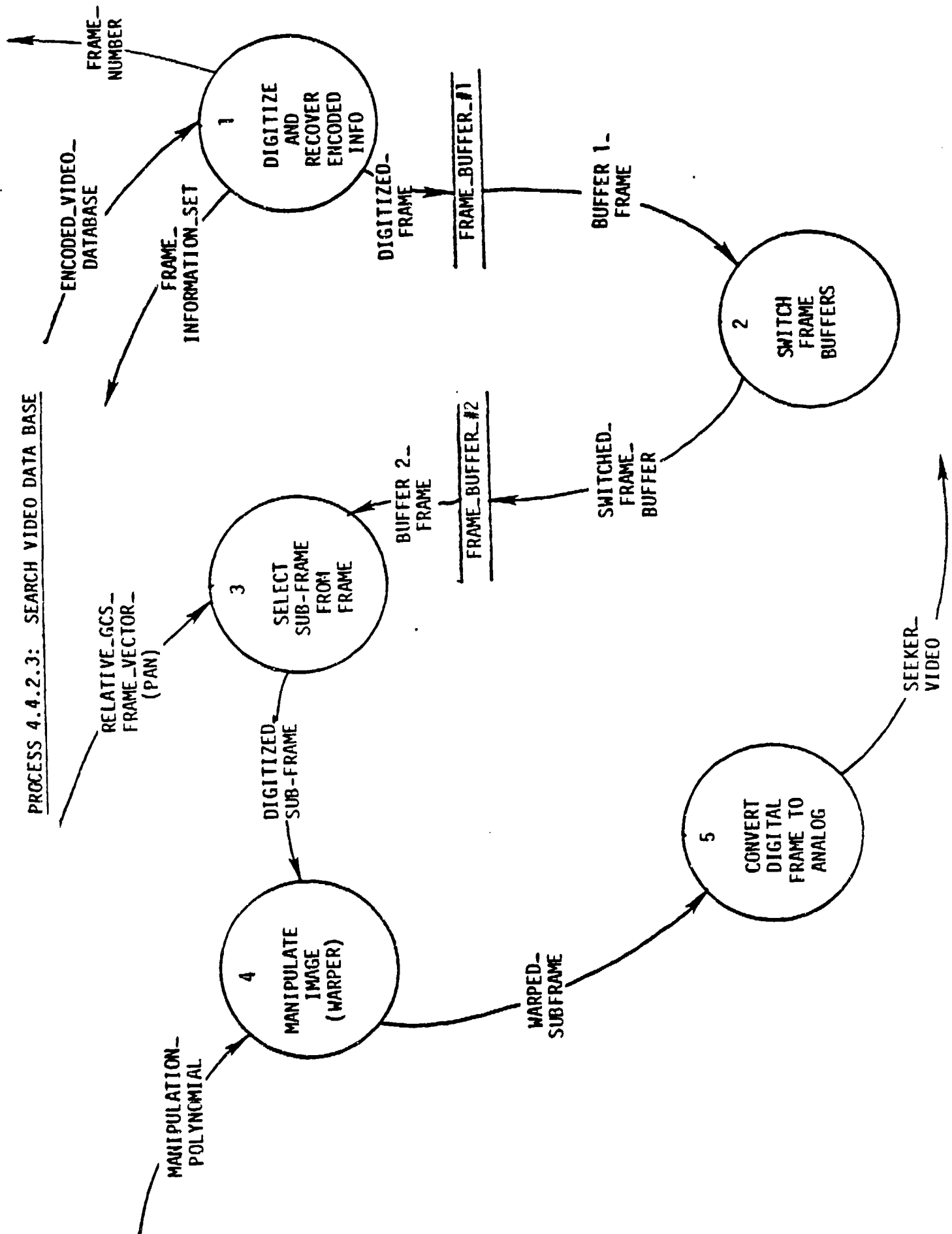
PROCESS 4.4.2.1: CALCULATE SEEKER SIMULATION



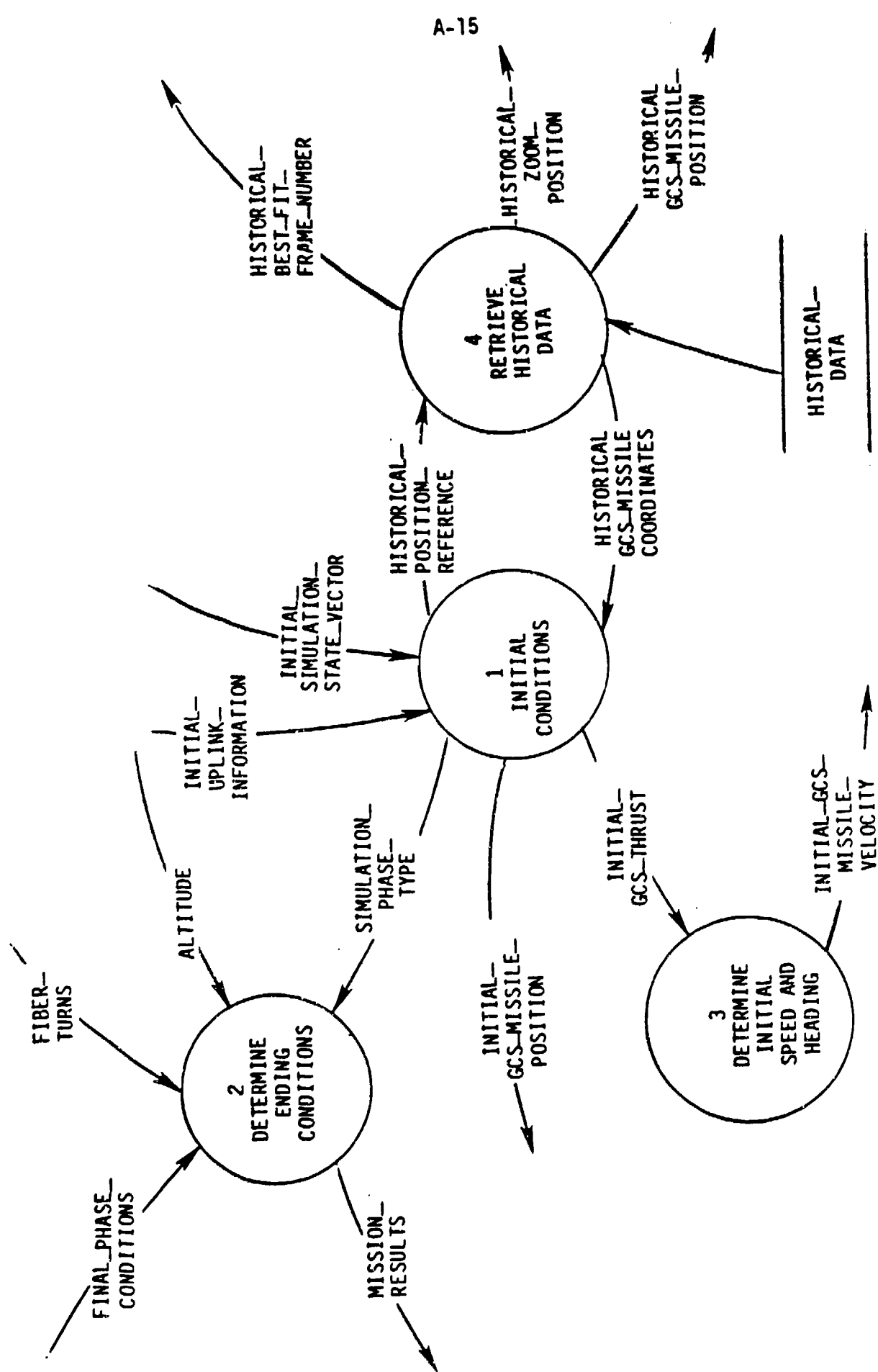
PROCESS 4.4.2.2: DETERMINE VIDEO PARAMETERS

A-11



PROCESS 4.4.2.3: SEARCH VIDEO DATA BASE

PROCESS 4.4.3: SUPERVISE SIMULATION



APPENDIX B: DESCRIPTIONS OF PROCESSES INTERFACING WITH FOG-M HARDWARE

This appendix contains descriptions (i.e., mini-specs) of processes that interface directly with the FOG-M hardware. Another appendix contains data flow diagrams for these processes.

4.1 PERFORM CAI

DESCRIPTION: Sequences instructional or explanatory material to the gunner.

A set of video disks, winchester file references, input rules, and courseware types is input into the process, where type refers to text paging, slide paging, or an audio visual movie.

If the type is a video disk frame, then the frame numbers are retrieved from the video disk file and sent to the video disk player.

If the type consists of text pages from a file, then the contents of the file are loaded and displayed on the screen a page at a time.

If the CAI is a timed frame sequence (movie), then a set of frame numbers is retrieved from the video disk file and sent to the video disk player.

The gunner steps through the CAI material according to the input rules for that courseware type.

INPUTS: CAI Courseware
Video Disk CAI Files
Winchester CAI Files

OUTPUTS: CAI Input Rules
CAI Output

4.2a PERFORM M/C KEYPAD SUBSEGMENT

DESCRIPTION: A multiple choice keypad subsegment consists of a question and a text of possible keypad selections displayed on the CRT.

The process receives a question reference and a set of key enables. The question reference consists of either a video disk frame number or a winchester file reference.

If the question is a frame number, it is sent to the video disk player for display.

If the question is a file reference, then the file loaded and displayed on the CRT.

When the question has been displayed, the appropriate keys are enabled, and the keypad response is released for assessment.

INPUTS: M/C Keypad_Subsegment Courseware
Video_Disk_m/c Keypad_Files
Winchester_M/C_Keypad_Files

OUTPUTS: M/C Keypad_Input Rules
M/C Keypad_Output

4.2b PERFORM M/C PDP SUBSEGMENT

DESCRIPTION: A multiple choice PDP subsegment consist of a question and set of PDP selections displayed on the CRT and the program-able display pushbuttons.

The process receives a question reference and a set of PDP input device enables. A PDP question consists of either a video disk frame number and a winchester file reference, or winchester file references.

If the question contains a video frame reference, it is sent to the video disk player for display. The file reference is accessed for the contents of the PDP displays.

If the question only contains the file references, then a page of text is written on the CRT and PDP text is written to the appropriate PDPs.

When the question has been displayed, the appropriate PDP input devices are enabled, and the response is released for assessment.

INPUTS: M/C_PDP_Subsegment_Courseware
Video_Disk_M/C_PDP_Files
Winchester_M/C_PDP_Files

OUTPUTS: M/C_PDP_Input_Rules
M/C_PDP_Output

4.3a PERFORM S/D TARGET SUBSEGMENT

DESCRIPTION: The S/D target segment is a target tracking practice in which the gunner attempts to keep a set of cross hairs continuously centered on a geometric figure.

The process receives an initial target location and a set of target parameters defining its type, the rates of change of its size and position, and a pointer to a function that controls the motion of the figure.

The geometric figure is then displayed on the screen together with a set of cross hairs and the joystick is enabled. Input from the joystick controls the cross hair movement on the screen.

The process continuously outputs the position of the geometric figure and the cross hair coordinates for assessment.

INPUTS: S/D_Subsegment_Courseware
S/D_Initialization_Parameter

OUTPUTS: SD_Input_Rules
S/D_ResuTts
S/D_Target_Locations

4.3b PERFORM M/S TARGET SUBSEGMENT

DESCRIPTION: The M/S target segment is a particular form of training involving multiple stationary (static) targets displayed on a digital map. The gunner is prompted to locate and mark a specific target on the map.

Upon receiving a map reference and a set of target locations and types, the process displays them on the CRT, together with a set of cross hairs.

A desired target prompt is received and displayed on the screen, and the joystick controls are enabled. Input from the joystick controls the cross hair movement on the screen.

When the joystick trigger is pulled, the location of the cross hair is released for assessment.

INPUTS: M/S_Subsegment_Courseware

OUTPUTS: M/S_Input_Rules
M/S_Result

4.4b PERFORM DPG FLIGHT SIMULATION

DESCRIPTION: This process simulates missile flight parameters used to drive the Digital_Perspective_Generator.

The process receives Mission Courseware to either begin a missile simulation or to inform an ongoing simulation of current phase parameters and changes in control of the missile or seeker.

During a cruise phase, the simulation deviates a navigational or flight parameter from a specified value. The gunner's task is to restore the altered parameter to its initial state. Upon receiving a cruise training response, it issues a Cruise_Parameter_Performance_Result.

During a target area phase, the simulation monitors the missile's position. Upon receiving a target area training response, it issues a Target_Area_Performance_Result.

During a lock-on phase, the simulation continuously calculates the projected seeker coordinates. Upon receiving a simulated missile lock-on (trigger pull), it issues a Target_Area_Performance_Result.

During an impact phase, the simulation returns missile coordinates as Impact_Performance_Results when the missile's altitude equals ground zero.

During all phases, current Altitude and number of Fiber Turns are monitored continuously to determine the occurrence of premature impact or maximum flight distance attained. In either case, Mission Performance Results are issued reflecting the manner in which the simulation terminated.

INPUT: Cruise_Phase Courseware
Target_Area Courseware
Lock-on Phase Courseware
Impact_Phase Courseware

OUTPUT: DPG Updates
Cruise_Parameter_Performance_Result
Target_Area_Performance_Result
Lock-on_Performance_Result
Impact_Performance_Result
Target_Vector

4.4.1.1 DETERMINE PRESENT POSITION

DESCRIPTION: Current GCS Missile Position is determined by integrating Current GCS Missile Velocity over an incremental period of time and adding this result to previous missile position. Fiber turns are determined by dividing the length of missile flight by the circumference of the spindle. Upon initiation, Initial GCS Missile Position provides the starting point for position integration.

INPUTS: Initial GCS Missile Position
Current GCS Missile Velocity

OUTPUTS: Altitude
Current GCS Missile Position
Fiber Turns

4.4.1.2 DETERMINE SPEED & HEADING

DESCRIPTION: The Current Missile Velocity is determined by updating the previous missile velocity by the integration of Acceleration over a time interval. Body_Rates are determined by angular accelerations.

INPUTS: Accelerations
Initial_GCS_Missile_Velocity

OUTPUTS: Current_GCS_Missile_Velocity
Body_Rates

4.4.1.3 DETERMINE ACCELERATION

DESCRIPTION: Fin positions are determined by integration of Fin Commands. Forces and moments are determined from Previous Missile Velocity in missile coordinates and Fin Positions. Accelerations are determined from these forces and moments. Acceleration is transformed from missile coordinates to gunner coordinates.

INPUTS: Previous GCS Missile Velocity
MCS To GCS Transform
Fin Commands

OUTPUTS: Acceleration
Fin Positions

4.4.2.1.1 CALCULATE PROJECTED POSITION

DESCRIPTION: This process calculates Projected GCS Missile Position by integrating Current GCS Missile Velocity over a stated time interval and adding the result to the Current GCS Missile Position.

INPUTS: Current GCS Missile Velocity
Current GCS Missile Position

OUTPUTS: Projected GCS Missile Position

4.4.2.1.2 CALCULATE CURRENT AND PROJ. IMAGE OF SEEKER

DESCRIPTION: Current GCS Seeker Position and Projected GCS Seeker Position are calculated by using the transformed Current GCS Missile Position and Projected GCS Missile Position

INPUTS: Current GCS Missile Position
Projected GCS Missile Position
MCS To GCS Transform
Current MCS Seeker Angles
Projected MCS Seeker Angles
Historical GCS Seeker Position

OUTPUTS: Current GCS Seeker Position
Projected GCS Seeker Position

4.4.2.1.3 CALCULATE SEEKER ANGLES AND DOWNLINK

DESCRIPTION: The settings of the angles of the seeker are stored and rates of turning are determined. The Projected MCS Seeker Angles are determined from the Seeker Commands and the Current MCS Seeker Angles. Issue Seeker System Downlink from Seeker Commands and seeker positions.

INPUTS: Seeker Commands
Historical_Zoom_Position

OUTPUTS: Seeker System Downlink
Current MCS Seeker Angles
Projected MCS Seeker Angles

4.4.2.2.1. CALCULATE MANIPULATION POLYNOMIAL

DESCRIPTION: This process receives the Frame Transform Set along with the difference between Φ , Ψ and Θ of the Current GCS Seeker Angle and the Φ , Ψ and Θ of the Present Frame GCS Position. Using this information and Zoom Position It creates the Manipulation Polynomial.

INPUTS: Zoom Position
Present GCS Frame Position
Frame Transform Set

OUTPUTS: Manipulation Polynomial

4.4.2.2.2 CALCULATE PAN INSTRUCTIONS

DESCRIPTION: This process locates the subpicture within the present frame representing the Current_GCS_Seeker_Position by finding the displacement of the Current_GCS_Seeker_Position vector from the Present_Frame_GCS_Position vector.

INPUTS: Present_Frame_GCS_Position
Current_GCS_Seeker_Coordinates
Historical_Best_Fit_Frame_Number

OUTPUTS: Relative_GCS_Frame_Vector

4.4.2.2.3 CALCULATE NEXT FRAME

DESCRIPTION: This process determines Best Fit Frame Number by comparing the Relative GCS Projected Vector with the implicit relative vector's associated with each frame number in the data flow Frame Numbers.

INPUTS: Relative GCS Projected Vector
Frame Numbers

OUTPUTS: Best Fit Frame Number

4.4.2.2.4 CALCULATE RELATIVE PROJECTED VECTOR

DESCRIPTION: This process subtracts the Current GCS Seeker Position vector from the Projected Seeker Position vector to obtain the Relative_GCS_Projectected_vector.

INPUTS: Current_GCS_Seeker_Coordinates
Projected_GCS_Seeker_Position
Current_GCS_Seeker_Angles

OUTPUTS: Relative_GCS_Projectected_VECTOR

4.4.2.3.1 DIGITIZE AND RECOVER ENCODED INFORMATION

DESCRIPTION: This process performs an A/D conversion of the video frame. Encoded Frame Information is separated from the digitized picture contained on the frame.

INPUTS: Encoded Video Database

OUTPUTS: Digitized Frame
Frame Information Set
Frame Numbers

4.4.2.3.2 SWITCH FRAME BUFFERS

DESCRIPTION: This process changes frame Buffer 1 to Buffer 2 and Buffer 2 to frame Buffer 1 after the Digitized Frame has been read into frame Buffer 1 and a timer has run a predetermined period of time.

INPUTS: Buffer 1_Frame

OUTPUTS: Switched_Frame_Buffer

4.4.2.3.3 SELECT SUBFRAME FROM FRAME

DESCRIPTION: This process reads a Digitized Sub frame from Buffer_2_Frame at a location determined by Relative_GCS_Frame_Vector.

INPUTS: Relative_GCS_Frame_Vector.
Buffer_2_Frame

OUTPUTS: Digitized_Subframe

4.4.2.3.4 MANIPULATE IMAGE (WARP)

DESCRIPTION: This process manipulates (warps) the Digitized_Sub_Frame by using the Manipulation Polynomial.

INPUTS: Digitized Subframe
Manipulation_Polynomial

OUTPUTS: Warped_Subframe

4.4.2.3.5 CONVERT DIGITAL FRAME TO ANALOG

DESCRIPTION: This process converts the Warped_Subframe from a digital to an analog signal.

INPUTS: Warped_Subframe

OUTPUTS: Seeker_Video

4.4.3.1 INITIAL CONDITIONS

DESCRIPTION: This process begins the simulation. It retrieves the starting position reference from the Initial Simulation State Vector, and uses the Historical Position Reference to retrieve the Historical GCS Missile Coordinates, which are then issued to the missile's processor as an Initial GCS Missile Position. The Initial Uplink Information contains the Initial GCS Thrust. The Initial Simulation Phase Type is issued to designate the phase of the mission being simulated.

INPUT: Initial Simulation State Vector
Initial Uplink Information
Historical GCS Missile Positions
Initial GCS Missile Position
Initial Phase Type

OUTPUT: Initial GCS Thrust

4.4.3.2 DETERMINE ENDING CONDITIONS

DESCRIPTION: This process receives a Simulation_Phase_Type designating which phase of the mission is being simulated, and Final_Phase_Conditions containing the parameters terminating that phase. Mission results for each phase are:

- 1) Cruise phase - navigation parameters,
- 2) Target area phase - missile coordinates,
- 3) Lock-on phase - projected seeker coordinates,
- 4) Impact phase - impact coordinates.

Additionally, the process monitors the current Altitude and number of Fiber_Turns continuously during the simulation to determine the occurrence of an impact and maximum distance of flight, respectively. In either case, mission results reflect the manner in which the simulation terminated.

INPUT: Current_GCS_Missile_Position
Simulation_Phase_Type
Final_Phase_Conditions
Altitude
Fiber_Turns

OUTPUT: Mission_Results

4.4.3.3 DETERMINE INITIAL SPEED & HEADING

DESCRIPTION: Initial GCS Missile Velocity is determined by integration of Initial_GCS_Thrust.

INPUTS: Initial_GCS_Thrust

OUTPUTS: Initial_GCS_Missile_Velocity

4.4.3.4 RETRIEVE HISTORICAL DATA

DESCRIPTION: Retrieved the historical missile information and video database parameters necessary for the initialization of a missile flight simulation.

INPUTS: Historical Data File
Historical_Position_Reference

OUTPUTS: Historical_GCS_Missile_Coordinator
Historical_Best_Fit_Frame_Number
Historical_Zoom_Position
Historical_GCS_Seeker

PSUEDOCODE: Upon receiving an Historical_Position_Reference,
Retrieve Historical Data
Derive Historical_GCS_Missile_Coordinates from
Retrieved_GCS_Seeker_Position
Issue the following:
Historical_GCS_Missile_Coordinates
Historical_GCS_Seeker_Position
Historical_Zoom_Position
Historical_Best_Fit_Frame_Number

GLOSSARY

A/D conversion	Analog to Digital conversion
CAI	Computer Assisted Instruction. The presentation of information, such as pages of text, timed frame sequences, or video frames, where no scoring or record keeping is performed. CAI segments generally precede or follow an assessable segment and serve introductory, explanatory or feedback purposes.
CGI	Computer Generated Imagery.
Cruise	This is the second phase of the mission training. During this phase the training consists of the system deviating a navigational or flight parameter and assessing the gunner's ability to restore the parameter correctly within the time allowed.
CW	Courseware
CW_Ref	Courseware references
D/A conversion	Digital to Analog conversion
DMG	Digital Map Generator.
DPG	Digital Perspective Generator.
GCS	Gunner Coordinate System. The coordinate system with respect to the launch site.
FOL-Uplink/ Downlink	Fiber Optic Uplink/Downlink. Within the context of the ET FOG-M specifications, the FOL is a data flow over which the communication between the missile and seeker simulators and the operational system occurs.
Impact	This is the fifth and final phase of the mission. The training
Launch	This is the first phase of the mission. It is supervised from the operational system, just as it would be during an actual mission.
Lesson	A lesson is made up of topics. There are generally two types of lessons: instructional and simulation. CAI and PTT are instructional, while MFS is simulation. When all of the topics in a lesson are passed, the lesson is considered passed and so marked on the training records.

Lock-on This is the fourth phase of the mission. The gunner is expected to locate and mark targets on the ground. The assessment consists primarily of determining whether or not the cross hair position is within a specified target radius.

M/C Multiple Choice. A multiple choice segment is a series of multiple choice questions. Each question is considered to be a subsegment of the M/C segment. Multiple choice questions are divided into PDP type questions and Keypad type question. PDP type questions use PDPs, the advance subfunction key, and the fire switch as the means for gunner response. These devices are usually grouped logically during a launch phase, and are used for an M/C test on the launch system. Keypad subsegments are structurally simpler, involving only a video display and a set of keypad choices.

MCS Missile Coordinate System. The coordinate system with respect to the missile; formed by the main axis of the missile and the two perpendicular fins.

MFS Mission Fidelity Simulation. A full or partial simulation of the FOG_M missile, from launch to impact.

Mission A mission can consists of repeated mission segments of the same type or an ordered list of mission segments. A mission can begin or end with any segment, but end phase number cannot precede the beginning phase number, and all segments in between must be run sequentially and in order.

M/S Multiple Static. Refers to a two dimensional P/D segment in which prompted targets (map symbols) are to be marked by the gunner on a digital map. There may be up to fifteen targets markable during any segment, but the targets are constrained to be stationary.

PDP Programmable Display Pushbuttons, located the gunner console and used during launch.

P/D Point-Disk. Refers to a particular kind of training in which the joystick is used to maneuver crosshairs across a video image. Targets are marked or hit by centering the crosshairs over a target and pulling the trigger. Targets must be marked with a circle formed by a central point in the target and a specified radius (error). Each trigger pull counts as a response.

P/F Pass/Fail.

PTT Part-Task Training. PTT consists of the practice of

SCS	manipulative skills (hand-eye) and all phases of skills from launch to impact (example: the P/D problem). Seeker Coordinate System. The coordinate system with respect to the missile seeker.
S/D	Single Dynamic. Refers to a two dimensional P/D target tracking segment in which the gunner uses the joystick to manipulate a set of cross hairs over a geometric figure on the CRT. Each figure constitutes a subsegment. The segments come in four levels of based on the speed, size, and shape of the geometric figure.
Segment	The smallest assessable portion of any topic for which training records are kept. Assessment is performed on the graded subsegments for M/C and Pt-Disk, or upon termination of a simulation.
Subsegment	The smallest gradeable portion of a segment. For M/C, a subsegment is a single graded question, and for Pt-Disk, it is a single trigger pull.
SV	State Vector. The state describing the operational or simulation system at a particular point in an MFS simulation. Used for evaluation or initialization purposes.
Target_Area	This is the third phase of the mission. The missile is in flight and the gunners task is to navigate the missile to a target area. The target area is considered found when the missile coordinates are within a specified target area.
Topic	Topics are breakdowns of lessons into logical modules which are then taught individually. Topics are menus composed of items or segments. When all of the component items or segments are passed, then the topic is considered passed.

REFERENCES

AML, 1985

U.S. Army Missile Laboratory. B-5 Specifications for the FOG-M Operational Software. U.S. Army Missile Laboratory, 1985.

ASA, 1985a

Applied Science Associates, Inc. Design Concepts for FOG-M System Embedded Training (ET). Working paper by Applied Science Associates, Inc.; Vector Research, Inc.; and Interactive Graphic Systems, Inc.

ASA, 1985b

Ditzian, Jan L., Adams, James E., Sullivan, Gregg K. FOG-M System Embedded Training (ET) Demonstration Courseware Outlines. Applied Science Associates, Inc., 1985.

DeMarco, 1978

DeMarco, Tom. Structured Analysis and System Specification. New York: Yourdon, Inc., 1978

DoD, 1968

US Department of Defense. Military Standard Specification Practices, MIL-STD-490. 30 October 1968.

Myers, 1978

Myers, Glenford J. Composite/Structured Design. New York: Van Nostrand, 1978.

Yourdon, 1976

Yourdon, Edward. How to Manage Structured Programming. New York: Yourdon, Inc., 1976.

Yourdon and Constantine, 1975

Yourdon, Edward and Constantine, Larry. Structured Design. New York: Yourdon, Inc., 1975.