LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

AD-A277 098

DTIC ACCESSION NUMBER

LEVEL

1

INVENTORY

ASC-TR-94-5008

DOCUMENT IDENTIFICATION

Aug 93

DISTRIBUTION STATEMENT

HANDLE WITH CARE

ACCESSION FOR
NTIS GRA&I ☒
DTIC TRAC ☐
UNANNOUNCED ☐
JUSTIFICATION

BY
DISTRIBUTION/
AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL	
A-1		

DISTRIBUTION STAMP

DTIC ELECTE MAR 1 6 1994 S C D

DATE ACCESSIONED

DATE RETURNED

94 3 15 006

DATE RECEIVED IN DTIC

48 94-08402

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM JUN 90 70A

DOCUMENT PROCESSING SHEET

PREVIOUS EDITIONS MAY BE USED UNTIL STOCK IS EXHAUSTED.

LOAN DOCUMENT

ASC-TR-94-5008

MODULAR SIMULATOR SYSTEM (MSS) MANAGEMENT GUIDE

K KELLY, J BROWN,
G KAMSICKAS, W TUCKER

BOEING DEFENSE AND SPACE GROUP
SIMULATION AND TRAINING SYSTEMS
499 BOEING BLVD
HUNTSVILLE, AL 35824

AUGUST 1993

FINAL REPORT

SYSTEMS ENGINEERING DIVISION
AERONAUTICAL SYSTEMS CENTER
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7126

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

JEFFREY C. VALITON, Maj, USAF
Program Manager
Special Programs Divsion

JAMES D. BASINGER
Team Leader
Special Programs Division

JAMES J. O'CONNELL
Chief, Systems Engineering Division
Training Systems Program Office

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify ASC/YTSD WPAFB, OH 45433-7111 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE 20 Aug 93

3. REPORT TYPE AND DATES COVERED FINAL

4. TITLE AND SUBTITLE
Modular Simulator System (MSS)
Management Guide

5. FUNDING NUMBERS
F33657-86-C-0149
64227F

6. AUTHOR(S) M. Kelly, J. Brown
G. Kamsickas, W. Tucker

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Boeing Defense and Space Group
Simulation and Training Systems
499 Boeing Blvd
Huntsville, AL 35824

8. PERFORMING ORGANIZATION REPORT NUMBER
S495-10459-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Aeronautical Systems Center
Systems Engineering Division
Bldg 11 2240 B St Ste 7
Wright-Patterson AFB, OH 45433-7111

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ASC-TR-94-5008

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release;
distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document provides guidance and reference information for management personnel in the use of the Modular Simulator System (MSS) design principles. It identifies those unique management considerations required when employing the modular simulator design to a simulation and/or training device. This guide serves as both an educational and decision tool for the manager. The guide will educate the manager in the basic concepts of the modular simulator design and its basis for development. Key program decisions and considerations that must be made when using the modular simulator design are identified and discussed. The advantages and disadvantages of using the modular simulator design along with associated cost/risk analysis, logistics impacts, management implications, lessons learned and basis for invoking the modular simulator design on a procurement/program are provided.

14. SUBJECT TERMS
Modular Simulator System (MSS)

15. NUMBER OF PAGES 49

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std 239-18
298-102

THIS PAGE INTENTIONALLY LEFT BLANK

Abstract

This document provides guidance and reference information for management personnel in the use of the Modular Simulator System (MSS) design principles. It identifies those unique management considerations required when employing the modular simulator design to a simulation and/or training device. This guide serves as both an educational and decision tool for the manager. The guide will educate the manager in the basic concepts of the modular simulator design and its basis for development. Key program decisions and considerations that must be made when using the modular simulator design are identified and discussed. The advantages and disadvantages of using the modular simulator design along with the associated cost/risk analysis, logistics impacts, management implications, lessons learned and basis for invoking the modular simulator design on a procurement/program are provided.

Table of Contents

List of Figures

1. INTRODUCTION.

1.1 Purpose. The purpose of this document is to provide guidance and reference information for managers who are considering the Modular Simulator System (MSS), or "Mod Sim" approach to simulator development. This guide serves as both an educational and decision tool for the manager. The guide will educate the manager in the basic concepts of the modular simulator design and its basis for development. Key decisions and considerations that must be made when using the modular simulator design are identified and discussed.

1.2 Scope. This management guide is applicable to a Mod Sim design based program. It identifies those unique management considerations required when employing the modular simulator design to a simulation and/or training device. The advantages and disadvantages of using the modular simulator design along with the associated schedule, cost, risk, support, and management considerations are discussed. Lessons learned during the demonstration project to develop an F-16C simulator using the Mod Sim approach are also provided. This guide is focused on management, specifically toward the first level supervisor, engineering manager, and project manager. It is intended to assist this level of management in the development of a program management plan for a Mod Sim based project.

Throughout this document the terms 'prime contractor', 'system integrator', 'system developer', 'subcontractor', 'supplier', and 'segment developer' are used. The prime contractor is the company responsible for delivery of the end system to the customer. In most cases, the prime contractor will also be the system integrator and system developer. However, this is not a requirement. The task of system integration can be assigned or subcontracted to another company. The terms prime contractor, system integrator and system developer are used interchangeably in this document. The terms subcontractor, supplier, and segment developer refer to the person or company responsible for the development of a Mod Sim segment. These terms are used interchangeably in this document. However, it is quite possible and likely that the prime contractor will develop several of the segments for an implementation and will therefore also be a segment developer.

2. REFERENCE DOCUMENTS.

The following documents are provided for further reference and would be useful in the management and design efforts of a modular simulator design program.

a. System/Segment Specification for the Generic Modular Simulator System, Volumes I-XIII, S495-10400.

b. System/Segment Specification for the F-16C Modular Simulator System Demonstrator, Volumes I-XII, S495-10415.

c. Interim Report on Research and Development for Modular Simulator System Phase III Program, Part 1, D495-10402-1, 15 September 1988.

d. Modular Simulator Design Program, Phase III, Part 2 Final Report, D495-10437-1, 29 August 1991, Revision B.

e. Follow-on Effort Final Report for the Modular Simulator Design Program, D495-10438-1, 28 August 1991.

f. Modular Simulator Engineering Design Guide, D495-10440-1.

g. Interface Requirements Specification for the Generic Modular Simulator System, S495-10734.

h. Interface Design Document for the Generic Modular Simulator System, D495-10735-1.

i. Modular Simulator Executive Report, D495-10441-1

3. MODULAR SIMULATOR DESIGN OVERVIEW.

3.1 Modular Simulator Concept. The Modular Simulator System (Mod Sim) Design program was a tri-service supported development program. The primary goals of the modular simulator design are to shorten simulator development schedules, reduce simulator development costs and improve simulator supportability. To achieve these goals a generic Weapon System Trainer (WST) was partitioned into a discrete number of modules or segments. Specifications that define/standardize each segments functions and a method for intersegment communication were then prepared.

The development of the Mod Sim was accomplished using the three phase process shown in Figure 3.1-1. Phase I consisted of a Request For Information (RFI) to the simulation industry. The purpose of this RFI was to survey the simulation industry to assess the industry's desire for and the general feasibility of the modular simulator concept. The results of this survey were very positive. This led to the Phase II, Modular Simulator Design Concept Development. This was a competitive procurement awarded to two companies, Boeing and Logicon. The products of this phase were a conceptual modular simulator architecture with a focus on aircrew simulator functional analysis and intermodule communication architecture/design. Each contractor developed a conceptual modular simulator design for this effort. The contractor's conceptual design formed the basis for their proposal for the Phase III contract. Boeing was awarded the Phase III contract, which consisted of design, demonstration and validation of the modular simulator concept.

To foster industry participation and "buy in" to the Mod Sim design, Boeing was required to subcontract the design and development of 50 to 75 percent of the segments. The Phase III subcontractors were Rediffusion Simulation Limited (RSL), Science Applications International Corporation (SAIC), AAI, and Intermetrics. To gain further industry participation, regular Interface Standards Working Group (ISWG) meetings were held. At these meetings both industry and government simulation experts were allowed to participate in the review of the modular simulator design and subsequent demonstration.

Phase III was divided into two parts. During Part 1, Boeing and the Boeing subcontractors accomplished four major tasks:

a. System Partitioning. The process for this task is shown in Figure 3.1-2. This task involved the analysis of simulations for a large number of fixed and rotary wing training devices. Each simulation was broken down to its low level objects and functions. This data, along with other raw data and the conceptual partitioning from Phase II were used to create a Functional Dictionary that contained an allocation of all

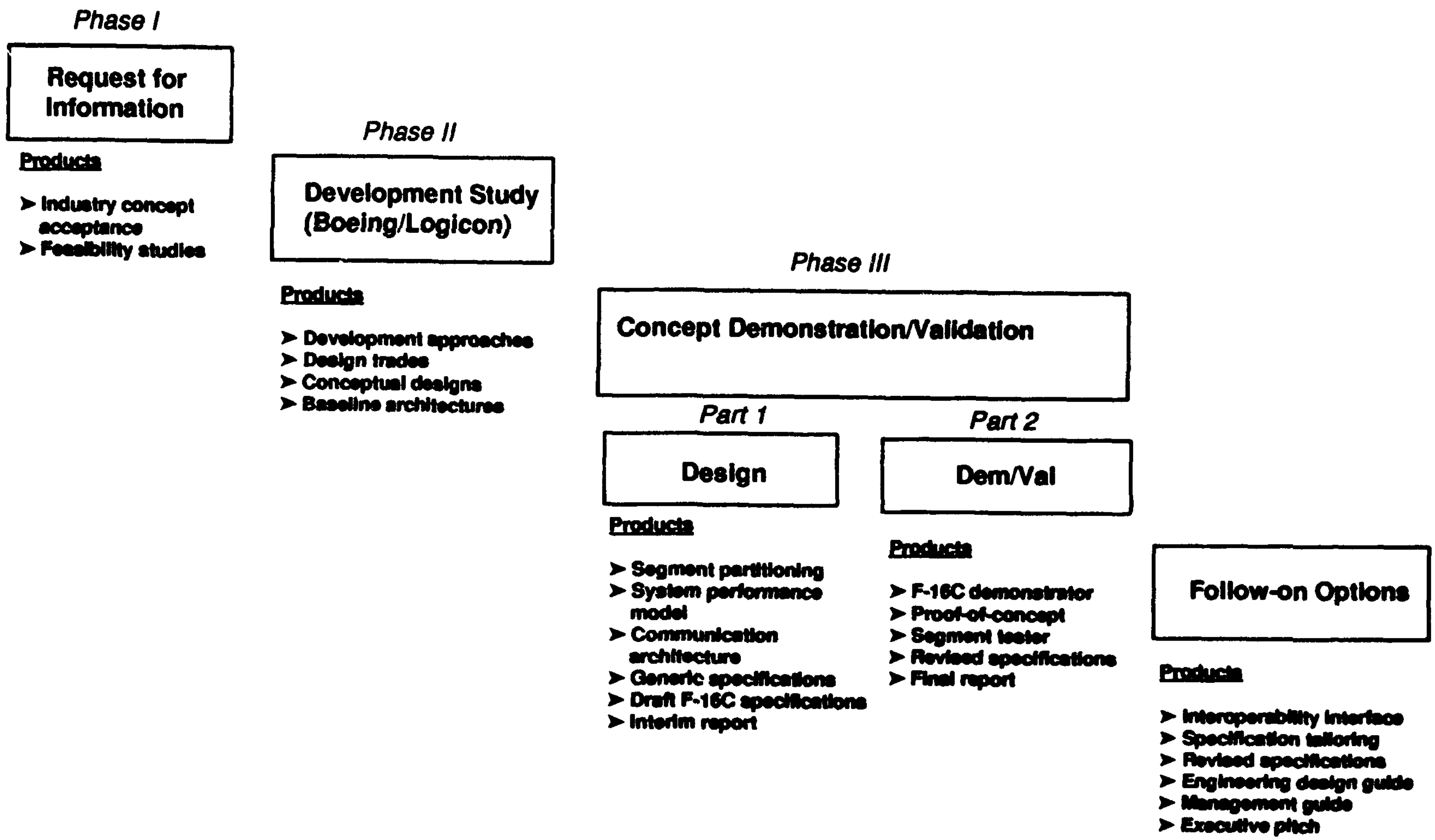

Figure 3.1-1 Modular Simulator System Development Process

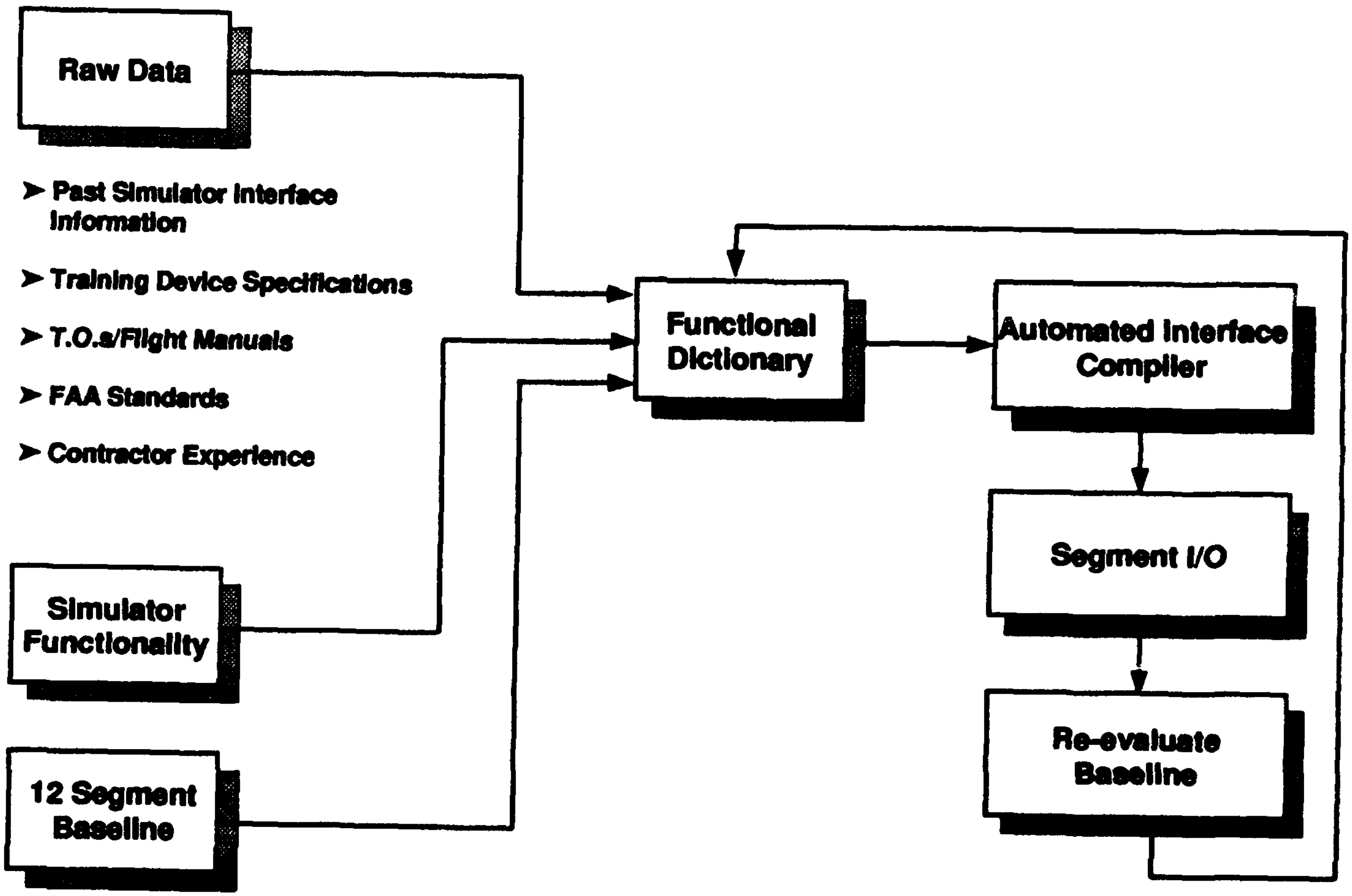

Figure 3.1-2 Modular Simulator System Partitioning Process

functions and the interface requirements between functions. The Functional Dictionary and segment partitioning were refined through an iterative process using an Artificial Intelligence tool. This resulted in segments that had generic intersegment interfaces,were loosely coupled, and focused on a specific area of simulation expertise

b. Communication Architecture. This task involved the identification and specification of a hardware and software communication architecture that would allow the segments to communicate effectively. The communication architecture had to be of a non-proprietary design, support industry standards, exhibit low data latency, provide 50% growth and be available for use in the concept demonstration.

c. System Performance Model. In order to efficiently select a communication architecture a System Performance Model was constructed to emulate the various design alternatives. Fourteen data buses and seven protocols were analyzed. The communication architecture selected was the Fiber Distributed Data Interface (FDDI) coupled with the Xpress Transfer Protocol (XTP).

d. Specifications. To promote the standardization of the Mod Sim architecture, a thirteen volume generic System/Segment Specification (DOD-STD-2167A) was prepared. The organization of this specification is shown in Figure 3.1-3. The system level specification defines the communication architecture and requirements common to all segments. The segment level specifications define the unique requirements applicable to the segment.

During Part 2 of Phase III, Boeing and the Boeing subcontractors demonstrated the modular simulator design developed in Part 1. The demonstration was accomplished using a government provided F-16 crew station and existing F-16 simulator source code. The government furnished products were adapted to the modular simulator partitioning and communicated using the modular simulator communication architecture. Other tasks accomplished during Part 2 included the development of a Segment Tester tool, used to test individual segments prior to integration into the system, and further refinement of the generic system/segment specifications based on lessons learned during the demonstration.

At the completion of the Phase III, Part 2 demonstration, several follow-on tasks were contracted to Boeing. These consisted of adding an interface to the Mod Sim architecture to support multiple simulator/team training (e.g.; Distributed Interactive Simulation), adding tailoring instructions to the generic specifications to ease adaptation to specific applications, and the creation of Mod Sim guidance documentation. This documentation included an engineering design guide, a management guide and an executive report that provides an overview of the Mod Sim approach.

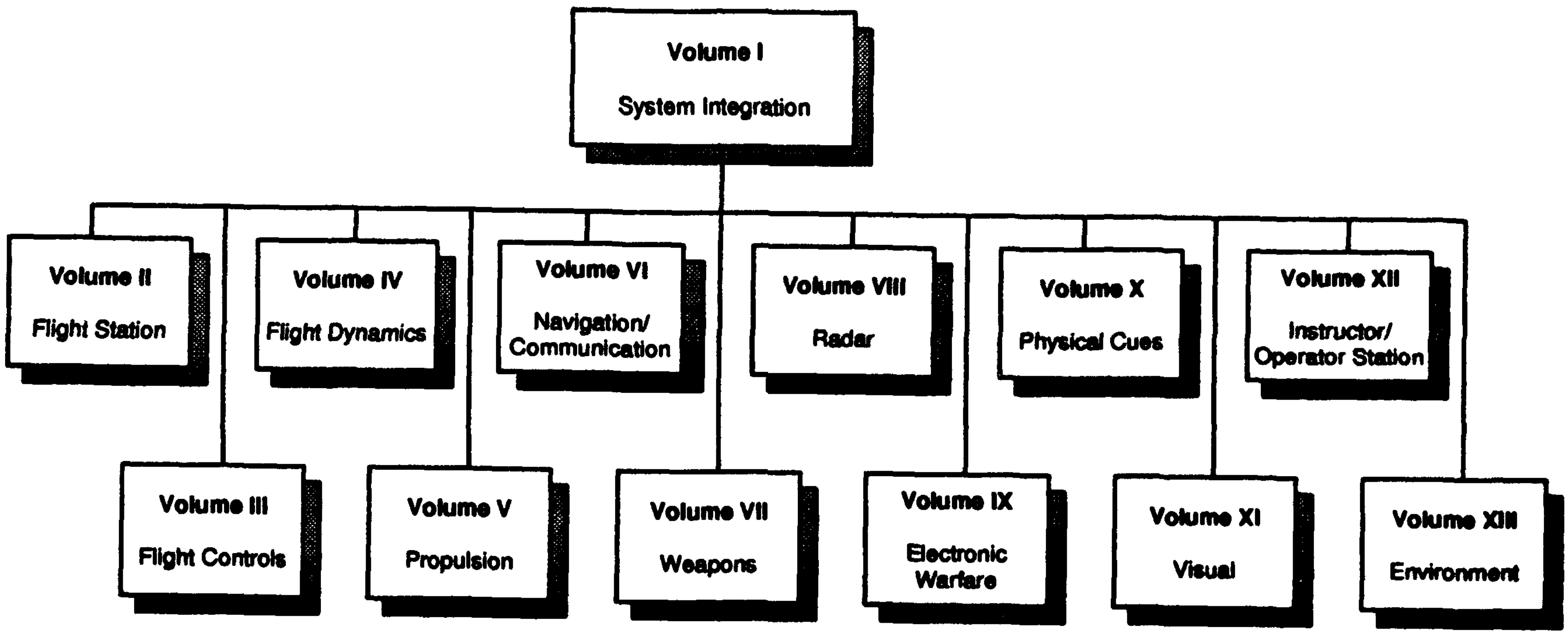

Figure 3.1-3 Modular Simulator System Generic Specification Organization

The current Mod Sim architecture is shown in Figure 3.1-4. The architecture consists of 12 distinct segments that communicate via a Virtual Network (VNET). The Mod Sim communication architecture was revised to the VNET to make the Mod Sim communication architecture independent of a specific hardware implementation. This allows the architecture to be more adaptable to high end and low end applications and further adaptable to advances in computer technology. The VNET is a conceptual mechanism for communication between segments using the message passing protocol regardless of the hardware implementation. The FDDI/XTP version of the communication architecture remains the default implementation.

3.2 Application of the Modular Simulator Design. The modular simulator design is applicable to all types of aircrew training devices for both rotary wing and fixed wing aircraft including Weapon System Trainers (WST), Operational Flight Trainers (OFT), Cockpit Procedures Trainers (CPT) and Part Task Trainers (PTT). With minor modifications the Mod Sim design has been applied to land and sea based vehicles. The Environment segment also promotes the application of the Mod Sim design to networked as well as stand-alone training devices.

One of the questions frequently asked about Mod Sim is "Should I use the Mod Sim Design in my application?'. In most cases the answer is yes. To determine if a specific application would benefit from a Mod Sim design the answers to the following questions should be reviewed.

a. Does the application involve a family of trainers? If the application involves the development of a WST and a CPT for the same aircraft then it would be beneficial to use the Mod Sim design. Entire segments could be reused on each device. This reduces development time and cost through segment reuse. Each segment is treated as an individual product that can be developed, tested, documented, and managed once and then reused several times.

b. Will your application require modifications or upgrades during development or after deployment? The modular simulator design provides a loose coupling of segments and well defined intersegment interfaces. This characteristic isolates and encapsulates changes to the training devices. For example, the upgrade to a visual system or addition of a motion system would only affect the Visual and Physical Cues segments respectively. An upgrade in fidelity or functionality of the simulation in any segment would, in most cases, only affect the internal design of that segment.

Maximum leverage will occur when the Mod Sim approach is applied to new simulator developments. New developments allow the Mod Sim principles to be applied from the ground up. This takes full advantage of the front end systems engineering work

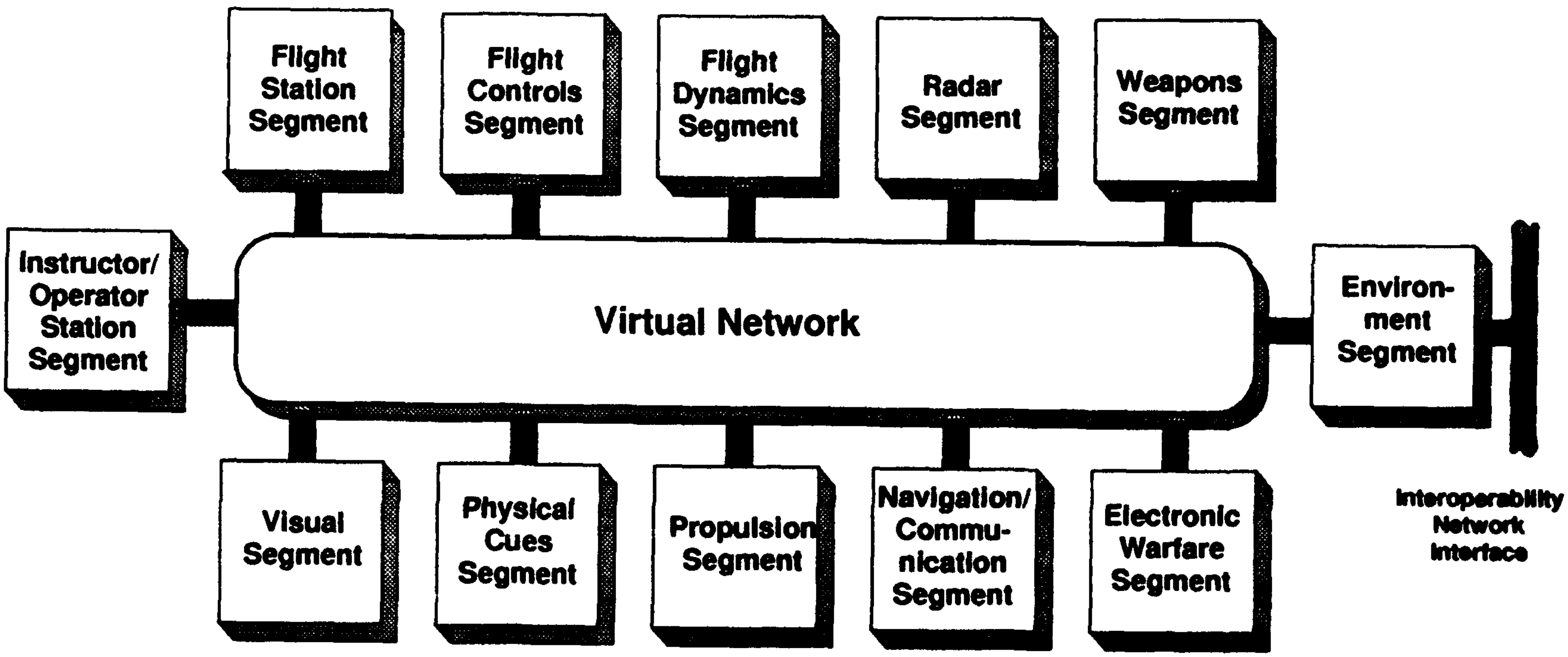

Figure 3.1-4 Modular Simulator System Architecture

that has been done in the generic system/segment specifications and intersegment interface definitions. The Mod Sim design approach can also be applied to existing simulation devices that are undergoing modification or upgrade. It is recommended that trade studies be performed to determine if it is cost effective to repartition an existing device into a Mod Sim architecture. In most cases a significant upgrade would be required to justify the cost of a total system redesign. Care should also be taken to avoid a "beat-to-fit" application of the Mod Sim design just for the sake of creating a modular simulator.

3.2.1 Benefits of the Modular Simulator Design. There are several distinct advantages to using the modular simulator design and design concepts in developing training devices. These advantages include:

a. Systems Engineering. The Mod Sim design provides a wealth of generic systems engineering products that are reusable for any application. These products include the generic System/Segment Specifications, Interface Requirements Specification, and Interface Design Document. Each of these products include tailoring instructions and guidance to create application specific documents from the generic products. This reduces front end development cost and schedule and mitigates risk throughout the project.

b. Subcontracting. One of the primary requirements for the Mod Sim architecture was the capability to independently specify, develop, and test individual segments as stand-alone products. This enhances the ability to subcontract development of segments by providing well-defined interfaces that reflect a straight-forward allocation of simulator functions along traditional subsystem product boundaries. This allows the prime contractor to more readily take advantage of expertise in other companies, create teaming agreements, or develop workshare allocations both internal and external to their company. The generic specifications and interface definitions provide a strong foundation for defining subcontracting/teaming agreements.

c. Integration. Use of the Mod Sim architecture can significantly reduce integration time. Because each of the Mod Sim segments are individually tested based on a well defined system interface prior to integration, the probability of finding major problems during integration is virtually eliminated. This was proven during the Mod Sim F-16C demonstration project. System integration was accomplished in several weeks versus the usual several months.

d. Reusability. The Mod Sim architecture promotes reuse among families of training devices and can also support reuse among different training device applications. The generic specifications and system interfaces have identified a large number of commonalities and variabilities to allow for the design of reusable segments and communication architectures.

e. Design Flexibility. The Mod Sim architecture allows latitude in design to support low cost and high cost devices. The Mod Sim architecture does not place any requirements on the internal design of the segments. A segment developer is free to determine the best design solution for their segment.

f. Parallel Development and Stand-alone Testing. Mod Sim segments can be developed and tested in parallel due to the well defined segment requirements and intersegment interfaces. This can significantly shorten the overall system development schedule and reduces integration risk by eliminating common interface problems early in the development and testing phases.

3.2.2 Common Misconceptions Regarding Modular Simulator Design. During the development and demonstration of the Mod Sim design, no major disadvantages were identified. However, several misconceptions have developed as the program progressed. A manager should be aware of them and consider the facts to make an informed decision. These common misconceptions include:

a. Hardware Solution to a Software Problem. The original modular simulator concept was based on partitioning the system into distinct stand-alone segments composed of all hardware and software components necessary to meet the segment performance requirements. Each segment would communicate with other segments via a rigidly defined network using a predefined protocol. Some individuals felt that this hardware partitioning was done to force the software into a more modular architecture. This was not the case. The Mod Sim architecture allows for a wide range of design solutions involving both hardware or software alternatives.

b. Use of FDDI. The original Mod Sim design used FDDI as the communication media. FDDI was selected based on a worst case analysis of communication traffic for a WST. The current Mod Sim architecture embraces a "Virtual Network" concept. The virtual network concept supports the use of a communication architecture that best fits the requirements of the application. For example, a low cost, low bandwidth application may use ethernet or reflective memory. FDDI is still specified in the generic specification for the Mod Sim since this is the architecture that was tested and demonstrated for the proof of concept. The specifications may be tailored to accommodate other alternatives based on the specific application.

c. Architecture Too Rigid. The generic Mod Sim architecture is based on 12 separate and distinct segments. This approach was selected to maximize large scale, whole segment reuse. The current architecture has adopted the approach that segment software may be combined into a single hardware platform or computational system. However, the segment software must still be stand-alone and communicate with other segments via the intersegment interfaces. In general, the segments should not be aware that they reside in the same machine. This promotes maximum software reuse.

d. No Successful Full Scale Demo. The Mod Sim demonstration involved rehosting existing software that was repartitioned to the Mod Sim architecture requirements. Because of the classified nature of portions of the F-16 software, several segments were not fully developed but "emulated" to generate representative intersegment data transfer. The goal of the demonstration was to prove the communication architecture and Mod Sim approach, not to show how well a simulation could be developed. The communication architecture was fully loaded as if the F-16 demonstrator was a full scale simulation. The communication architecture performed within the modeling predictions and provided acceptable performance. It should be noted that the Mod Sim architecture has since been applied to other simulator applications both within and external to Boeing.

e. Specifications. The original Mod Sim generic specifications and interface data were deemed difficult to understand and use by industry. Because of this concern, Boeing incorporated tailoring instructions in the specifications to help other contractors better understand and apply the Mod Sim architecture. To ensure maximum understanding, the tailoring instructions were developed by engineers who were not part of the original development effort.

f. Message Passing. The Mod Sim architecture uses message passing to transfer data between segments. This method of communication was thought to be too slow for real time simulation. This is not the case. The Mod Sim proof of concept demonstrated that message passing would work. Even at worst case only 6% of the FDDI bandwidth was used. Message passing has been used in many real time applications other than Mod Sim in the past.

g. Functional Decomposition. One complaint is that Mod Sim dictates a functional decomposition versus an object oriented decomposition. The Mod Sim architecture is not intended to force any specific software architecture within a segment. Software design internal to the segment is at the discretion of the segment developer and may be of any design methodology. The top level segment partitioning was determined based on several factors including; domain expertise, subcontractability, functionality, simulation industry trends, communication performance, etc. Each of the segments deal with a recognized discipline within the training system industry. These are high level disciplines which traditionally fall within established product line boundaries and involved specific technical specialities such as image processing, audio processing, electromechanics, aerodynamic modeling, electronic combat modeling, etc. Segment functionality was allocated to maximize the potential for specialization while considering technology trends. Specialization invites increased competition by breaking large applications into subsystems that can be built by smaller suppliers. Complexity and quantity of data flow between segments was also considered in the allocation to reduce the interdependence of the segments. In summary, there was no specific, traditional decomposition methodology employed.

A review of the segments shows that some have a functional nature, such as Flight Dynamics, others have a traditional simulator subsystem nature, such as Visual, and some have an object oriented nature, such as Propulsion.

h. Mod Sim Validation. The criteria for a device to be considered a "Mod Sim" is unclear. Industry was unsure which parts of the Mod Sim architecture were required and which were optional. To help better define the boundaries of Mod Sim, Boeing prepared tailoring instructions for the generic specifications and a Mod Sim design guide. These two documents provide the guidance necessary to competently apply the Mod Sim design principles.

i. Proprietary. There is a misconception that Mod Sim is a proprietary Boeing product. It is not. Boeing was the prime contractor for the Mod Sim development. The Mod Sim architecture is intended to be used throughout the simulation industry and is a public domain development. Boeing and the government freely distribute information on the Mod Sim design.

3.3 Other Related Standards and Processes. There are several other DoD initiatives that involve the standardization of simulators and training systems. The Mod Sim architecture does not constrain the use of these standards. In fact, some characteristics of the architecture were designed to accommodate or enhance compatibility with these standards.

3.3.1 Distributed Interactive Simulation (DIS). The Mod Sim architecture supports multiple simulator operations, such as those provided by the DIS protocol. The environment segment provides a seamless connection between the Mod Sim segments and the multiple simulator environment. All remaining segments, except the Instructor/Operator Station (IOS), function the same in multiple simulator and autonomous operations.

3.3.2 Project 2851. Project 2851 is working to standardize visual and radar databases. Project 2851 causes no impact on the Mod Sim architecture. The use of databases are internal to the segments who use the databases. These segments include Visual, Radar, and Environment. Database information is not communicated via the Mod Sim communication architecture during real-time operations.

3.3.3 Simulator Data Integrity Program (SDIP). The SDIP has developed standards for supplying and specifying design data for simulation devices. This has no affect on the Mod Sim design or architecture. The SDIP standards should be used when developing any simulation device to avoid or reduce design criteria problems.

3.3.4 Universal Threat System for Simulators (UTSS). UTSS is in the process of developing common or reusable threat databases. Once again, the affect of this initiative will be internal to the Mod Sim segments. In the current Mod Sim architecture, the environment segment provides the simulation of the external threat environment and the electronic warfare segment provides the simulation of the ownship Electronic Warfare (EW) equipment. The only segments that would be internally affected by UTSS will be electronic warfare and environment.

4. MODULAR SIMULATOR PROGRAM MANAGEMENT

4.1 Management Overview. The management of a Mod Sim based simulator development is very similar to a traditional simulator development. There are still the day-to-day management activities associated with cost, schedule and resource (personnel, equipment, data, etc.) allocation. The prime contractor plays a system integrator and coordination role in the development of the training system. For some contractors this would not be a change, but for those companies that produce the majority of the training device in-house this could be a drastic change. The prime contractor should posses strong systems engineering and integration skills to successfully implement a Mod Sim. If segments will be developed and furnished by subcontractors, a strong subcontract management capability is also required,

From a customer viewpoint, a prime contractor should be selected based on their past successful experience in system integration and demonstration of a broad training system experience base. At a system level the use of the Mod Sim architecture is transparent to the user. The Mod Sim approach affects the underlying architecture of the system to generate a more efficient simulator development in terms of cost, schedule and risk. In the long term, the Mod Sim approach should reduce the cost of system upgrades throughout the life cycle of the system.

4.2 Schedule. There are several scheduling factors to consider when planning a modular simulator implementation. Figure 4.2-1 provides a comparison between a typical development schedule and a suggested Mod Sim development schedule. This schedule assumes that the simulation device is a WST of average to above average complexity, the user requirements are defined and relatively stable, and the device requirements, such as those provided by a formal Instructional Systems Development (ISD) process, are also available. The Mod Sim approach to development of such a system can save approximately 9 months or 20-25% when compared to a traditional development approach for the same device. The supporting rationale for this estimate is provided in the following paragraphs.

A modular simulator implementation is similar to managing a set of small, individual programs that require integration at some point. One of the advantages of a Mod Sim implementation is a potentially shorter development schedule. The development schedule is shorter due to several factors. First, there is a significant amount of reusable systems engineering provided by the generic specifications and interface design. These reusable work products reduce the requirements definition and system level design phases considerably. Once the interfaces for the modules/segments are defined; design, development and stand-alone segment test can occur in parallel for each segment.

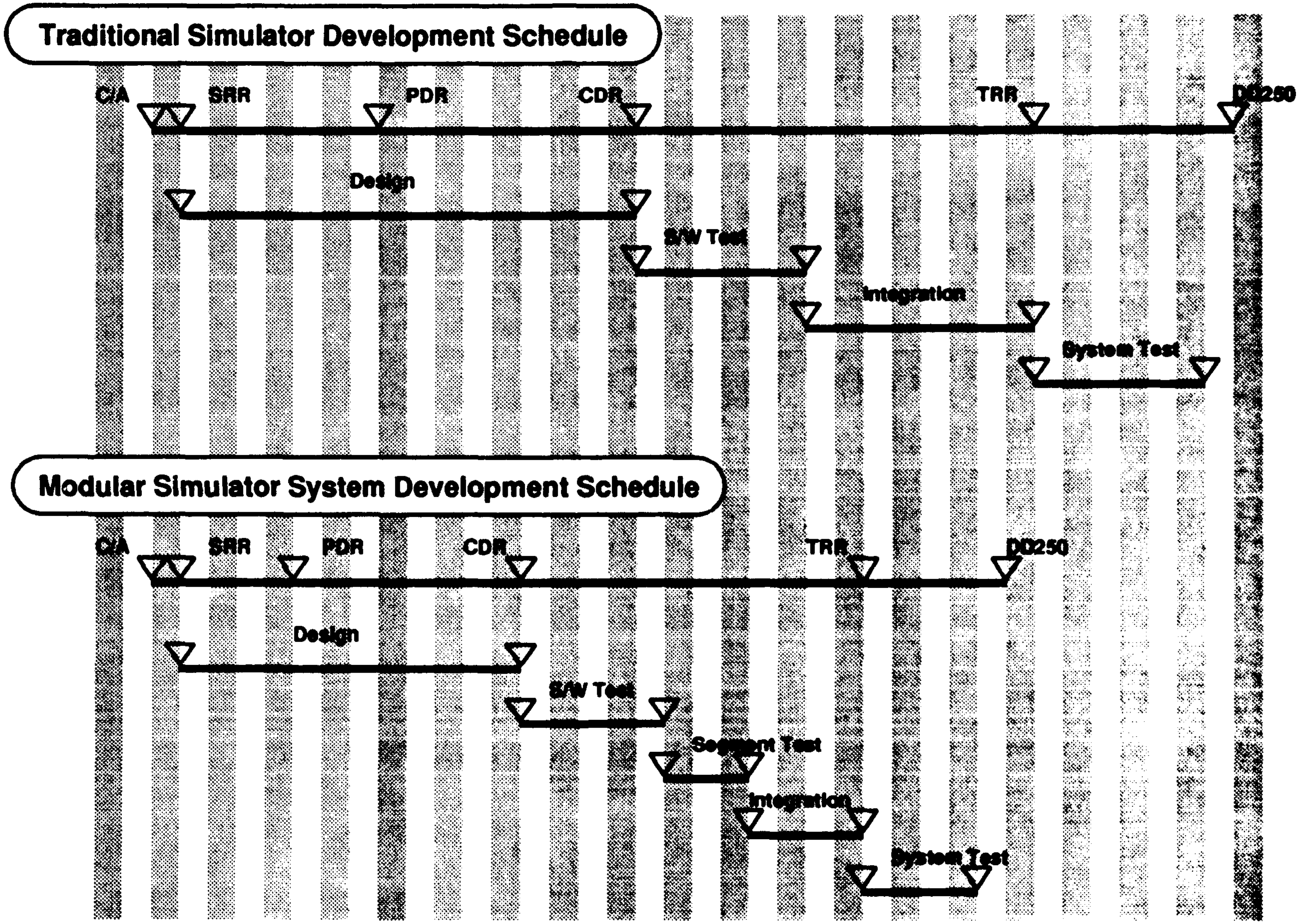

Figure 4.2-1 Modular Simulator System Schedule Comparison

The software and segment test phase for a Mod Sim development will typically be longer than traditional software test due to the addition of formal segment stand-alone testing. Stand-alone segment testing tests each segment as a product when accepted from the segment builder or subcontractor. When each of the segments are thoroughly and successfully tested prior to integration, the time required for integration and system level testing is significantly reduced. Stand-alone segment test identifies segment design errors earlier in the development cycle where they can normally be resolved in less time and at a reduced cost. The use of an automated segment testing device can reduce the stand-alone segment testing time significantly.

Experience from the development of the F-16C simulation device developed during the Mod Sim demonstration program showed reductions in system integration and system level test. This experience shows that integration and system level testing time can be reduced by at least 40 and 10 percent respectively when compared with traditional simulation device development. It should be noted that there is a preferred order for integration of the segments into the system. This is shown in Figure 4.2-2. The rationale for this sequence is provided in the "Modular Simulator Engineering Design Guide".

4.2.1 Schedule Risk. Schedule risk is inherent in any program. However, two phases in the Mod Sim development schedule are noteworthy when segments are developed by subcontractors. First, schedule risk may increase when the subcontractors are provided with contractor furnished equipment or data. For example, it may be cost effective for the prime contractor to make or buy hardware and software components that are common to several segments, such as computational equipment and the interface to the Virtual Network. In such cases, schedule reserve may be necessary to insure that procurement or development schedule problems incurred by the prime contractor do not impact the segment suppliers. A second schedule risk factor involves stand-alone segment test completion and the start of integration. Again, schedule reserve may be necessary to negate adverse integration schedule impacts due to late segment delivery from a supplier. The amount of schedule reserve will depend on the level of risk associated with each situation. For example, the prime contractor may have a common, fully tested, off-the-shelf VNET interface or a supplier may have an off-the-shelf segment, thereby reducing or eliminating associated schedule risk and the need for schedule reserves.

4.2.2 Program Phasing and Milestones. A Mod Sim implementation is compatible with MIL-STD-1521 program phasing and risk reduction milestones. However, several of the typical MIL-STD-1521 meetings and reviews between the prime contractor and the customer can be held sooner in a Mod Sim implementation. Many aspects of the front end systems engineering and system design are provided in the generic system/segment specifications and the modular simulator design/management

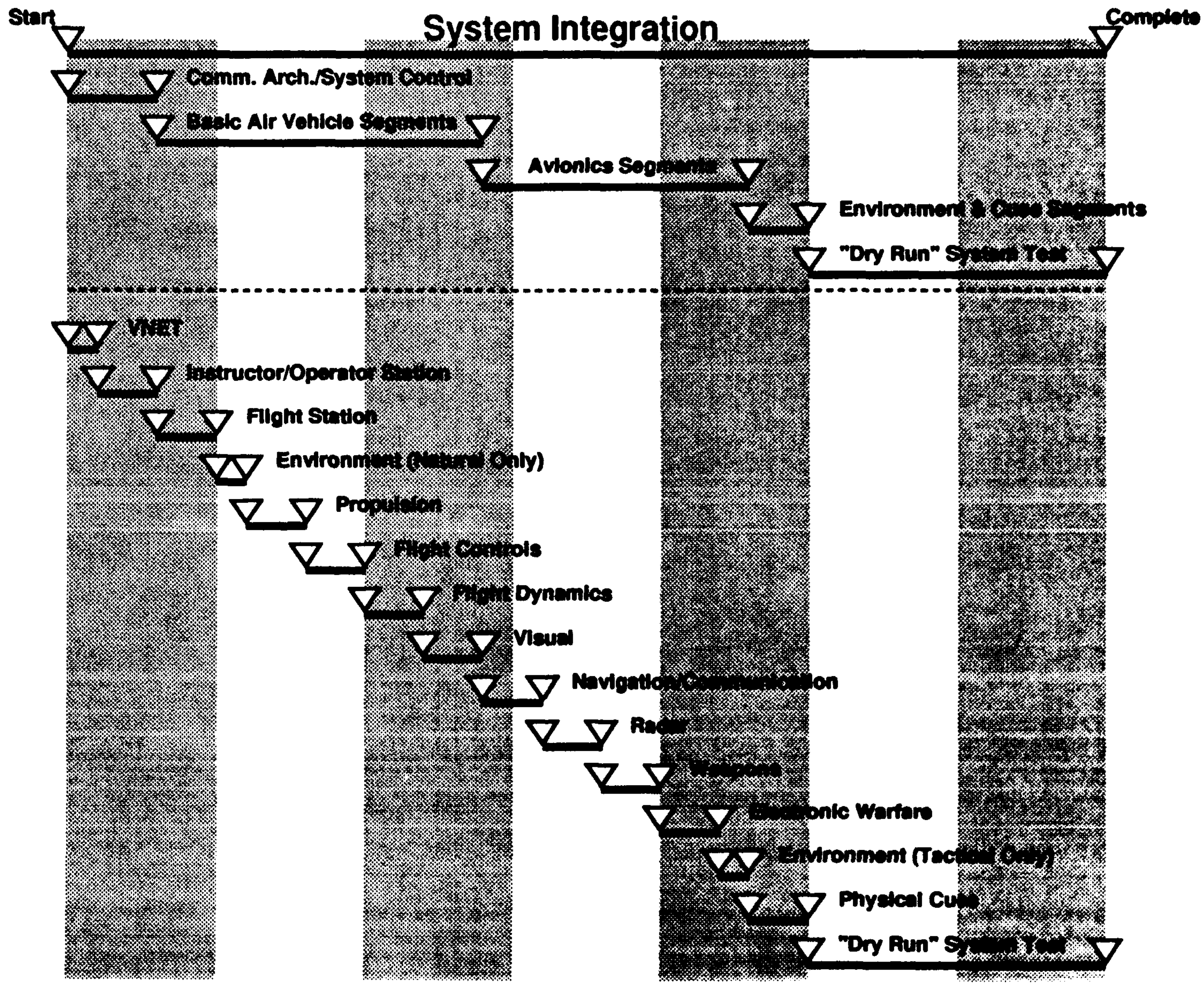

Figure 4.2-2 Modular Simulator System Segment Preferred Integration Order

guides reducing time necessary to develop the top level system architecture. Effort can be focused on understanding the requirements specific to the application. Guidance for the more prominent customer reviews is as follows:

a. System Requirements Review (SRR). SRR should probably still be held at the typical 30 days after contract award. This allows the prime contractor and the subcontractors to come to a common agreement and understanding of the system level and supporting segment level requirements. The prime contractor and the segment suppliers should be working very closely during this initial 30 days. It is suggested that the development team be collocated during this time to increase productivity, foster team building and reduce feedback time. This would also fulfill the requirement for a supplier SRR.

b. Preliminary Design Review (PDR). PDR can be held significantly earlier in a Mod Sim development, possibly as early as 30 days after SRR. This assumes that a detailed Prime Item Development Specification (PIDS) is available at the SRR. By the end of SRR each segment should have a concise specification and a set of well defined, compilable, external interfaces. The system level specification will be completed and the system level design should be documented and allocated to each of the team members. The remaining tasks to get to PDR will consist of internal segment preliminary design and creation of the top level hardware drawings. These tasks could be completed in 30 to 90 days. The Mod Sim approach can save approximately three months in the preliminary design phase.

c. Critical Design Review (CDR). The Mod Sim approach does not dramatically accelerate the detailed design phase that occurs between PDR and CDR. Some schedule improvement may be enabled by parallel development of the segments. However, parallel development is also possible in traditional simulator developments. In a traditional simulator development, detail design is normally done in parallel by smaller design teams. It is likely that the Mod Sim approach can save approximately one month during the detailed design phase due to the sound systems engineering foundation created early in the project and reduced coupling between the segments.

d. Test Readiness Review (TRR). A TRR will be held for the system level testing. This will be after the development, segment test and integration phases. The timing for the TRR will be earlier than for a traditional development effort. A portion of the traditional software tests will be accomplished during segment testing, resulting in a shorter software test period. The segment test period, which is unique to Mod Sim, will increase lower level testing due to a longer and more rigorous stand-alone segment test period. However, the integration period will be significantly shorter as discussed earlier, offsetting the longer test period. The TRR itself can potentially be longer in duration. The data from segment testing could be reviewed by the customer requiring an extra 1 to 2 days. This data would be in addition to the typical review of contractor run system level test results. At least two months can be saved during the time between CDR and the TRR using the Mod Sim approach.

Typical MIL-STD-1521 meetings were conducted for the modular simulator demonstration device development and were successful. The only problem encountered with the demonstration device meetings was preparation time. This problem is not unique to a Mod Sim implementation and may occur whenever subcontractors are involved. Preparation for the meetings is longer due to the subcontracting of the design effort for a large number of segments. For example, in preparation for customer CDR, major supplier CDRs were held with all segment developers. This had to occur several weeks in advance of the customer CDR in order to review and resolve supplier designs and integrate supplier briefings into the customer CDR. There are options to this approach that can save time but may involve some risk. One option is to allow the subcontractor to prepare and present their respective portion of the design at the customer CDR. To reduce the risk of the supplier not being prepared, dry runs can be held prior to the customer review. To reduce the rework of presentation materials, chart formats, media and graphics software should be compatible among team members to ensure changes can be made rapidly and charts can be modified by the prime contractor if changes are required. The prime contractor should allocate a small amount of additional time (3 to 5 days) for segment supplier reviews prior to customer reviews as a minimum.

4.3 Cost. One of the major goals of the modular simulator architecture is to reduce the overall cost of the end product, the training device(s). Cost savings are normally a result of savings in several areas, such as reduced schedules and higher productivity. Based on the results of the Mod Sim demonstration project, engineering estimates from Mod Sim based proposal efforts, and future projections, the Mod Sim architecture is predicted to reduce the cost of training devices. The most significant cost reductions will occur in the following areas:

a. Schedule. The schedule for a modular simulator program will be shorter due to parallel development of segments, shorter integration due to more extensive segment testing, and less retest in system level testing.

b. Reuse. The potential for reuse of segments is enhanced in a Mod Sim implementation. Reuse can occur in two places, reuse of segments in different devices within the same training system and reuse of segments from previous programs. The reuse of segments must be planned at proposal time and managed throughout the program. Reuse will not happen of its own accord. Reuse among devices within a specific training system (e.g.; WST and CPT) may be possible for many application specific segments such as Flight Controls, Flight Dynamics, Flight Station, Propulsion, etc. Reuse from program to program is normally more difficult to achieve for these segments. However, some key segments such as Visual, IOS, and Environment lend themselves to wide scale reuse across multiple programs.

c. Problem Avoidance, Detection and Resolution. One of the major cost drivers in simulation device development is resolution of unanticipated problems. It has been shown that the cost of resolving a problem grows exponentially as you approach the end of system level testing. The Mod Sim approach addresses this issue through two paths; generic front end systems engineering that has anticipated common simulation development problems and rigid segment testing that detects problems early in the test program. Management should ensure that these two areas are addressed properly during program planning and performance.

4.4 Risk. Architecture development and integration risk in a Mod Sim based design effort is very low. Many of the high level program and design issues have either been resolved or a process has been identified to aid in the resolution of these issues. The Mod Sim approach provides a proven method for specifying and identifying the interfaces for all major system components. This reduces the typically high integration risk associated with simulator development. The tailoring instructions built into the generic specifications and interface documents promote risk reduction by identifying specification and interface alternatives along with how the alternatives may affect the design. The "Modular Simulator Engineering Design Guide" and this document identify key design and program issues and methods for identifying and addressing these issues.

Although the Mod Sim approach attempts to mitigate a significant amount of development risk, there will always be program unique areas of risk that the Mod Sim approach cannot anticipate. These areas include integrated avionics, future aircraft technology developments, and challenging requirements for Radar and Electronic Warfare segments. The Mod Sim architecture works well where objects and functionality can be allocated to one and only one segment, but can be more difficult to apply when objects and their associated functionalities are complex and span several of the existing segments. The VNET interface provides a well defined, generic, intersegment communication method that is adaptable to the majority of applications. However, it may not readily address every possible variability existing in todays aircraft integrated avionics and still maintain its generic quality and design flexibility. To provide a solution to this problem, the Mod Sim architecture allows "back door" connections from segments to shared system components such as integrated avionics. Figure 4.4-1 provides an illustration of typical back door connections. Back door connections are not to be used to replace the defined VNET communication path between segments. Back door interfaces are special, alternate communication paths that allow segments to communicate with shared hardware and software components. System components that are connected only to one segment are considered part of the segment and not a back door interface.

The back door interface provides the Mod Sim architecture with an added dimension of design flexibility. The back door interface should be used with care and should be

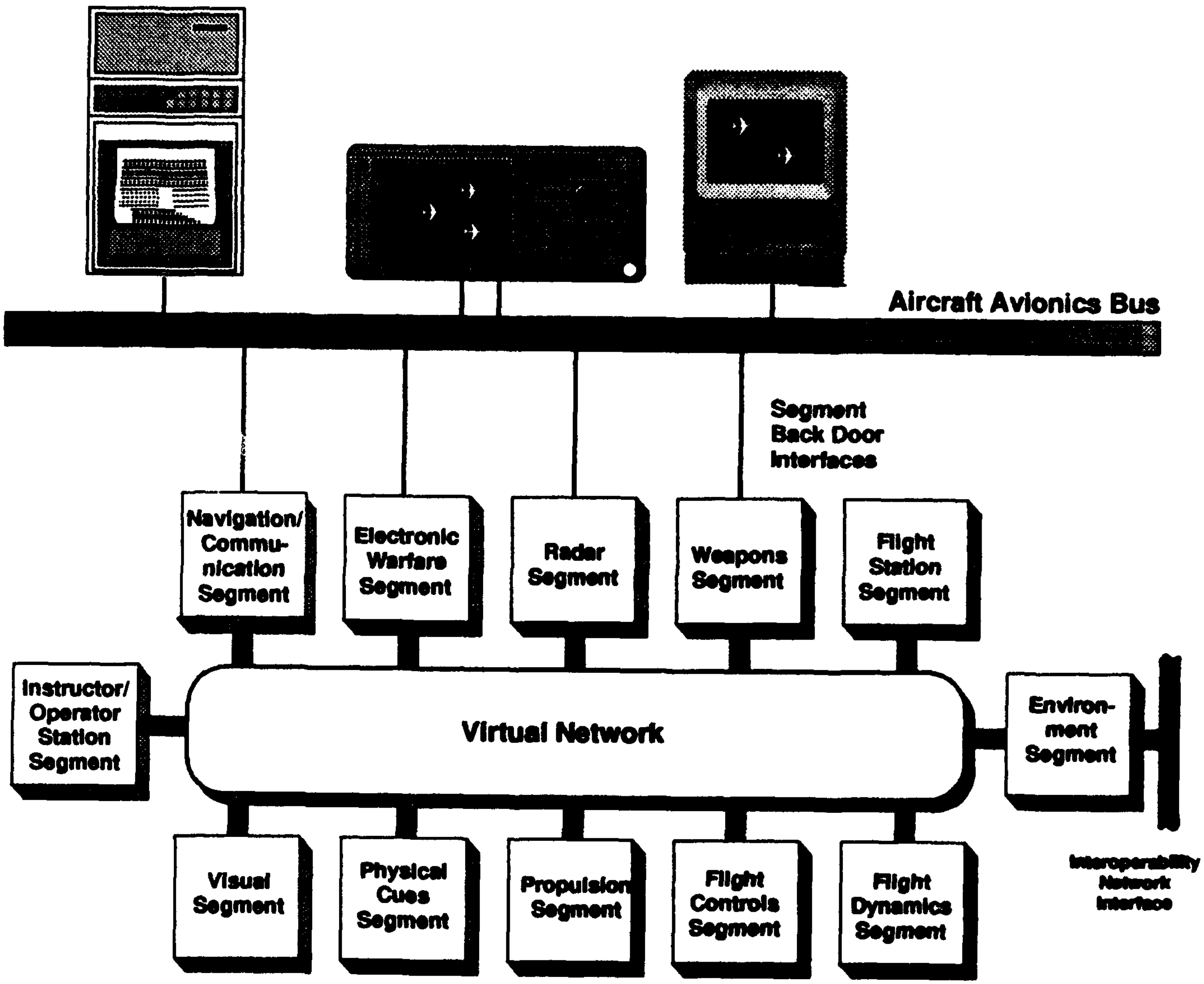

Figure 4.4-1 Modular Simulator System Back Door Interface

adequately documented in the specifications. A plan for the design, implementation and testing of each back door interface must be defined to reduce integration risk. When used effectively the back door interface design option can be very successful.

4.5 Staffing. Prime contractor staffing for a modular simulator based program is determined by the amount of subcontracted effort. Figure 4.5-1 provides a typical staffing plan for a Mod Sim development where approximately 50 percent of the segments have been subcontracted to three external suppliers. In most cases the prime contractor's staff will be primarily systems engineering oriented in the initial phases. Systems engineering will continue to participate in the management of subcontracted efforts throughout the project then ramp up slightly during segment integration. Design staff will fluctuate based on the number of segments which are subcontracted. If all the segments are subcontracted then design staff would be minimal and the systems engineering staff would remain constant or increase slightly to support subcontracts management and review subcontractor deliverables. Total test engineering support will remain similar to a traditional program with the exception that formal test support will be required during stand-alone segment testing to perform segment acceptance. There will be a greater amount of systems engineering work completed by the end of the system requirements analysis phase (SRR time frame). In most cases subcontractors will be identified and will be part of the proposal team. Subcontractor specifications, statements of work, and interface definitions must be defined early in the process to define tasks that the subcontractors can price. At the time of contract award the majority of initial systems engineering work should be complete except for resolution of small issues.

4.6 Subcontract Management and Procurement Philosophy. In the past, simulation devices have been developed entirely by the prime contractor with the exception of off-the-shelf components (computational hardware, linkage, aircraft hardware), some manufacturing tasks (crew station, simulated instruments), and the visual system or motion system. The prime contractor was usually a large company who had expertise in all areas of simulation technology. One of the original concepts of the Mod Sim design was that segments would be built by the contractor who specialized in the technical realm of that segment. This would expand the competitive base for each segment by allowing smaller segment specialists to bid portions of the development effort. The result would be high quality, low cost components leading to a lower total system cost. For example, the propulsion segment may be built by the engine manufacturer or by a company that specializes in propulsion simulation. The engine manufacture knows the engine design, has design criteria in hand, and most likely has done some level of engineering simulation during development of the aircraft engine.

If current trends continue, the government will continue to award large system developments to prime contractors who subcontract significant portions of the development to other suppliers. Also, large training systems development programs

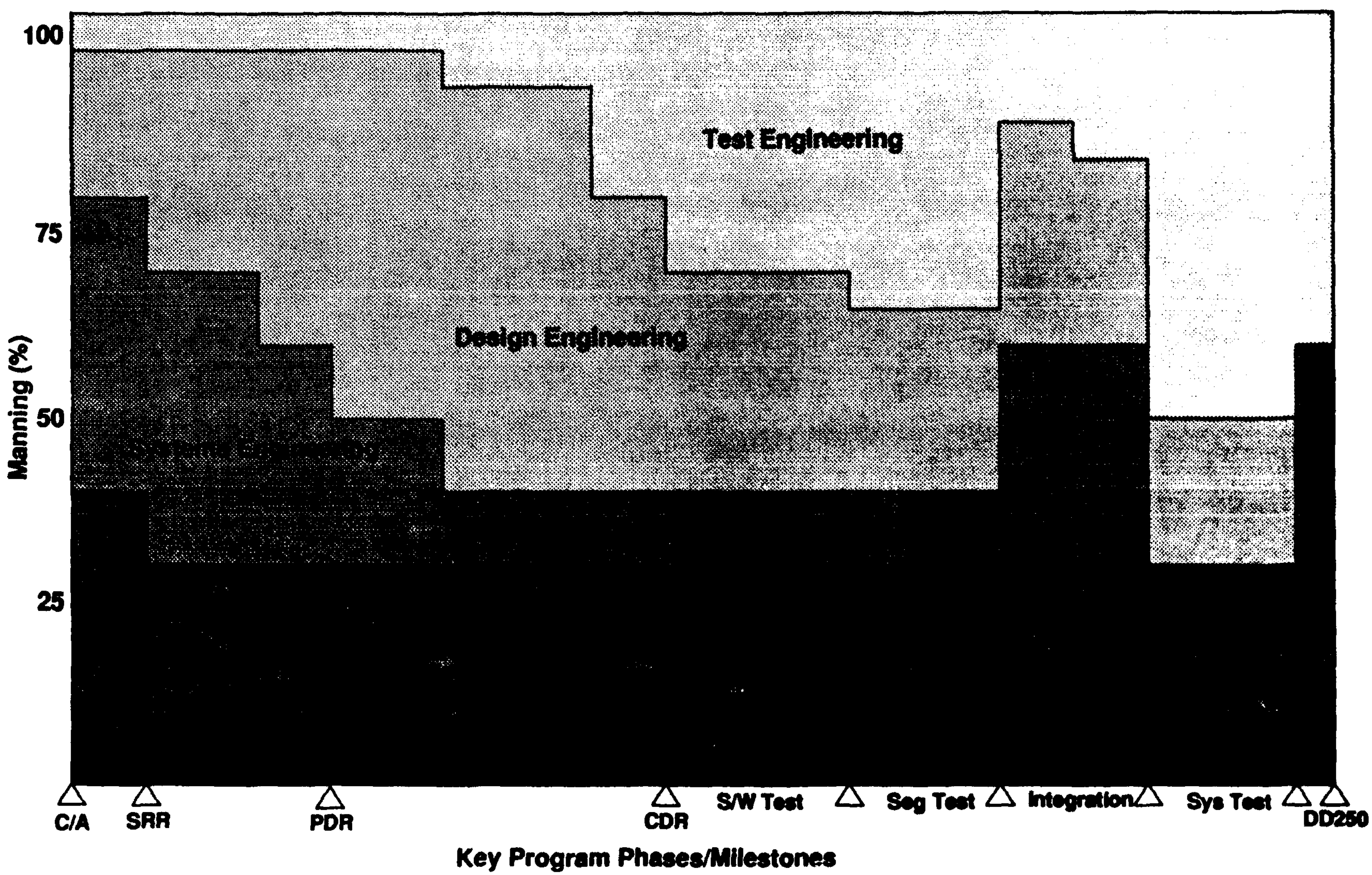

Figure 4.5-1 Modular Simulator System Skill Mix

have increasingly involved teaming arrangements because multicompany teams are often more competitive and share in development programs for mutual benefit. The Mod Sim architecture lends itself to teaming arrangements because the system is pre-partitioned and interfaces for each partition are clearly defined.

The prime contractor will require additional effort to manage the contractual and technical aspects of subcontracting segments to multiple vendors in the Mod Sim approach. This can be a significant effort depending on the number of segment subcontractors. The worst case would be to subcontract each of the twelve segments to twelve subcontractors. This approach would be very costly to organize and coordinate and is not recommended. A more realistic situation would have two to four subcontractors with one to three segments each. This minimizes the amount of subcontractor coordination and increases the design commonality among the segments. The prime contractor should also build one or more of the segments and the items that are common to all segments. This keeps the prime contractor involved in the day-to-day problems of segment development, allowing better understanding of subcontractor problems, and reduces the chances of dual development of components that are common to several segments. There is no best or optimum approach that would apply to every development effort, but an approach similar to that provided in the following paragraphs will fit many applications.

The prime contractor can reduce the overall system cost by leveraging the use of common hardware and software components across segments. Procurement or development of common hardware and software is one method to save cost, maintain design commonality, and improve supportability across the segments. Instead of each subcontractor procuring their own hardware and software it is procured by the prime contractor and provided to each subcontractor as Contractor Furnished Equipment. Another approach, that may be a lower risk for the prime contractor, is to issue a preferred parts list for the project and allow the subcontractors to select their segment hardware from the list. This forces the subcontractors to stand behind their design and reduces the potential of hardware problems returning to the prime contractor for resolution. These approaches reduce the cost of each segment and the system in several ways. First, a discount can usually be negotiated for a quantity buy of hardware. Second, the prime contractor eliminates the cost of the subcontractor personnel required to purchase equipment. The final savings comes in the area of system maintenance and support. Common hardware will cause a reduction in unique spares and support equipment required for the system. The commonality of hardware and software also reduces risk. Normally each hardware/software system has some number of unique bugs or unknowns that cause design workarounds or delays. Using common products reduces the number of these unplanned events in the development of all segments.

An example subcontracting strategy is provided in Figure 4.6-1. It is recommended that the prime contractor or system integrator always build the Instructor Operator

Development Team Member	MSS Segment
Prime Contractor (System Integrator)	Instructor Operator Station Flight Dynamics Flight Controls Propulsion
Subcontractor A	Visual
Subcontractor B	Flight Station Physical Cues
Subcontractor C	Navigation/Communication Electronic Warfare Radar Weapons
Subcontractor D	Environment

Figure 4.6-1 Modular Simulator System Development Team Segment Allocation

Station (IOS) segment. The IOS is the central point of control for the system and the main interface to the user/customer. The IOS segment may have many changes in the user interface throughout the development effort due to customer preferences in display layouts and data. It would be costly to continually flow down these formal changes to a subcontractor. An exception to this approach might occur if a generic IOS was available from a vendor that fit the application. In this case the prime contractor could purchase and integrate the IOS or subcontract the entire IOS segment to the vendor.

It is recommended that the Flight Dynamics and Flight Controls segments be developed by the same vendor due to probable coupling and latency issues. In order to reduce overall risk it is further suggested that the core segments that simulate the airframe, Flight Controls, Flight Dynamics and Propulsion be developed by the same vendor. Since these segments provide the basis for the aircraft simulation, the prime contractor may be best suited to develop these segments. An exception may be for the Propulsion segment which may be best developed by the aircraft engine contractor.

The Visual segment will, in most applications, consist of 90 percent purchased equipment from a visual system vendor. The remaining effort will involve interfacing this equipment to the Mod Sim VNET. The entire segment could be subcontracted to the visual system vendor or the prime contractor could build the Visual segment and purchase the vendor's equipment. Another subcontractor could also perform this task. However, in many cases the prime contractor would pay the subcontractor to buy the visual equipment and then pay to buy the equipment from the subcontractor. Depending on the companies involved and their internal procurement process this may be costly. The most logical subcontractor to purchase and integrate the Visual segment would be the Flight Station segment builder. In most cases, the Flight Station segment builder will be responsible for the physical interface between the crew station and the visual system display(s). This approach may further reduce system level interface problems.

The Flight Station and Physical Cues segments should be provided by the same subcontractor. In most applications this subcontractor would be integrating purchased equipment (linkage, crew compartment, motion system, sound system, etc.) into the two segments. There may be a large number of physical interfaces between these two segments depending on the design. Having the same subcontractor provide both segments will reduce integration problems. The majority of the hardware drawings will be for these two segments in most applications. A single subcontractor would result in a more uniform drawing package.

While not mandatory, it may be wise in most applications to have one subcontractor provide the Navigation/Communication, Weapons, Electronic Warfare, and Radar segments. These segments normally share a great deal of the aircraft avionics. As the avionics become more closely coupled and integrated these segments will provide more data directly to those avionics, causing a greater need for coordination between these segments. A single avionics segment builder would be the best approach for these segments.

The Environment segment could be provided by the prime contractor or a subcontractor without any impact. The Environment segment has a great potential for reuse between applications since it is not dependant on the application aircraft or vehicle. The prime contractor may want to provide this segment in order to construct a reusable asset for future applications. There may also be a supplier who has developed an off-the-shelf product that could be adapted to a Mod Sim Environment segment. For example, a Distributed Interactive Simulation (DIS) compliant device could provide the majority of the Environment segment functionality.

This example is provided to illustrate issues and factors involved in subcontracting within a Mod Sim development and should only be used for general guidance. Each application will have unique requirements, funding profiles, teaming arrangements, and restrictions that will affect the subcontractor to segment distribution. The goal is to make intelligent, informed decisions that will produce a system that is cost effective throughout its life cycle with minimal development risk.

4.7 Development Environment. For purposes of this discussion, development environment refers to the complement of tools, processes and standards employed in the development of the segment software products. The development environment for a Mod Sim will depend on the selection of the host computer(s) and the specific application. In general, the development environment will have the same requirements as a development environment for a traditional simulation approach. However, there are a few considerations that should be made that can help in a Mod Sim implementation. These are:

a. Commonality. The typical Mod Sim implementation will frequently have more than one development environment due to the subcontracting of segment development. The potential exists for each segment developer to have a unique development environment. This tends to be more costly in the long run because an environment must be identified, procured, installed, tested and maintained at each subcontractor facility. A better approach is normally to have a common development environment among the subcontractors and the prime. This saves time and resources during development in several ways. Most training systems require a Training System Support Center (TSSC) to be used after delivery. A common development environment would reduce the cost of the TSSC substantially. Software and data can also be more rapidly reused or transferred among team members without extensive compatibility problems. A common development environment is also a benefit at integration. Changes can be made to correct integration problems in subcontractor segments at the prime contractor's facility with the common environment tools. Finally, development environments normally have problems that require workarounds. If a common environment is used sharing of workarounds can occur. The problem with mandating a single common development environment is that it may not be common to each subcontractor's company standards and processes. This would cause these subcontractors to do extra work to comply with the common environment constraint resulting in higher costs. There is no single answer that will fit all applications. The best approach is to come to a compromise that best suits the requirements of the customer within the constraints of the development team.

b. Segment Tester. A segment tester should be used to perform stand-alone segment testing. The segment tester performs black box testing on a segment at the segment's VNET interface level. The tester provides controlled input data to the segment under test and collects the segment's response or output data for analysis. The segment tester is the mechanism that enables segment integration prior to system integration. The segment tester models the VNET environment emulating all other segments to the segment under test. In this way, a high degree of confidence can be established in segment operation and interface compliance prior to system integration.

A rudimentary segment tester was developed on the Mod Sim demonstration project. This tester was a portable software program that could be hosted on the common development platform used by each segment. This afforded the capability to ship an electronic copy of the segment tester program to each segment developer. The segment developer could then use their development environment to host the tester reducing hardware costs.

c. Network Analyzer. Depending on the implementation of the VNET for the specific application, a special network analyzer may be useful during development and integration. The Mod Sim demonstrator project used FDDI as the media for the VNET implementation. An FDDI network analyzer was used during development, integration and test to resolve communication problems, troubleshoot network error conditions and collect network performance data. Without this tool VNET problem resolution would have been extremely time consuming.

4.8 Logistics Considerations. The Mod Sim architecture can provide either a logistics benefit or nightmare. Theoretically each segment developer is free to design their segment(s) in any manner, using the software and hardware of their choice. The only firm requirement is that they comply with the defined VNET intersegment interface. This freedom has the potential for twelve different computational systems using twelve different software design approaches. Consider the impact to spares and support equipment requirements, skills required for maintenance personnel, computational system service contracts, and the cost of future upgrades to the system in such a case.

The good news is that this catastrophe is avoidable if logistics support is considered during the initial system design phase. The prime contractor and segment subcontractors should agree upon common computational system platforms, software language , software tools (compilers, editors, Software Engineering Environment, word processing, graphics, etc.), design methodology, processes and standards. Selection of these items should not impose excessive design restrictions on any segment developer. Coordinating these decisions early in the program can assure that these potential logistics support problems are avoided.

4.9 Facilities Considerations. It is possible to have a variety of heating, cooling, power, and space requirements for each segment based on unique segment requirements, the selection of segment hardware and available support equipment. Specific facilities interfaces will be highly dependent on the application. For example, in a land based simulator installation there is normally a great deal of flexibility with respect to power, space, cooling, etc. However, for remote or deployable training devices, such as ship based trainers, the options for facilities support are usually limited and in short supply. Another factor to be considered is the commonality, or sharing of facilities among segments. The motion system (Physical Cues segment) and the control loading system (Flight Controls segment) may both require hydraulic power. In such a case, it may be cost effective to share a hydraulic power unit instead of procuring and maintaining a separate unit for each segment.

Facilities interfaces should be developed at a system level and then allocated to each segment. When a certain facilities resource is limited or requires a special interface then those requirements should be uniquely identified and allocated to the segment from the overall system requirement. Even if the use of a resource is not limited, the facility requirements should be identified at the system level and budgeted to the segments. This includes cabinet space allocation and the owner of shared resources Requirements for shared resources should be identified early in the system design phase and documented in the system/segment specifications. Who will provide the shared resource must also be determined so that it may be included in the supplier's pricing. In the example of the shared hydraulic power unit, the prime contractor, Physical Cues segment supplier or Flight Controls segment supplier could provide the unit. The decision must also consider how and where each unit will be stand-alone tested. In this example each of the segments may need access to the unit to accomplish testing or development.

4.10 Lifecycle Considerations. In determining the specific implementation of the Mod Sim approach, it is important to consider the lifecycle cost of the simulation device as well as the development cost. It has been demonstrated that the Mod Sim approach can reduce the development cost of simulation and training devices. However, as discussed previously, there is a potential that if the Mod Sim approach is applied without an overall lifecycle cost management approach there could be shortfalls. It is recommended that the prime contractor and subcontractors focus on designing at a system level as well implementing their individual segments. Since there are no life cycle metrics from past Mod Sim implementations, the Mod Sim developer should review lessons learned from the Mod Sim demonstrator project and develop life cycle cost reduction plans accordingly. The Mod Sim approach can be a powerful tool in reducing overall life cycle and support costs if applied effectively. The Mod Sim approach works particularly well in facilitating upgrades to individual segments without affecting the entire system. Some examples include visual system upgrades (affects only Visual segment), computer upgrades (affects only segment requiring upgrade), and addition of weapons (may affect only Weapons segment).

5. MANAGEMENT LESSONS LEARNED.

There were many lessons learned during the development and demonstration of the Mod Sim architecture. These lessons learned involved all major components of the Mod Sim including the specifications, intersegment interfaces, subcontract management, segment testing, logistics, system implementation, integration and system test. Many of the lessons learned resulted in a change to the specifications or to the basic Mod Sim architecture and approach. The following lessons learned are specifically applicable to the management of a typical Mod Sim project.

5.1 Specifications/Interfaces. The generic System/Segment Specifications and the intersegment interfaces between segments were the major products of the Mod Sim design effort. The following lessons learned were based on experience in using the specifications and interfaces on the Mod Sim demonstration project and the development of technical approaches for several competitive proposals.

a. The version of specifications used for the Mod Sim demonstration project were very tailorable. However, tailoring methods and guidelines were not obvious to companies outside the Mod Sim development team. Boeing incorporated tailoring instructions into the specifications to clarify these tailoring methods and guidelines as a result of this concern.

b. The Mod Sim architecture did not support an interface to allow interoperability with other simulation devices, such as in a Distributed Interactive Simulation (DIS) exercise. Increased emphasis on team training and interoperability of simulators dictated that such an interface was required to make the Mod Sim architecture adaptable to these applications. This interface was provided in the form of a twelfth segment called 'Environment'.

c. The Ada interface used as the data definition for the intersegment interface was not portable to some hardware platforms and compilers. This problem was solved by adding Ada Representation Specifications to the existing intersegment interfaces. This eliminates the portability problems found in most applications.

d. Once a development program is underway it is inevitable that there will be changes to the intersegment interfaces. This change process must be strictly controlled to avoid different versions of the interface during development and test. It is suggested that the prime contractor maintain the master interface and distribute periodic updates to the segment developers. The frequency of the updates should be responsive to segment development testing and integration activities. A configuration control plan should be developed and applied specifically for the interfaces.

e. The Ada based, intersegment interfaces are identified by outputs for each segment. This was done because only one segment may create data, whereas several segments may use the data. Specifying outputs allows for a single definition of the data. The destination of the interface is provided as a comment in the code for each interface. Developers felt that it would be more helpful to have a summary of the inputs and outputs for each segment. A simple software tool can be created to automatically sort through the output interfaces and destination comments to create an Input/Output (I/O) summary listing with minimal effort.

f. The system level modes and states are clearly defined in the specifications. However, during the demonstration project several different interpretations existed among the segment developers regarding the transition between modes and states and the minor details of operation within the modes and states. It is suggested that the specifications be tailored very carefully in this area for future applications. The prime contractor and segment subcontractors should discuss and clarify this area of the specification prior to development to avoid problems during test and integration.

5.2 Subcontract Management. There was a significant amount of subcontracting performed on the Mod Sim demonstration project. The intent of this subcontracting was to make the demonstration similar to a real development project. Boeing subcontracted eight of the eleven segments, about 75% of the system. This led to a significant number of lessons learned in subcontract management including the following:

a. Subcontract management could be a significant cost in the development of future modular simulators. Subcontract management costs will increase as the number of subcontractors and segments subcontracted increases. The Mod Sim demonstration project required a 1 to 1.5 man level of effort to coordinate and manage the subcontracted effort of three subcontractors building seven segments. This was for subcontracted segment developers that were well versed in the Mod Sim approach and design goals. For future Mod Sim implementations, it is suggested that an engineer familiar with the Mod Sim concepts and subcontract management practices be assigned to each subcontractor.

b. Future subcontractors will rely heavily on the specifications for Mod Sim information and requirements. The specifications for any application should be complete and consistent to avoid conflicts between segments during integration. Regular telecons should be held with each subcontractor to resolve open issues if the development team is not collocated. While this lesson learned is not specific to a Mod Sim implementation it is an important issue.

c. It is normally a desire to have all data submissions to the customer exhibit the same format and content. This is very difficult when many of the data items are prepared by subcontractors. The prime contractor should provide boilerplates and examples to each segment developer to increase the commonality of data submissions. This also helps the prime contractor in reviewing and accepting data submissions from the segment developers.

d. After segment acceptance for the subcontractor, the prime contractor is normally responsible for maintaining configuration of the segment developer's hardware and software. A method for accomplishing this should be in place and identified early in the program. This will help to avoid configuration management problems after segment delivery.

e. Several items 'common' to all segments were provided as customer furnished equipment to the segment subcontractors in the Mod Sim demonstration effort. This included hardware, commercial software, customer developed software and engineering support. This was done in an effort to save cost, schedule and avoid duplication of effort. The prime contractor should ensure that commitments to the subcontractors for these 'common' items are upheld. Missing a delivery date for an asset required by all segment developers can have severe impacts to program schedule, cost and contractual obligations. Schedule delays of common components have an impact on all segments in the system.

f. To ensure a smoother system integration subcontractors should remain on the program to support integration of their segment into the system and resolve errors during system level testing. How this arrangement is implemented with each subcontractor will be program specific. For the Mod Sim demonstration, a specific period of integration support was specified and priced. Other arrangements, such as on-call support, may also work for a specific program.

5.3 Segment Testing. Stand-alone segment testing was one of the major concepts to be proven during the Mod Sim demonstration project. The design, development and use of the segment tester tool resulted in many lessons learned including suggestions for extended use of the segment tester during integration and support roles. The following lessons learned are a result of the segment testing activity on the Mod Sim demonstration project.

a. The segment tester tool used to accomplish stand-alone segment test must be based on the configuration managed version of the intersegment interfaces used by all of the segments. This ensures that the segments are being tested against the correct interface specification. If the segment tester tool is provided by the prime contractor, an updated version of the tool should be provided each time the interface changes after the segment tester has been developed.

b. During most of the Mod Sim demonstration project segment testing activities focused on only positive or known test cases. It is suggested that segments also be tested by subjecting them to bad or faulty data to test segment response. This is important because each segment has a responsibility to maintain an error free system. Periodically a segment may get bad data. This data should not cause the segment or system to malfunction. Such test cases assure that a segment's reaction to faulty data can be determined and repaired if required before integration.

c. The prime contractor must be very familiar with the segment and its operation to allow for an effective review of the segment test procedures. Segment testing is a much lower level test in a Mod Sim implementation. Traditionally, flight simulator test procedures are at a system level and can be compared against known system level data such as technical orders and flight manuals. This type of data does not typically exist at a segment level and must be derived. It is suggested that sufficient time be allocated for this activity in the test program. Backup data used for test case development should also be retained for reference by the prime contractor. This data may be stored in Unit Development Folders at the segment level.

d. The segment tester should be a stand-alone tool with its own mass storage capability or a stand-alone software program that is machine independent. The segment's application hardware was used for segment testing on the Mod Sim demonstration project. This was inexpensive with respect to hardware but very cumbersome and inefficient in terms of engineering labor. A stand-alone segment tester would reduce the time required to configure systems for test and aid in the analysis of test data.

e. The segment tester is very useful for segment testing but could easily be expanded into a versatile tool useful during segment development and system integration. The segment tester is basically an emulator of other segments that are communicating with the segment under test. The segment tester could be used to incrementally test a segment during development or could reside on the VNET during system integration to emulate those segments that are not ready for system integration. The segment tester could also be delivered with the system as support equipment to perform the role of a segment/system diagnostics and debugging tool.

f. The development of the segment tester tool is not a short, simple task. The segment tester must be capable of effectively modeling all the message traffic on the VNET. If the segment tester will also serve as an integration and support tool, this added functionality will require additional development effort. Development of the segment tester should start early in the project to ensure its availability for segment test.

5.4 *Logistics/Implementation.* Since the Mod Sim approach does not dictate the internal hardware or software design of the segments, it is possible to have logistics and implementation problems due to internal differences among the segments. The following lessons learned are intended to provide some guidance in these areas.

a. It is important to have regular, informal team Technical Interchange Meetings (TIM) between the prime contractor and the segment subcontractors. The purpose of these meetings are to keep the team communicating and sharing information, ideas, and solutions to common problems. An agenda is required for each meeting but formal presentations and charts are not required. The concept should be a working meeting with a free exchange of information.

b. Every modular simulator should have a network analyzer for the VNET. This device can be used during development, testing and integration for collecting network data and verifying system performance. It can also be used as support equipment for system debug and hardware failure isolation. The network analyzer used during the Mod Sim demonstration project was used to resolve several difficult system integration problems.

c. System start-up can be laborious task if the operator has to start each of the twelve segments one at a time. It is important to implement a single point start-up process for the system. This process, or the requirements for supporting the process should be identified in the specification and tested during segment and system level testing.

d. Unless specific requirements are defined, each subcontractor that builds a segment will probably use a different design methodology for their segments. This hurts supportability at a system level. It is suggested that all segment developers agree on a common design methodology, coding standards, etc. if possible. This will improve the efficiency of integration and future system upgrades that affect more than one segment.

e. Each segment developer may buy or build several software tools to perform basic engineering tasks and data manipulation. Requirements for these tools should be identified early in the project. Once developed, these tools should be exchanged among team members to reduce redundant development and procurement effort. This exchange could occur electronically or at the TIMs.

f. Identification and specification of segment and system level diagnostics and error messages should be emphasized. Requirements for these diagnostics and error messages should be included in the system/segment specifications. The Mod Sim demonstration project did not implement many diagnostics. This caused all phases of test and integration to be more labor intensive. It would have been helpful during test and integration to have a wider range of diagnostic capabilities to locate and correct hardware and software problems.

g. A great deal of the Mod Sim demonstrator development was done using the target, run-time hardware to reduce overall project cost. This required the target hardware to be available very early in the program with a short lead time. The use of commercial off-the-shelf products is the most effective method to meet this type of implementation approach.

5.5 Testing/Integration. System level testing and integration are two program phases that are accomplished more efficiently and with less problems using the Mod Sim approach. The following lessons learned apply to these program phases.

a. System test and integration may cause some rework at the segment test level due to the formal test of a segment as a stand-alone product . It is likely that some segment problems will be discovered during system integration and test. Resolution of these problems may involve modification of previously qualified segments. When the segment is changed, changes to the segment test procedures and retest at the segment level may be required. The need to update segment tests to reflect segment changes during system integration/test depends on the long term plan for support and upgrade of the segment. If a segment is being developed as a reusable asset then the extra cost of regression testing at the segment level may be warranted. Similarly, if future upgrades or modifications will be developed at a segment level or by a supplier, update of the segment tests is advisable.

b. In order to reduce manpower and segment subcontractor support during integration and system testing, it may be useful to have each segment developer supply a user or operator's guide for the segments they develop. This guide should describe the hardware and software configuration, software compilation order and any segment unique characteristics that a user should know to maintain and make changes to the segment.

c. The prime contractor will perform a very important integration role in the development of future modular simulators. They will perform significant management and coordination tasks throughout the development to ensure that all of the segments comply with the Mod Sim concepts. There is no Mod Sim validation criteria that can be used to determine if a particular implementation is a "Modular Simulator System". The key items that would identify a modular simulator would be the identification of the standard twelve stand-alone segments; message based interfaces written as compilable Ada software and a well defined communication architecture or Virtual Network. It is not always important that the entire Mod Sim approach be followed to the letter. What the Mod Sim approach provides is a proven, systematic, structured approach to developing training systems.

6. NOTES.

6.1 Acronyms and Abbreviations.

CDR	Critical Design Review
CDRL	Contract Data Requirements List
CPT	Cockpit Procedures Trainer
DIS	Distributed Interactive Simulation
DoD	Department of Defense
EW	Electronic Warfare
FAA	Federal Aviation Administration
FDDI	Fiber Distributed Data Interface
I/O	Input/Output
IOS	Instructor Operator Station
ISD	Instructional Systems Development
ISWG	Interface Standards Working Group
Mod Sim	Modular Simulator System
MSS	Modular Simulator System
OFT	Operational Flight Trainer
PDR	Preliminary Design Review
PIDS	Prime Item Development Specification
PTT	Part Task Trainer
RFI	Request for Information
RSL	Rediffusion Simulation Limited
SAIC	Science Applications International Corporation
SDIP	Simulator Data Integrity Program
SRR	System Requirements Review
TIM	Technical Interchange Meeting
T.O.	Technical Order

TRR	Test Readiness Review
TSSC	Training System Support Center
UTSS	Universal Threat System for Simulators
VNET	Virtual Network
WST	Weapon System Trainer
XTP	Xpress Transfer Protocol

ACTIVE PAGE RECORD

PAGE NO.	REV LTR	ADDED PAGES				PAGE NO.	REV LTR	ADDED PAGES			
		PAGE NO.	REV LTR	PAGE NO.	REV LTR			PAGE NO.	REV LTR	PAGE NO.	REV LTR
1	-	40	-								
2	-	41	-								
3	-	42	-								
4	-	43	-								
5	-	44	-								
6	-	45	-								
7	-	46	-								
8	-	47	-								
9	-										
10	-										
11	-										
12	-										
13	-										
14	-										
15	-										
16	-										
17	-										
18	-										
19	-										
20	-										
21	-										
22	-										
23	-										
24	-										
25	-										
26	-										
27	-										
28	-										
29	-										
30	-										
31	-										
32	-										
33	-										
34	-										
35	-										
36	-										
37	-										
38	-										
39	-										

REVISIONS			
LTR	DESCRIPTION	DATE	APPROVAL