FOR FURTHER TRAN

12

ESD-TR-78-139

AD A055573

AN AIR FORCE GUIDE TO THE COMPUTER PROGRAM DEVELOPMENT SPECIFICATION

Lloyd V. Searle
System Development Corporation
2500 Colorado Avenue
Santa Monica, CA 90406

Technical rept.,

Nov[redacted]977

105 p.

SDC-TM-5772/008/02

Approved for Public Release;
Distribution Unlimited.

F19628-76-C-0236

Prepared for

DEPUTY FOR TECHNICAL OPERATIONS
ELECTRONIC SYSTEMS DIVISION
HANSCOM AIR FORCE BASE, MA 01731

LEGAL NOTICE

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

OTHER NOTICES

Do not return this copy. Retain or destroy.

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

John C. Mott-Smith

JOHN C. MOTT-SMITH
Project Manager

John T. Holland

JOHN T. HOLLAND, Lt Colonel, USAF
Chief, Techniques Engineering Division

FOR THE COMMANDER

Stanley P. Dereska

STANLEY P. DERESKA, Colonel, USAF
Director, Computer Systems Engineering
Deputy for Technical Operations

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-78-139	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Air Force Guide to the Computer Program Development Specification		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER TM-5772/008/02
7. AUTHOR(s) Lloyd V. Searle		8. CONTRACT OR GRANT NUMBER(s) F19628-76-C-0236
9. PERFORMING ORGANIZATION NAME AND ADDRESS System Development Corporation 2500 Colorado Avenue Santa Monica, CA. 90406		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Command and Management Systems Electronic Systems Division Hanscom AFB, MA. 01731		12. REPORT DATE March 1978
		13. NUMBER OF PAGES 103
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

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

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Program Specifications
Computer Program Management
Computer Program Acquisition Management
System Acquisition Management
Computer Program Configuration Management
System/Software Acquisition Management
Software Documentation
Software Configuration Management

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

This report provides explanatory guidance and examples to support the effective preparation and evaluation of development specifications for computer programs. It is one of a series of guidebooks prepared to assist members of Air Force program offices in managing the software aspects of military system acquisitions. The guide contains an introductory section which outlines the manner in which requirements for computer programs should be developed during the conceptual and validation phases of a system program, including attention to implications for the nature of required technical skills of responsible Government and

over

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

-A

78 06 21 043

20. (cont'd)

contractor personnel. The body of the guide is devoted to descriptions of the proper emphasis and level of content for each section and paragraph of a completed specification when prepared in accordance with the instructions contained in MIL-STD-483 for the computer program development (Part I, or Type B5) specification.

B

PREFACE

This guide is one of a series of Software Acquisition Management (SAM) guidebooks being sponsored by the Electronic Systems Division (ESD) of Air Force Systems Command. The purpose of the series as a whole is to assist members of system program offices in managing the software aspects of military system acquisitions.

Air Force management of the SAM guidebook program is being provided by ESD's Directorate of Computer Systems Engineering (ESD/TOI). Administrative guidance, review, and technical coordination of this guidebook have been accomplished for ESD/TOI by the project manager, Mr. John Mott-Smith.

The SAM guidebook series consists of individual documents issued as they are completed in the form of ESD technical reports. The first seven reports of the series were prepared by members of the MITRE Corporation and published during the period, 1975-1977. Additional guidebooks to complete the series, including this one, are being prepared by the System Development Corporation (SDC) under Air Force contract #F19628-C-76-0236. SDC's manager responsible for the project is Mr. Harvey I. Gold.

Assistance in preparing materials for this guide to the computer program development specification has been provided by Mr. Stanley G. Benson of SDC, to whom the writer is indebted for samples of specification content used in some of the Section 3 illustrations. Of particular note are the functional diagrams illustrated in Figures 9 and 10, which Mr. Benson has developed for his own past uses in documenting system engineering analyses of computer program requirements.

Topics covered, and to be covered, in the SAM series as a whole are identified in the following list. National Technical Information Service (NTIS) accession numbers shown in parentheses identify those topics for which guidebooks have already been published, and for which copies are available through that service.

- Regulations, Specifications and Standards (AD-A016401)
- Contracting for Software Acquisition (AD-A020444)
- Monitoring and Reporting Software Development Status (AD-A016488)
- Statement of Work Preparation (AD-A035924)
- Software Documentation Requirements (AD-A027051)
- Software Development and Maintenance Facilities (AD-A038234)

- Life Cycle Events (AD-A037115)
- Reviews and Audits
- Configuration Management (AD-A047308)
- Computer Program Development Specification
- Verification (AD-A048577)
- Validation and Certification
- Overview of the SAM Guidebook Series
- Software Maintenance
- Software Quality Assurance (AD-A047318)
- Software Cost Estimation and Measurement

TABLE OF CONTENTS

<u>CONTENTS</u> (cont'd)

LIST OF FIGURES

SECTION 1. INTRODUCTION

This guidebook is written to support measures being taken by the Air Force to improve the management of software in system programs. Its general concern is with the topic of how to manage the analysis and documentation of adequately-defined requirements, during early phases of a system program, as a basis for initiating and controlling contracts for computer program development. As a key aspect of that general problem, major emphasis is placed in this guidebook on explanations and examples which are designed to clarify and supplement the brief, but significant, instructions provided in current military standards to govern the format and content of the computer program development specification.

The guidance is addressed, jointly, to contractor and/or Government personnel responsible for developing and preparing the specification, and to personnel of Air Force system program offices who are responsible for its evaluation, acceptance, and subsequent control.

It has been noted in studies of problems encountered with computer program acquisition in systems that success or failure is often a direct function of how well, or whether, the acquisition was initiated on the basis of well defined and properly documented technical requirements. The development of those technical requirements is in itself a complex and lengthy process. Considered very generally, it involves:

- First developing a system specification which includes requirements for information processing functions at the appropriate levels and properly integrated with requirements for the system as a whole.

- Accomplishing proper allocations of system functions to the various system elements to be developed or otherwise acquired, including computer programs.

- Analyzing, evaluating, and expanding those functions and associated performance requirements which have been allocated to individually-identified computer program configuration items (CPCIs).

- Finally, for each identified CPCI to be developed for the given system, formulating detailed definitions of the requirements and documenting those in the form of a computer program development (Type B5, or Part I*) specification.

---

*Among the standard specification types and subtypes prescribed for military uses, the computer program development specification is Type B5. For a given developmental CPCI, the Type B5 and subsequent Type C5 (product) specifications are normally identified as one, two-part specification, of which the B5 is Part I and the C5 is Part II. Hence, the subject specification is referred to, interchangeably, as the Type B5, development, or Part I CPCI specification.

Since all of those successive steps of analysis and definition have a direct bearing on the adequacy of computer program requirements, a summary overview of that total process is presented below as a part of this first, introductory section. However, the focus of attention in the body of this guidebook is on the last of those four generalized steps. That emphasis derives in part from considerations of space and available information, since the process outlined is lengthy, complex, and subject to many normal variations in methods and procedures. But attention to the end objectives is also indicated as a logical "first step" in any further detailing of the generalized process as a whole. Judging from many samples of Part I CPCI specifications which have been examined, the initial need is for a better common perception of the desired end product than appears yet to exist.

1.1 SPECIFICATION ROLES AND OBJECTIVES - GENERAL

Current specification standards for computer programs are designed to be compatible with the Air Force/DoD structure of uniform specifications approved for use in defense system acquisitions. While the standards are potentially useful for broader application, certain aspects of the uniform specification structure and content tend to be both peculiar and significant to the military system practices. In the Air Force, in particular, the specifications are integrated with a spectrum of related management concepts and practices which are typical of the system phasing and environment--notably, pertaining to configuration management, data management, the test program, and contracting.

Air Force practice is to require (a) one performance-level specification prepared for a *system as a whole*, and (b) one specification for each developmental, Government inventory, or commercial "off-the-shelf" end item. For each developmental item, the specification is prepared in two successive parts: Part I (the development specification), defining primarily performance-level requirements to govern the item's development; and Part II, (the product specification), defining detail design and construction of the developed item. This approach to the structure of specifications for a system evolved and became firmly established within the Air Force Systems Command during the early 1960s. A few of the associated principles and implications which are significant to the purposes and orientation of this guidebook are summarized as follows:

- The system specification performs important functions in governing the system program--i.e., the time-phased series of activities and events through which the system is brought into being--as well as in defining the general-level configuration of the required system. This relationship is depicted in Figure 1. It is fundamental to the level and scope of the system specification that its requirements function as the basis for extensive efforts, during the course of the system program, to derive, analyze, and detail the specification of requirements for individual items.

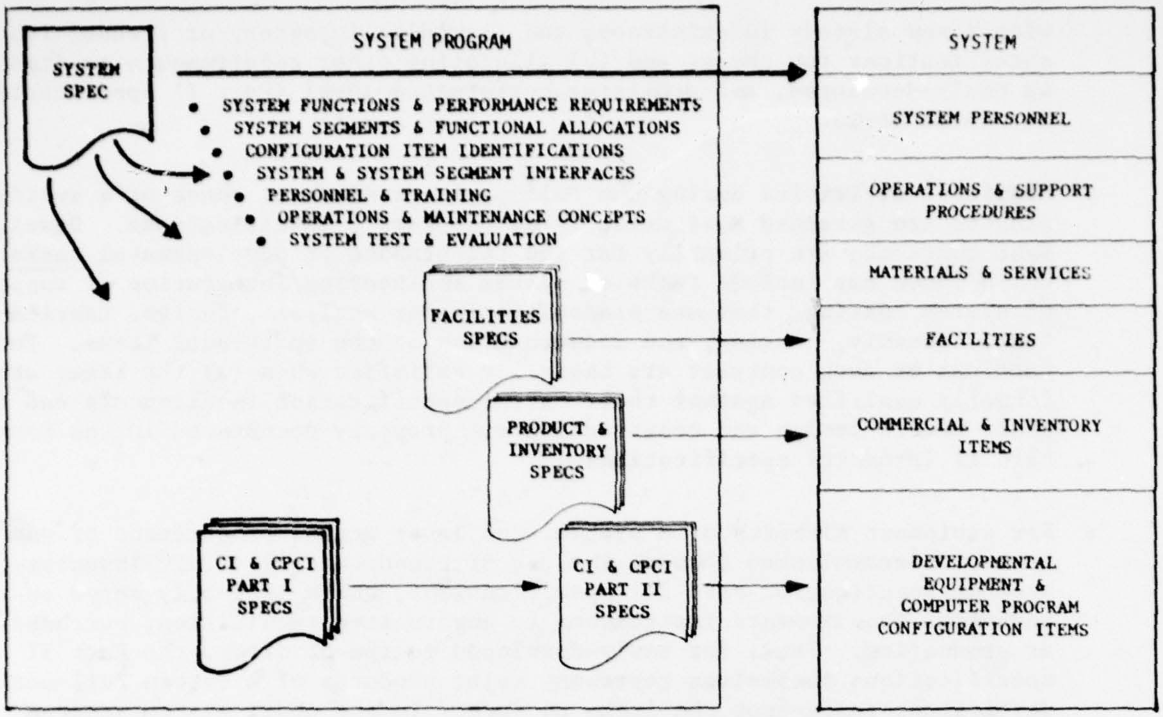

Figure 1. Functions and Coverage of the System Specification (see text).

- The system specification covers broad requirements to be met by both Government agencies and contractors, including significant requirements for system elements--such as personnel and facilities--which cannot normally be acquired through contracts administered by the system program office. It has no direct counterpart at the product level--i.e., there is no "system design specification". Specifications for design and construction are prepared only for those portions of the system that consist of defense materiel items, and only for each item individually.

- In those respects, the system specification reflects established Air Force philosophy that a system is not acquired as an entity, but as a collection of individually-identified end items. While some (partial) exceptions have occurred, that principle is basic not only to the structure of uniform specifications but to a spectrum of the current standards and practices of system acquisition management.*

- A major first step in a system program, following completion and issuance of the initial system specification, is to identify the required end items of defense materiel and to document detailed requirements for those items in the form of item-level specifications appropriate to the class of item and intended approach to its acquisition. This step is accomplished through system engineering studies which result in: (a) matching some requirements

*See amplifying NOTE on this point, p. 11.

with items already in existence, and providing inventory or product function specifications for those; and (b) allocating other requirements to items to be newly-developed, and providing performance-level (Part I) specifications for those items.

- Prominent activities during the full-scale development phase of a system program are governed most directly by the Part I specifications. Development contracts are primarily for the performance of developmental tasks. While those may include tasks of system engineering/integration or support of system testing, they are predominantly for analysis, design, fabrication, assembly, testing, and documentation of the individual items. Those portions of each contract are basically satisfied when (a) the items are formally qualified against their Part I specification requirements and (b) their detail design and construction are properly documented in the form of Part II (product) specifications.

- For equipment elements of a system, the later actual procurement of end items is accomplished through the use of product-level (i.e., inventory, product function, or Part II) specifications, which typically serve as the technical requirements instruments to govern item requisition, purchase, or production. Thus, for newly-developed equipment items, the Part II specifications themselves represent major products of a system full-scale development phase--not the items as such. In the model system program addressed in the DoD 5000-series directives, and for the most part in the Air Force 800-series regulations, the question of whether the Part II specifications are put to use for actual acquisition is a major decision to be reserved for the end of that phase.*

- Except for the last point mentioned, computer program specifications are designed to fit into that process essentially as outlined, including relations to the system specification. The notable difference is that the Part II specification for a computer program item is developed, in conjunction with development of the CPCI, not as a requirements document to be used for subsequent procurement but purely as an "as built" technical description of the developed item. Time and expense required to duplicate (produce) a computer program in quantity for system deployment are typically trivial. One important consequence is that the CPCI Part I specification --the subject of this guidebook--serves the dual purpose of governing the actual item acquisition as well as its development.

*Current policy in this respect tends to emphasize circumstances that are likely to be typical of major ballistic missile or aircraft systems. It would also appear to apply to a major electronic system when the intent is to produce the system as a whole in quantity--although it does not clearly address certain questions of equipment production phasing which arise regarding those as well as for smaller or "one-of-a-kind" systems. It does not clarify, for example, how to manage the production of significant items which may be needed in order to conduct the system test program.

NOTE: Objections have been expressed to statements made above (second paragraph, page 9), on the basis that they contradict a notion that the Air Force acquires each system as an entity. As a whole, the preceding brief discussion of specification roles is intended only to summarize, not to explain nor necessarily defend, relationships which are inherent in currently-documented standards and normal practice. As indicated, there are some partial exceptions. However:

- Acquiring a total system "as an entity" from one contractor is not possible, considering the accepted Air Force definition of a system and the corresponding content of a system specification (see Figure 1 and the first paragraph, page 9).

- In a few programs, contracts have been negotiated against the system specification itself, calling for delivery of an integrated assembly of that contractor's end items (e.g., COBRA DANE). When that contracting method is chosen, it implies significantly less-active PO management at the end-item level during acquisition. Normal procedures in many areas --as regards specifications, configuration management, the test program, design reviews, management reporting--do not apply as prescribed in the standards. Consistently with the level of his contract responsibility for delivering an integrated end product, the contractor retains control throughout development over the selection, design, and construction of component end items, including changes (e.g., note the "record-only" classification of ECPs described in paragraph 4.3.2.5 of MIL-STD-480).

- When CI-level specifications are placed on contract (the normal approach), PO acceptance also occurs at that level. The system specification may also be cited, together with contractor tasks in such areas as system/design integration and interface control. And in some cases the attempt has been made to establish the system specification as having precedence over the CI specifications. Legal rulings resulting from litigations on that point have indicated, however, that the binding order of precedence to a contractor is actually the reverse--i.e., the contractor must comply with the lowest level specification or requirement whenever there is a conflict with any higher-level specification or requirement cited in the contract.

1.2 OVERVIEW OF REQUIREMENTS DEVELOPMENT*

The following description is based on system phasing concepts set forth in current Air Force/DoD documents governing defense system acquisitions. It should be recognized that the process outlined represents only one among various alternative approaches available to program managers. With respect to the conduct of a validation phase, in particular, it should also be recognized that some aspects of the process are presently based more on selective analysis of the regulations and standards than on actual experience, in that very few system programs in the past several years have included a validation phase. That circumstance is believed to be one of the important reasons for the frequent inadequacies of documented system and software requirements.

1.2.1 The System Specification

A system program is initiated for the purpose of providing new or improved military capabilities required by an operational command. The initiation of a given program occurs during the conceptual phase, as the result of an iterative process during which alternative system concepts are examined in relation to documented operational requirements and a proposed system is selected on the basis of estimated performance, feasibility, and cost factors. A simplified summary of that process and its major end products is illustrated in Figure 2, derived from descriptions of Air Force policy and concepts pertaining to the conceptual phase which are provided more fully in AFSCP 800-3 (Chapter 2).

The most prominent technical product of the conceptual phase is an initial system specification, prepared in accordance with MIL-STD-490 Type A format. At that time, its primary purpose is to define the technical portion of documented program requirements to be evaluated and approved by higher headquarters before proceeding to the next phase. After that review and approval, it is established as the functional baseline for purposes of configuration management.**

"Requirements" set forth in the system specification should be concerned primarily with the operational mission functions and associated performance capabilities which the system-to-be-developed must provide, as a total system. Emphasis is placed on defining and expanding those functional and performance

---

* The Life Cycle guidebook (Glore, J.B.; see ref. 9) also includes a summary account of the conceptual and validation phases, but from a quite different point of view; the overlap in coverage with material in this overview of requirements development is relatively slight.

**The document referred to as "the" system specification is rarely a single document in fact. It may consist of multiple volumes; and much of its effective content is nearly always specified by reference to associated system engineering documentation, military specifications and standards, and other documents which it identifies as applicable.

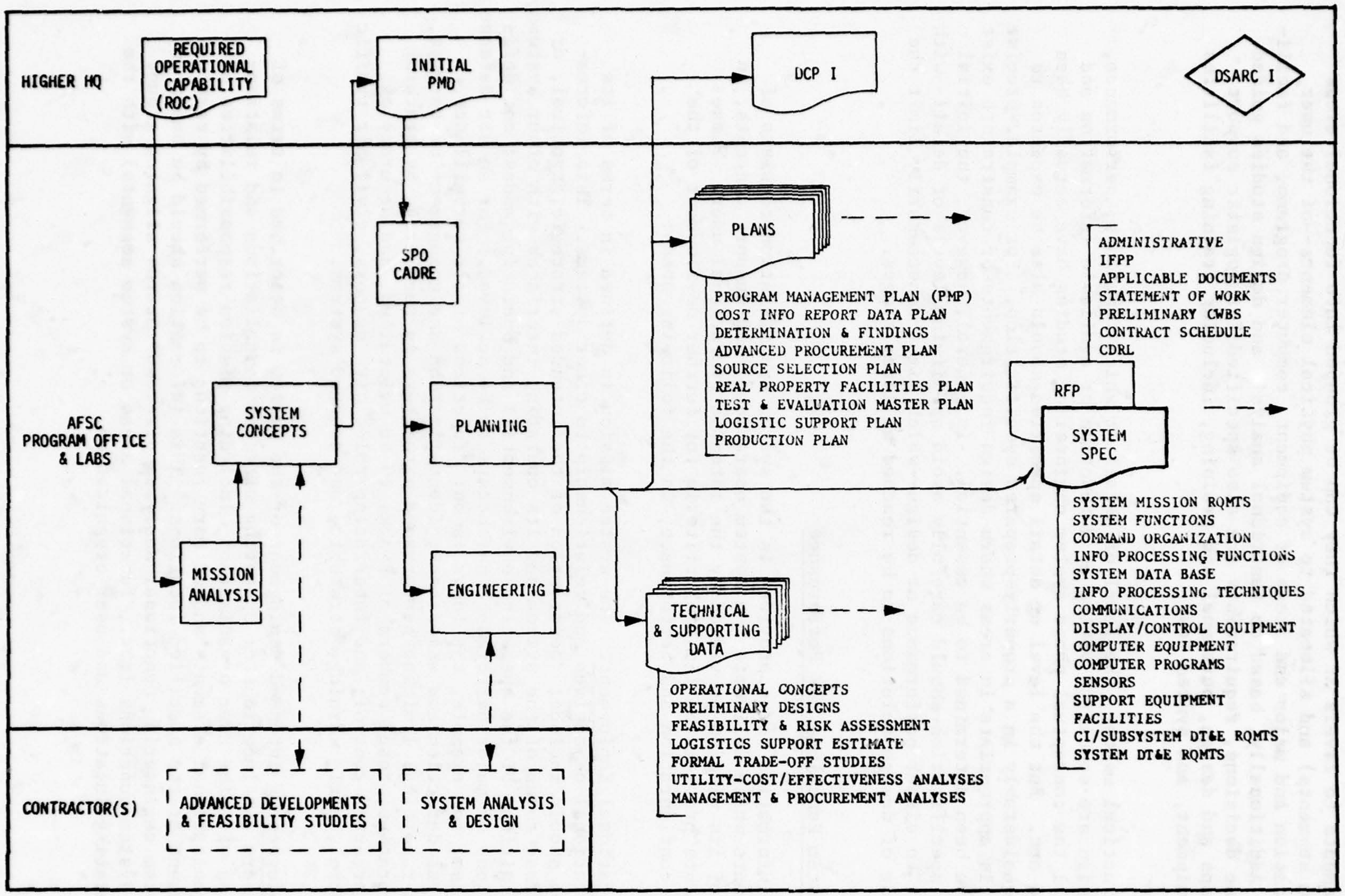

Figure 2. **Synoptic Diagram of Conceptual Phase Events, Activities, and Major Products.**

requirements to levels at which they can be grouped into functional areas (system segments) and allocated to system physical elements--of the user organization and major end items of equipment, computer programs, and facilities. Additionally, based on functional analysis and design studies guided by those decisions, requirements are also specified for logistic support functions and design, personnel and training, including training facilities and equipment, and system test.

As a practical matter, the level of detail at which functions, performance, and design are specified varies as a function of available information and how well the conceptual phase system engineering studies have actually been carried out. But the level of detail specified should also be expected to vary considerably in a properly-prepared specification. For example, precise detail is appropriate in areas where design requirements or constraints exist and have been determined to be essential. In general, however, the initial system specification should carefully avoid specifying levels of detail--with respect to either performance or design--which might unnecessarily limit the latitude of design solutions to be reached at later stages.

a. System Functions and Performance

Basic information to be provided in the system specification consists of statements which delineate the system operational and support concepts, in terms of its mission as viewed by the intended operational user. These statements provide the limiting criteria for further development of the system configuration and performance, in the following areas:

- Operational Employment. The system mission is defined in terms of its operational objectives and relationship to other systems. This information should include: description of the intended strategic, tactical, or defense roles of the system and its operating interfaces with other systems; definitions of the operating environment(s) and operating modes; and definitions of major performance parameters to be achieved. For an air defense system, for example, the focus is on: functions of air surveillance, threat detection and assessment, identification and engagement of targets, and reporting; limiting performance parameters in such areas as tracking accuracies, total numbers of tracks to be maintained, and accuracy of interceptor control; and interfacing roles with sensors, civil air traffic control, early warning, strategic, and command systems.

- Deployment. Intended deployment of the system is described in terms of numbers and locations of operating sites or installations and relationships with the user organization, including mission responsibilities of organizational elements and primary functions to be performed by each element at the specified locations. This information should be based on system engineering functional analyses directed towards aligning groups of system functions (e.g., functional areas or system segments) with the operating locations and user organization.

• Logistic Support. The system specification should summarize the requirements imposed by considerations of supply, maintenance, and support facilities. This information should include: impacts on the supply system, with respect to such functions as introduction of new items, re-supply methods, and distribution; levels of maintenance to be performed (organizational, field, depot) and command responsibilities; and requirements for new or use of existing maintenance facilities and auxiliary equipment.

In the area of operational functions and requirements for information processing aspects of a system, the pacing criteria tend to be matters of required system outputs, which may be in the nature of control actions, decisions, orders, recommendations, reports, or other information to be either used directly by operational command personnel at the system location or transmitted to outside destinations. Hence, much of the conceptual phase study leading to the system specification should have resulted in: (1) identifying those outputs and defining their associated performance requirements with respect to such characteristics as accuracies, completeness, volumes, frequencies, timeliness, and traceability; and (2) deriving similar requirements for inputs and processing functions necessary to produce those outputs. Where a large data base is involved, the data categories and files should have been identified, together with estimated volumes, requirements for site or operating mode adaptation, and special requirements for data collection and generation.* This information should constitute prominent portions of the performance requirements specified in paragraph 3.2.1 of the system specification.

Again, the level of detail will normally vary. Message inputs/outputs internal to the system--e.g., between system segments--may often be defined only to the extent of identifying their existence and general nature, whereas messages input from other existing systems, or required for output to those systems, may be precisely defined at the level of format/content, timing, volumes, etc. for individual messages. Subsequent efforts during the validation phase should result in reducing all of those, including data base definitions, to the lowest level needed as a basis for design (or in some cases, selection) of computer equipment, consoles, communications equipment, and computer programs.

---

*The term "data base" refers here to data which are of interest to the user, and which must later be fully specified in the development specifications for CPCIs. It is typically only one portion of the "data base" eventually specified for developed CPCIs in their Part II (product) specifications. See 3.14.1 herein for a further discussion.

b. System Design

In order to arrive at assessments of feasibility, cost, and schedule, much of the conceptual phase effort is necessarily devoted to studies of design, during which firm decisions are reached with regard to the overall system configuration. Those decisions should be arrived at through the general system engineering process of: analyzing functions and performance requirements for the system as a whole; performing trade-off, feasibility, and/or advanced development studies to identify major system segments and equipments and allocating functions to those; determining design requirements and constraints; and identifying interfaces between segments and with other systems. While some significant design decisions are typically predetermined by policy, economic considerations, or other constraints--e.g., use of a Government-inventory computer,existing facilities, or existing trained personnel--, a systematic application of that analytical process is generally essential in order to establish and verify the technical integrity of the resulting configuration.

"Design" aspects of the system are expressed in the system specification in the form of (1) identifications of the system elements and their structure into system segments, (2) physical constraints such as space or weight limitations, and (3) design and construction standards which apply generally to system equipment or computer programs. The system specification for a command, control, and communications (C^3) system should typically include coverage in the following areas:

- Personnel are to be identified in the form of a preliminary estimate of numbers and types of personnel allocated to system operations, control, and maintenance. These estimates should take into account the planned deployment modes, normal and emergency conditions, and intended duty cycles. Factors of organization, command levels, geographic locations, and operator positions should be specified to provide a basis for subsequent detailed analyses, during the validation phase, leading to Part I specifications for mission computer programs, associated manual and man-machine procedural data, and expanded personnel requirements information.

- Data processing and display equipment should generally be specified in terms of required functional characteristics, at levels which permit latitude for subsequent selection, or approaches to the design of, individual items. The system specification should incorporate schematic block diagrams and associated system engineering documentation which portray the logical and physical equipment configuration and geographic locations. Numbers, types, capacities, and similar requirements should be specified for the central processor(s), peripheral storage and input/output equipment, operator consoles, and special data displays. This information is normally subject to expansion, refinements, and possibly some revisions, during the validation phase. Minimum design and construction standards which apply generally to system equipment are specified in paragraphs 3.3.1 through

3.3.7 of the system specification, e.g., for materials, electromagnetic radiation, workmanship, safety, and human engineering. To the extent possible, these are specified by citing established military standards and specifications.

- Communications capabilities vary considerably, among different C^3 systems, as regards the extent to which they constitute prominent elements of the system--i.e., in the specific sense of whether they are being developed, acquired, or modified under the given system program. Their treatment in the system specification varies accordingly. In the minimum case, existing (common use) capabilities are identified and specified as system interfaces. To the degree that the system program involves the acquisition of specialized and dedicated capabilities, the performance, design, and test requirements for communications hardware and software will constitute correspondingly prominent portions of the system specification as a whole.

- Facilities are typically long leadtime items for which concepts and requirements should be determined very early in the program. The initial system specification should identify all facilities to be used, and should specify: whether existing, or to be modified or newly constructed; nature and intended use (operations, maintenance, training); acquisition approach (military construction program, or other); and required facility characteristics which are essential to be known as the starting basis for validation phase development of detailed requirements and planning for the acquisition of other system elements, particularly for system equipment and personnel. The detailed requirements for those other elements, in turn, should then be reflected in performance-level specifications for facilities (MIL-STD-490 Type B4) to be completed during the validation phase.

- Computer programs should generally be identified in the initial system specification in terms of types of functions to be performed--e.g., operational, simulation, maintenance-diagnostic, and other support functions. Specific individual items may be identified if they have already been selected or if the identification is relatively obvious, as it might be, for example, for a special compiler or various other off-line computer programs. MIL-STD-483 (Appendix III) provides for adding a "design and construction" subparagraph (3.3.8) in the system specification for citing computer programming language and other standards which have general applicability to all computer programs in the system. In line with the expressed intent for paragraph 3.3 as a whole, this subparagraph should be confined to specifying minimum standards which have general applicability, making maximum use of references to approved military standards and specifications. In accordance with general policy for the system specification, it should especially avoid imposing constraints which might unnecessarily limit the latitude of later design solutions.

System segments* must be identified in the system specification, together with information in the areas listed below. Allocations of performance requirements, including interfaces with external systems, should be relatively complete and definitive in the initial system specification. Definitions of inter-segment interfaces and identifications of configuration items are normally subject to significant expansion and refinement as a result of continued system engineering studies to be conducted during the validation phase.

- Each segment is identified, normally by a generic name (e.g., Communications Segment, Command Center Segment), and system characteristics specified in paragraph 3.2 of the specification are allocated among the segments. Allocations consist of (1) apportioning some requirements to two or more segments, such that the sum of the allocations is equal to the total requirement, and (2) specifying requirements peculiar to each segment. The latter may consist of system requirements specified in paragraph 3.2 which are allocated in their entirety to the segment (usually by reference), plus some requirements peculiar to the segment that may not have been specified for the system as a whole. Design and construction standards specified in paragraph 3.3 of the system specification are not included in those allocations, since they apply to all segments.

- Functional interfaces are identified for each system segment and defined to a level of detail which is adequate to permit concurrent and compatible further development of the segments during the validation phase. Interface definitions are derived jointly from the system functional flows and allocations of system functions to the segments, including allocations of interfaces with external systems when they affect an individual segment. For information processing elements of the system, the most prominent interfaces to be identified (and defined to the level that they are known at the outset of validation) are the message inputs and outputs, among segments and with other systems.

- Configuration items of equipment, facilities, and computer programs (see above) are identified and listed for each system segment. In the initial system specification, these lists will normally be provided in terms of generic names for the items (e.g., central processor), emphasizing items of major significance, and will normally be incomplete with respect to both the identified items and quantities. **At the end of the validation phase, the lists should be complete with respect to identifying numbers, approved nomenclature of each item, and quantities required for the full-scale development phase.**

---

*The terms "system segment" and "functional area" are being used for purposes of this description as being essentially equivalent. See paragraphs 3.1.1 and 7.3 of ref. 10 for a further discussion of these terms.

1.2.2 The Validation Phase

The validation phase is characterized in AFSCP 800-3 (Chapter 1) in the following summary terms:

> **"During this phase, major program characteristics are validated and refined, and program risks and costs are assessed, resolved, or minimized. A ratification decision is sought when the confidence of success and cost realism becomes high enough to warrant progression to the next phase..."**

Those are the same objectives that were formerly attached to the contract definition phase, prior to the advent of current DoD 5000-series directives and Air Force 800-series regulations. The major difference is that emphasis is now placed on performing advanced development and prototype testing, in areas of identified high technical risk, as opposed to the former emphasis on purely-paper analysis and planning. The major overall goal, however, is still to advance the definition of the program as a whole to a level which provides a sound and adequate starting point for full-scale system development. Specific objectives to support that goal, as outlined in AFSCM 800-3 (Chapter 3) and elsewhere, are the following:

- Establish firm and realistic performance specifications (allocated baseline) which meet the operational and support requirements.

- Accomplish planning for program office management of the next phase; release RFPs; acquire and evaluate contractor technical and business proposals; and negotiate the full-scale development contract(s).

Current policy also emphasizes flexibility in the methods, intermediate milestones, and approaches employed by program offices to meet those generalized goals. Each program manager is responsible for tailoring the sequence and content of activities to meet the needs specified for his program.

a. Technical Risk

While the activities of hardware proofing and prototype demonstration may be indicated in some cases for electronic systems, the brief overview of the validation phase presented herein does not attempt to include a description of those. If they should be required in a given program, it is assumed that their primary purposes will be to reduce the technical and cost risks associated with entering into full-scale development of the system hardware.

As stated in Air Force/DoD policy documents, prototype demonstration is linked with the "fly-before-you-buy" principle. That principle does not

clearly apply to a one-of-a-kind system for either hardware or software, although it might apply to an electronic system to be produced in quantity, or possibly in modified form to a very large and complex communications network.

Prototype development in the "fly-before-buy" sense is not a realistic practice for software elements of a system, since the quantity production of a computer program does not account for an appreciable part of the total expense of its acquisition. By the time the performance of a computer program can be demonstrated, that particular computer program has already been effectively "bought". While it is conceivable that some form of advanced development to reduce technical risk might apply, there are no known examples of cases in which that approach would appear to be realistic--i.e., for purposes of proving out the software as such. Experience has demonstrated that the risks of entering into full-scale development of software are often real and substantial. But it has rarely if ever been indicated that the problems encountered result from limitations in technical state-of-the-art. Typically, they are matters of inadequate requirements definition and management planning.

Hence, the brief description provided below outlines a validation phase as it might be conducted to alleviate those latter, major problems. To that end, it does not attempt to address the various complications which are necessarily introduced if a given program should also happen to involve hardware proofing and/or prototype demonstration and testing.

b. Overview of Events

Major technical activities and events during the validation phase are depicted in Figure 3, emphasizing activities which would normally be accomplished by one or more validation phase contractors. The period shown would be preceded by a subphase during which RFPs are prepared and issued, contractor proposals prepared and submitted, the source(s) selected, and contract(s) awarded. It should normally be followed by another subphase devoted to evaluation of products, negotiation of full-scale development contracts, and review and decision by higher headquarters.

Within the period shown, the diagram summarizes the following points:

- Objectives during the first part of the contracted validation phase are to analyze and expand the definition of requirements at the system and system segment levels, verifying system mission and support functions and refining their allocations to system personnel, facilities, and configuration items. This basic system engineering effort continues throughout the phase, resulting in expansion/refinement of the system specification and supporting system engineering documentation.

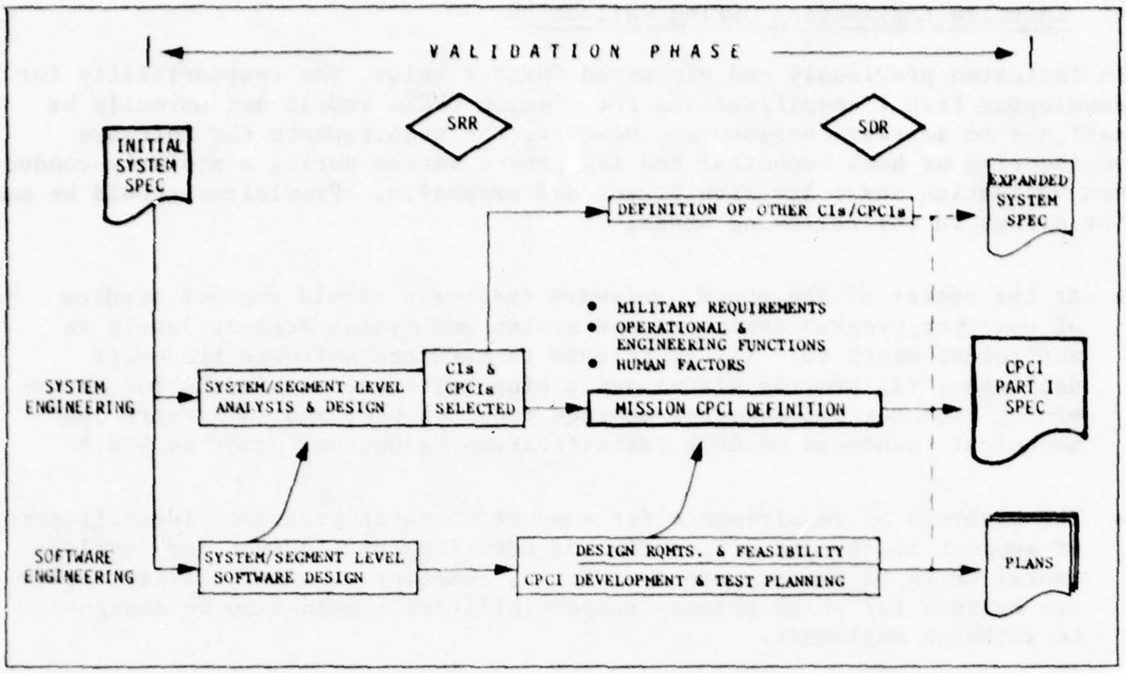

Figure 3. Summary Diagram of Validation Phase Activities and Events.

- Following the system requirements review (SRR), the focus of effort shifts to developing detailed definitions of requirements for individual configuration items and documenting those in the form of development (Part I) or other specifications appropriate to the type or class of each identified item.

- A system design review (SDR) is held as a final review prior to submittal of validation phase products to review and assess their validity and completeness.

- The mainline technical effort, throughout this phase, is at the system engineering level--i.e., basically interdisciplinary, and focused on system compatibility of requirements documented in the CI development specifications as well as in the expanded system specification.

- Software and various hardware (component) engineering specialists provide essential inputs and support, and bear major responsibilities for design studies upon which to base plans, schedules, and costs for the ensuing full-scale development phase.

c. Software Engineering During Validation

As indicated previously and discussed further below, the responsibility for developing Part I specifications for mission CPCIs should not normally be assigned to software engineers. However, the requirements for software engineering at both technical and management levels during a properly-conducted validation phase are significant and extensive. Provisions should be made for effort in the following areas:

- At the outset of the phase, software engineers should conduct studies of computer program design at the system and system segment levels in sufficient depth to: (1) contribute to hardware/software trade-off decisions; (2) provide sizing and timing estimates as a basis for determining computer and computer storage requirements; and (3) assure the technical soundness of CPCI identification/selection, prior to SRR.*

- The analysis of requirements for support computer programs, identifications of support items, and preparation of Part I specifications for developmental CPCIs in the support area (e.g., compiler or other utility tools) are matters for which primary responsibilities should also be assigned to software engineers.

- It should be recognized that it is usually necessary to develop a design approach to each proposed CPCI in parallel with the development of its Part I specification. While the level is likely to vary, it may be indicated in some cases that the design for a major CPCI should be studied in sufficient depth to yield a first approximation to the overall CPCI design to be later reviewed at the preliminary design review (PDR). Direct documentation of that design should never be included in the Part I specification; and it is not clearly called for in other deliverable documents at the end of validation. However, the fact that it has been accomplished should be reflected in a variety of related validation phase efforts and products. As examples:

  (1) In support of the Part I specification development, sufficient design studies should be accomplished to verify the design feasibility of performance requirements, and in some cases to estimate the cost-effectiveness of alternative requirements being investigated by the Part I specification development team.

---

*Failure to fully appreciate the fact that CPCI selection is an important design decision, which necessarily precedes the development of Part I specifications, has been a source of serious problems. See ref. 10, paragraph 2.2.

(2) The computer program development plan (CPDP; DI-S-30567) and formal CPCI test plan (DI-T-3703), jointly, require descriptions of the approach, methodology, schedules, and levels of manpower required during full-scale development for CPCI design, development, and testing. To be realistic, that planning must be based on preliminary information in such areas as the CPCI structure into CPCs, allocations of functions to CPCs, the planned sequencing of CPC coding and assembly, and sizing estimates for individual CPCs.*

d. Development of the CPCI Part I Specification.

It has been stated above that development of the Part I specification for a mission CPCI is a system engineering activity. As such, it is not a process for which specific approaches and techniques are prescribed in any current standards. That situation exists, in part, because of the inherent difficulties involved in standardizing management procedures for technical development--particularly at the levels which demand creative, new products--and in part because it is Air Force policy to emphasize control of the technical aspects of a system program at the level of objectives and general procedures, as opposed to detailed methodologies.

At a general level, the process for a mission CPCI can be described as one of: progressively identifying the system functions and subfunctions allocated to the CPCI; examining mission requirements and modes of operational usage; performing trade studies (among functional alternatives) and time line analyses when indicated; and establishing detailed performance requirements for each function and subfunction. A few characteristics and rules which can be stated for that process are summarized as follows:

- The analyst's direct purpose is to develop, verify, and document detailed requirements for the mission computer program, at the levels described in Section 3 of this guidebook. In the course of that activity, however, he must also develop and verify detailed requirements for manual, man-machine, and equipment functions associated with that computer program. His (or their) job includes assuring that requirements and plans for command

*Those plans should not normally be approved by the procuring activity until they are later updated and expanded to reflect the results of a successfully-completed PDR. Even then, "approval" should be confined to acceptance of their delivery against the CDRL and acknowledgement of their compliance with requirements of their respective DIDs. As for the technical design reviews themselves, procuring activity acknowledgement of compliance should be accomplished in a manner which does not constitute approval of the design as such (see Section 6, "Notes", in MIL-STD-1521A).

actions, operator procedures, and manual or automatic input/output devices are all fully integrated with requirements for the computer program operation.

- One technique which has been found useful is to develop and document a series of scenarios ("operational system description") detailing each step and event for each mode of mission operation--and to ask the questions of What, How, When, How Urgent, etc. associated with each step, iterating that process until the questions are resolved and the necessary level of definition has been reached. These and related system engineering documents--e.g., in the form of functional flow block diagrams, time line or trade study reports, and requirements allocation sheets--function primarily as interim working tools, which aid in developing and verifying the requirements and other information to be documented separately in the CPCI Part I specification, inputs to the expanded system specification and development specifications for equipment items, and operator task analysis data.

- To perform those tasks effectively, the analyst's major orientation is necessarily towards the user's mission operations. His real concern is to develop a progressively-detailed definition of how those operations will be supported by automated functions, and to continually evaluate the results from that point of view.

- That process should start with the assurance that it is feasible to design a computer program which will perform functions within the given general scope, based on system-level design studies completed prior to SRR; and it should be supported by further assessments of design feasibility performed by software engineers for the given CPCI (see 1.2.2,c), as necessary during the process. But the mainline activity of the Part I specification developers should otherwise largely ignore matters of computer program design, in the interests of formulating a comprehensive and definitive statement of user/procuring activity requirements.

Thus, while the approach and techniques required to develop the Part I specification for a CPCI may be characterized as being those of system engineering, because they involve multiple disciplines and multiple classes of system components, it also tends to be true that the major and generally indispensable ingredient is the matter of expertise in the user's operational functions. In general, the <u>focal</u> concern of the analysis team must be with the detailed information processing needs associated with those operational functions--whether they be air defense, tactical operations, weapons control, space communications, interstellar navigation, configuration management, petroleum refining, or automated rapid transit.

By providing a more expanded description of proper content for the completed specification than has previously been available, this guidebook is intended to help alleviate one factor in the difficulties which procuring activities have encountered. It is a matter of experience that a good Part I CPCI specification is relatively rare--good, that is, in the sense that it actually contains the scope and levels of information specified, and provides a sound basis for initiating development of the CPCI. While it is hoped that the situation will be improved through better guidance and training in what the specification should contain, there are other, related factors which should also be recognized as important contributors to the prevalent difficulties. A few of those are identified in the following comments.

- Requirements definition for a complex operational computer program is a tedious and time-consuming task, which results in nothing immediately visible except paper. To be performed properly, it requires high priority emphasis on the part of top-level management to provide the qualified operational/system engineering personnel, adequate funding, time, and firm insistence upon a satisfactory product before permitting a system program to proceed further downstream. Those conditions rarely if ever exist; and they are not likely to exist until a validation phase, or equivalent effort, is planned, scheduled, and funded for that specific purpose.

- There is a widespread tendency to assume that Part I CPCI specifications should be prepared by software engineers--since they are, after all, specifications for computer programs. However, as outlined above, they are specifications primarily of what the CPCI must accomplish for the user, not of how the computer program is to be designed and coded. By training and interest, the software engineer is naturally focused on the latter--often, grossly at the expense of the former. While there are exceptions--e.g., where the software engineer has acquired adequate knowledge of user requirements through experience in the given applications area--the more frequent result is a long and costly "evolutionary" process through which the initially-developed CPCI may eventually become reconciled with the user's actual needs.

- System engineering is often thought of as a single, structured discipline having its own body of technical knowledge, tools, and techniques. While this is true to a degree, the concept is also subject to significant qualifications. Existing system engineering approaches and techniques have evolved and become known principally in the working environments of aircraft and missile systems. Relatively little effort has been devoted to adapting, documenting, and applying those or similar techniques to the information processing elements of C^3 systems. Thus, the generalizable system engineering technology required to support the development of Part I CPCI specifications is relatively undeveloped; and that limitation is further aggravated by the typical need, if the techniques are to be truly effective, to combine their application with intimate knowledge of the specific military operations to be performed by the given system.

SECTION 2. GENERAL GUIDELINES

The purpose of this section is to summarize a few considerations and rules which apply to the development specification as a whole, affecting the preparation and evaluation of all sections/paragraphs discussed individually in the following section (Section 3).

2.1 NOTES ON SPECIFICATION EVALUATION

a. As discussed here, "evaluation" of a CPCI Part I specification is the process of reviewing and assessing a prepared specification for compliance with its preparation requirements, as a basis for approval, authentication, and baselining for configuration control.* Preliminary evaluation of the partially-completed specification should occur at the SDR, late in the validation phase (see Appendix B of MIL-STD-1521A). Formal evaluation at the end of the validation phase should be conducted in accordance with the policies and guidelines set forth in Chapter 2 of AFSCM/AFLCM 375-7.

b. The evaluation process for a Part I specification does not involve a formal audit, comparable to the physical configuration audit which is held to verify the adequacy of a Part II specification. It is normally conducted by an in-house specification review team, chaired by the procuring activity's configuration manager. As appropriate to the system and CPCI, the team may consist of members representing engineering and support management activities of the program office, not-for-profit contractor, AFLC, and the using command. Following coordination of in-house comments, a last phase of the review may be held at the contractor's facility with participation by corresponding contractor personnel.

c. Thus, that pattern provides for a comprehensive review of the specification with respect to its many significant aspects, reflecting the same variety of technical, management, user/operational, and support skills and interests which should have been represented in the specification's preparation. The approach taken by individual evaluators, and the importance of different aspects of the specification, will vary accordingly. As examples:

***Those terms associated with the process are often used as being interchangeable. Distinctions based on the order in which events occur can be useful, however, as follows: Evaluation, when its results are favorable, leads to <u>approval</u>, which signifies acceptance of the preparation activity's delivery of the specification (i.e., as a CDRL item). <u>Authentication</u> occurs when the program manager's signature is affixed to the specification title page; and the specification is <u>baselined</u> for configuration control, by the configuration management office, as a result of authentication.**

- Emphasis in the system engineering evaluation will typically be placed on examining the completeness and integrity of detailed functional descriptions, interfaces, and data base definitions contained in Section 3. The primary objective of this evaluation is to verify (as well as it can be done, analytically) that the CPCI, when later qualified against the specified performance, will indeed satisfy the system requirements.

- Software engineers should examine the entire specification carefully from the point of view of its implications for computer program design, with attention to such factors as feasibility, design requirements/constraints, and adequacy of information needed to govern the subsequent CPCI development and testing. An important objective of the specification is to minimize the need for computer programmers to conduct further research into system requirements, during the course of their development.

- Configuration and data managers should evaluate the specification for compliance with standards for its format and content, including attention to considerations in such areas as organization, style, security, identifying numbers and nomenclature, specification maintenance, and precision/clarity of the stated requirements for purposes of configuration control.

d. This guidebook is designed to present improved criteria for judging whether the specification--i.e., any CPCI Part I specification, in general--contains information in the required areas and at the proper level of detail. In that respect, it should provide an initial "screening" aid for use by those various evaluators in determining whether a given specification meets requirements imposed by the CDRL.

e. For reasons outlined in the preceding section, the situation encountered most often is the one in which the specification is prepared without benefit of sufficient or appropriate system engineering analysis and total effort. Assuming that it does address the proper general subject matter--i.e., functional/performance requirements vs. design--common results are (a) an overall absence of specific and detailed information in the specification itself and (b) a dearth of supporting system engineering documentation. With regard to the specification itself:

- One very gross index which merits further study is a simple page count of the Part I specification in relation to estimated number of instructions in the CPCI. The author made this comparison using data which happened to be available for eight system programs, among which three were relatively successful and five encountered serious problems. For the three, the count of Part I specification pages per 1000 eventual instructions (assembly) ranged from 18 to 25. For the five, the same count ranged from something less than 1 up to a high of 5.

- More direct indicators (but clearly factors affecting the total page count) are provided by examining the content of input, output, and data base paragraphs of the specification. These often consist of little more than brief narrative statements describing the general nature of the data, with little or no detail specifying each data element at the levels required by the preparation instructions.

2.2 SOURCE REQUIREMENTS AND CONVENTIONS

While the primary source of direct instructions for the CPCI Part I specification format and content is Appendix VI of MIL-STD-483, some of the general requirements governing specifications set forth in other Government documents and elsewhere in MIL-STD-483 also apply. Principally, those consist of requirements in the following areas:

- <u>Specification Identification and Style</u>. MIL-STD-490, paragraph 3.2, is the basic source of rules for identifying numbers, general format, and style--covering conventions to be observed in such areas as language style, abbreviations, symbols, paragraph identification, figures, tables, footnotes, definitions, and use of references. Air Force specifications must also comply with supplementary requirements for title pages specified in paragraph 3.4.9 and Figure 1 of MIL-STD-483.

- <u>Security Markings</u>. Specifications containing classified information must be marked and handled in accordance with security regulations specified in the Industrial Security Manual for Safeguarding Classified Information, DoD 5220.22-M.

- <u>Multiple-Volume Specifications</u>. Specific instructions for preparing the CPCI specification in the form of a document series (multiple volumes) are not contained in Appendix VI of MIL-STD-483, but are provided directly in the data item description, DI-E-3119A. Implications for the use of that option are further discussed in 2.3, below.

2.3 SPECIFICATION STRUCTURE AND OUTLINE

2.3.1 General Preparation Requirements.

> (From Block 10 of DI-E-3119A)
>
> 1. The contractor shall prepare a development specification for each CPCI in accordance with the requirements of MIL-STD-483, Appendix VI, as stated in the contract or work statement. When other than Form 1a specifications are called out in Block 16 of the CDRL, Appendix VI of MIL-STD-483 will be used as a guide in the preparation of the specification, employing the specific form from MIL-S-83490 which is set forth in Block 16 of the CDRL. The specification cover page shall be in accordance with MIL-STD-483, Figure 1.
>
> For convenience in describing the minimum essential content, the paragraphs outlined in Appendix VI of MIL-STD-483 (USAF) are arranged in a format which might apply if the specification were to be issued as a single document. However, the specification material required for a large information system is typically too complex and bulky to be published and distributed physically in one bound volume. In this case, the material shall be arranged in separate volumes corresponding to individual functions or as determined by mutual agreement between the contractor and procuring activity to meet the requirements of a particular system. At least one volume of the series shall utilize the complete format and content to define the performance, design, and qualification requirements for the CI as a whole.

Instructions to be noted in the first paragraph above are the following:

- Preparation of the specification must follow the detailed format/content instructions provided in Appendix VI of MIL-STD-483 (specifically, paragraph 60.4).

- The second sentence, referring to other than Form 1a specifications, is meaningful only when it is confirmed and amplified by special instructions provided in the contract statement of work. As yet, there is no uniform guidance available for the application of other than Form 1a specifications to computer programs (cf. reference 10, paragraph 3.5.1).

- Cover pages must comply with both Figure 1 of MIL-STD-490 and Figure 1 of MIL-STD-483. The latter adds the significant requirement for authenticating signatures. For cover page identification of volumes and appendices of a specification issued as a document series, see Note 2.3.2,d below.

The second paragraph of the instructions provides for preparation of the specification as a set of separately-bound volumes. Note the phrase, "... mutual agreement between the contractor and procuring activity...", which indicates that the list of volumes and appendices, by number and title, should be prepared and submitted by the developer for procuring activity concurrence at an early stage of planning. General guidelines are provided in 2.3.2 below. The actual structure in each case should be examined primarily from the point of view of efficient use and maintenance of the completed specification. Sound planning should also result in a volume structure which corresponds with a manageable division of technical responsibilities for its initial development.*

2.3.2 Guidelines for Volume Structure.

Figure 4 summarizes general rules to be considered in planning the structure of volumes and appendices for a large and complex CPCI. Related considerations are amplified in the following notes.

a. General Volume. An appropriate number and title for the volume mentioned in the DI-E-3119A instructions is "Volume 1, General". This volume must contain the full set of format elements identified in MIL-STD-483 (i.e., numbers and titles of sections, paragraphs, and major subparagraphs), plus additional subparagraphs required by content of the given specification.

All requirements pertaining to the CPCI as a whole are specified directly in this volume, except for (a) classified material which may be provided in a separate supplement for the purpose, or (b) detailed definitions of messages or common data base items. Detailed requirements pertaining to those, and to individual functions covered in other volumes, are specified in Volume 1 by referencing the appropriate other volume or appendix. Section 4, Quality Assurance, should normally be provided completely in Volume 1.

b. Other Volumes. A separate volume may be devoted to one or more major functions to detail input, processing, and output requirements for the individual functions or, when indicated, by subfunction. Each such volume must follow the Section 1 through 10 breakdown, including the numbers and titles of all paragraphs specified in MIL-STD-483 which apply to the given volume. Sections or paragraphs "not applicable" to the given volume are so indicated.

c. Appendices. Separately-bound appendices may be used to provide classified supplements or common data definitions, including inputs/outputs and detailed interface definitions (e.g., message formats). With the exception

*Decisions in this area also relate directly to the matter of achieving "visibility" during development. When the structure of data processing functions to be implemented by a given developer is in fact large and complex, use of the multi-volume option is a much sounder approach to achieving visibility of individual functions (and of CPCs, at the product level) than that of forcing an artifical breakout into many small CPCIs (cf. reference 10, paragraph 2.2).

of paragraph numbers (see Note e below), the format of each appendix is optional; it should be designed to suit the purpose and content of each appendix.

Note that all information contained in separately-bound volumes or appendices, with the exception of any Section 6, Notes, is an integral part of the specification and contractually binding. Hence, an appendix is not a suitable form for recording derivations, discussions of alternatives, or procedures for the CPCI operation and use.

d. Title Pages. A volume or appendix number and title must be provided on the title page of each volume or separately-bound appendix in addition to the identifying information illustrated in Figure 1 of MIL-STD-483.* Volumes are numbered in Arabic numerals, beginning with "1"; appendices are numbered in Roman numerals, beginning with "I". Example:

COMPUTER PROGRAM DEVELOPMENT SPECIFICATION

FOR

SEEK DUSK INTERFACE COMPUTER PROGRAM

CPCI No. CG41609

Volume 10. RECORDING AND DATA REDUCTION

[or: Appendix III. TACC MESSAGE FORMATS]

e. Appendix Paragraph Numbers. Within each appendix, the first element of all paragraph or subparagraph numbers is the appendix number converted to Arabic numerals and multiplied by 10. For illustrations, note the numbering of paragraphs throughout the appendices of MIL-STD-483 and MIL-STD-490, both of which conform with the requirements that apply generally to specifications.

Figure 5 illustrates one outline, adapted from an actual case, of the type which should be constructed at an early stage in planning the structure of a multi-volume specification.

---

*Requirements for a "Computer Program Identification Number (CPIN)" are mentioned in AFR 800-14, Volume II (paragraph 6-5). As yet, no specific instructions for AFSC uses of that number are known to have been issued.

CONTENT REQUIREMENTS	VOLUME 1 - GENERAL	VOLUMES 2 - n	APPENDIXES
1. SCOPE	o Standard statement o Organization of the specification o Coverage of this volume	State coverage of the given volume	State coverage of the volume
2. APPLICABLE DOCUMENTS	List only documents referenced in this volume	List only documents referenced in this volume	List documents referenced
3. REQUIREMENTS (through 3.1.1.2)	Complete response in this volume	N/A	Detailed definitions of Messages and/or Classified Supplement When Needed
3.2 Detailed Functional Rqmts. (through 3.2.x.3)	Identify each function by name; reference volume in which specified	Detailed specification of each function in the given volume	
3.2.n *Special Requirements* (incl. 3.2.n.1)	Complete response in this volume	N/A	
3.3 Data Base	o General requirements in this volume o Identify & reference separate appendix(es) for detail	Include data specific to given function(s) and/or reference common appendix for detail	Detailed data base definitions
3.4 System Capacities	Complete response in this volume	N/A	Classified Supplement When Needed
4. QUALITY ASSURANCE	Complete response in this volume	N/A	
5. PREPARATION FOR DELIVERY	N/A	N/A	
6. NOTES	Optional	Optional	
10. APPENDIX	(Integral with volume, optional)	(Integral with volume, opt.)	N/A

Figure 4. Outline Guide for Structuring the Part I CPCI Specification Content into Volumes and Appendices.

OUTLINE OF VOLUME/APPENDIX STRUCTURE
for
System 4XXL Mission Computer Program (MCP)

Volume 1, GENERAL

- Sections 1 and 2
- Paragraph 3.1 - complete for the MCP
- Paragraphs 3.2.1 through 3.2.3 - general level coverage of MCP functions
- Paragraph 3.2.4 - complete coverage of Special Requirements
- Paragraphs 3.3 and 3.4 - (3.3) general; (3.4) complete for the MCP
- Section 4 - complete for the MCP
- Sections 5 and 6

Volume 2, SURVEILLANCE FUNCTIONAL ELEMENT

- Sections 1 and 2
- Paragraph 3.2.1, Surveillance Function
- Paragraph 3.2.1.1, Sensor Data Processing Subfunction
- Paragraph 3.2.1.2, Track Initiation Subfunction
- Paragraph 3.2.1.3, Track Management Subfunction
- Paragraph 3.2.1.4, Identification Subfunction
- Section 6, Notes

Volume 3, MISSION CONTROL FUNCTIONAL ELEMENT

- Sections 1 and 2
- Paragraph 3.2.2, Mission Control Function
- Paragraph 3.2.2.1, Weapons Guidance Subfunction
- Paragraph 3.2.2.2, Air Traffic Control Subfunction
- Paragraph 3.2.2.3, Mission Aircraft Communications Subfunction
- Section 6, Notes

Volume 4, COMMAND FUNCTIONAL ELEMENT

- Sections 1 and 2
- Paragraph 3.2.3, Command Function
- Paragraph 3.2.3.1, Command Support Subfunction
- Paragraph 3.2.3.2, Mission Management Subfunction
- Section 6, Notes

Appendix I, CLASSIFIED SUPPLEMENT

- Paragraph 10 - supplement to paragraphs 3.2.1.3 and 3.2.2.3

Appendix II, INTERFACE MESSAGE CONTENTS AND FORMATS

- Paragraph 20 - supplement to paragraphs 3.1.1.2, 3.2.x.y.1, and 3.2.x.y.3

Appendix III, ADAPTATION DATA

- Paragraph 30 - data base supplement to inputs paragraphs, 3.2.x.y.1

Appendix IV, WEAPONS CHARACTERISTICS

- Paragraph 40 - data base supplement to inputs paragraphs, 3.2.x.y.1

Appendix V, COORDINATE CONVERSION AND TRANSFORMATION EQUATIONS

- Paragraph 50 - supplement to paragraph 3.2.1.1, Sensor Data Processing

Figure 5. Sample Outline of the Volume/Appendix Structure for a CPCI Part I Specification.

SECTION 3. CONTENT GUIDANCE

The intent of this section is to provide guidelines which can be used by both Air Force and contractor personnel in clarifying requirements for, preparing, and evaluating a computer program *development* (Part I) specification against the instructions provided for that specification in Appendix VI of MIL-STD-483 (USAF). To that end, the following subsections are organized to correspond with successive portions of those MIL-STD-483 instructions, and to provide information in the following categories (as applicable to each portion):

- The MIL-STD-483 instructions pertaining to the given section or paragraph(s). Those instructions are copied verbatim from the source and provided in a box immediately below each subsection title.

- An analysis and explanation of the instructions.

- Notes identifying alternative *interpretations*, *techniques*, and/or commonly-encountered problems.

- Examples illustrating proper or alternative ways of stating CPCI requirements in the specification.

Examples are drawn from a number of actual specifications, modified to eliminate classified elements and/or to improve their understandability out of context. They are necessarily limited with respect to (a) coverage of the many different ways in which information under the major paragraphs, in particular, may be organized and presented, and (b) their ability to illustrate essential interdependencies among the various paragraphs which would exist in any properly-prepared, *single* specification.

This guidance is based on the premise that the MIL-STD-483 instructions for preparation of the Part I CPCI specification are well conceived and eminently sound in relation to needs and circumstances of the system acquisition environment. A few modifications are recommended in the text, in the form of suggested CDRL backup instructions for general application. However, those are generally matters of secondary significance which are designed to amplify and reinforce, *rather than change*, the clear intent of the instructions as a whole.

3.1 SECTION 1, SCOPE

> 1.1 Identification. This paragraph shall contain the approved identification, nomenclature, and authorized abbreviation for the computer program. This section of the CI specification shall begin with the following opening phrase: "This part of this specification establishes the requirements for performance, design, test, and qualification of a computer program identified as (insert nomenclature and configuration item number). This CPCI is used to (provide) (accomplish) ...".
>
> 1.2 Functional summary. This paragraph shall contain a summary of the purpose of the specification and a brief description of the overall computer program by major functions (tasks). It shall further identify and summarize the specification content, composition, and intent.

3.1.1 General

Section 1 is to be written in a predominantly informational, rather than directive, style. While the information should be presented concisely, additional material following the specified opening phrase should be provided as necessary to fully inform the reader of relevant facts in at least the following areas:

- The CPCI intended use, including any peculiar conditions or circumstances of general interest and significance pertaining to its use or development.
- The CPCI's major functions, described briefly in terms which are meaningful and informative to the intended CPCI users.
- The organization and coverage of information provided in the specification. This part should include a statement that it is written to comply with (as applicable): Appendix VI of MIL-STD-483 (USAF); the data item description, DI-E-3119A; and backup instructions pertaining to this specification provided with the CDRL, identifying any particular backup instruction which might affect significantly the manner in which the specification is organized.
- If the specification is organized into multiple volumes: (a) List, in paragraph 1.2 of the general volume, all volumes and appendices of the specification by number and title; explain any special rules on which the volume structure is based; and summarize what parts of the total specification are covered in this volume. (b) In each other volume or separately-bound appendix, reference the Scope section of the general volume and summarize the organization and coverage of information provided in the given volume or appendix.

3.1.2 Note on Specifying CPCI Types

In some systems, the requirement exists to adapt a given CPCI--i.e., alter its configuration in relatively minor ways, including adaptation data to be contained in its fixed data base--for effective use at a number of different site locations, or perhaps for other multiple but closely-related applications. In such cases, the CPCI may be identified as consisting of several "types", corresponding to the number of different configurations to be provided. All such types must be identified and defined precisely in the CPCI specification at both the development and product levels. The employment of the type classification is one option, among others, that should have been evaluated and determined during the process of initial configuration item selection for the system (see reference 10, paragraph 2.3).

General requirements for specifying types and other subordinate classifications (addressed primarily to specifications for equipment and materials) are set forth in paragraphs 4.1.2 and 4.3,b of MIL-STD-490. The term "type" is used here for convenience; other terms may be used if they better describe the kind of distinction being made in a given case (see 4.1.2.1.6 of MIL-STD-490). In specifying types for CPCIs, suggested rules are as follows:

- Provide an additional paragraph in Section 1, entitled "Classification", beginning with an opening phrase similar to the following: "The (nomenclature of the CPCI, or abbreviation identified previously in the first paragraph) shall be of the following types, as specified:" That statement is followed simply by a listing of the designated types.

- In any paragraph or subparagraph of Section 3 of the specification which involves differentiating requirements for the individual types: (a) first specify basic requirements for the CPCI as a whole; then (b) provide a subsequent separate paragraph devoted to each type, using additional breakdowns into subparagraphs, tables, etc., as necessary for the given information. In some systems, adaptation data identifications and definitions have been provided in a separately-bound appendix to the development specification.

3.2 SECTION 2, APPLICABLE DOCUMENTS

> <u>Section 2 Applicable documents</u>. The content of this section shall be in accordance with paragraph 4.2 of MIL-STD-490.

Paragraph 4.2 of MIL-STD-490 sets forth requirements for this section which are standard for all DoD specifications and fully applicable to the Part I specification for a computer program. The instructions include policies affecting the function of this section, the handling of references to Government and non-Government documents, and examples of the order in which documents are listed. Points of general interest to be understood and observed include those summarized briefly below:

- The purpose of this section is to provide an organized listing of all documents referenced in other sections (Sections 3, 4, and 5). Documents not cited elsewhere in the body of the specification are to be excluded--i.e., Section 2 is a list of references, not a bibliography.

- Each document listed must be identified specifically and accurately, with respect to document number, title, and current date of issue and/or revision status.

- Each document listed effectively forms a part of the given specification, but <u>only to the extent that</u> specific requirements set forth in other sections of the specification are stated by reference to the given applicable document. The proper statement of any requirement by reference is normally to a specifically-designated method or requirement stated in the referenced document.

- In the event of conflict, it is normal Government policy that any requirement stated directly in the given specification supersedes requirements stated by reference to other applicable documents.

3.3 SECTION 3, REQUIREMENTS

Section 3 Requirements. This section shall contain performance and design requirements for the CPCI. It shall further include the functional requirements for the CPCI and establish those requirements which normally will be verified during category I, or equivalent, test. This section shall also define the CPCI and specify design constraints and standards necessary to assure compatibility of the CPCI with other computer programs and equipments. Performance and design requirements to be included herein shall be allocated from, identical with, or in recognition of, requirements established by the system/system segment specification. Requirements included in the system/system segment specification, which are directly related to requirements specified herein, may be incorporated by reference. Requirements shall be specified to the level of detail necessary to establish limits for design. Quantitative requirements shall be within the three principal subparagraphs included herein. The introductory paragraph shall include a general description of the CPCI and its functions within the system/equipment to which it applies.

Section 3 constitutes the body of the specification as a whole. Its purpose is to set forth all requirements necessary to govern the CPCI design and development in the following major areas:

- Characteristics of interfacing systems, equipment, and other computer programs which affect the CPCI design and operation (paragraph 3.1).

- System functions to be performed by the CPCI, including detailed input, processing, and output requirements pertaining to each function (paragraph 3.2.x).

- Design requirements and constraints (paragraph 3.2.n).

- Detailed definitions of all items to be contained in the CPCI's fixed data base (paragraph 3.3).

The above instructions apply primarily to Section 3 as a whole. Except for the last sentence, they do not specify information to be supplied in a basic paragraph under the section number and title. Since general descriptions of the CPCI are called for elsewhere (e.g., in paragraphs 1.2 and 3.2), the basic paragraph is frequently omitted.

3.4 PARAGRAPH 3.1, COMPUTER PROGRAM DEFINITION

> 3.1 Computer program definition. This paragraph shall, in subparagraphs included herein, specify the functional relationship of the CPCI to other equipment/computer programs and identify Government-furnished computer programs incorporated in the CPCI. General and/or descriptive material may be included in basic paragraph 3.1.
>
> 3.1.1 Interface requirements. This paragraph shall specify, either directly or by reference, requirements imposed on the design of the CPCI because of its relationship to other equipment/computer programs. It shall also include detailed interface definition resulting from contractor analysis and requirements contained in the system/system segment specification. General and/or descriptive material may be included in basic paragraph 3.1.1. Quantitative requirements shall be included in the subparagraphs included herein.
>
> NOTE: Interfaces defined in this section shall include, at a minimum, all relevant characteristics of the computer, such as memory size, word size, access and operation times, interrupt capabilities, and special hardware capabilities. The computer characteristics may be described by references to the applicable documentation and descriptions. If the compiler/assembler is another, or part of another CI, the computer program language(s) to be employed shall be specified as one of the interfaces in subparagraph 3.1.1.2. If the compiler/assembler is a Government-furnished component to be incorporated into this CPCI, it shall be referenced in subparagraph 3.2.4.2. If the compiler/assembler is to be constructed as part of the development of this CPCI, the language characteristics shall be defined under paragraph 3.2 Detailed functional requirements.
>
> 3.1.1.1 Interface block diagram. The relationship of the CPCI to other equipment/computer programs with which it must interface shall be graphically portrayed in this paragraph. This paragraph shall incorporate, in subparagraphs as appropriate, a functional block diagram or equivalent representation of the interface requirements of the CPCI. The graphic portrayal of the CPCI shall be accomplished to the level of detail necessary to identify the functional interfaces between the CPCI and other identified equipment/computer programs.

3.4.1 Paragraph Structure.

This paragraph contains one more level of breakdown into subparagraphs than necessary. An earlier version of the instructions included another subparagraph which was moved, at the time MIL-STD-483 was written, to its present location as 3.2.n.2. Note that there are now two minor errors in the instructions as written: (a) the requirement to identify Government-furnished computer programs in paragraph 3.1 is now redundant with later instructions for paragraph 3.2.n.2; and (b) in the fourth sentence of the NOTE under instructions for paragraph 3.1.1, "3.2.4.2" should read "3.2.n.2".

The intent for paragraph 3.1 as a whole is now expressed completely in the instructions for the first subparagraph, 3.1.1. Hence, it is recommended that the specification provide only the number and title for 3.1, Computer Program Definition, followed immediately by the number and title for paragraph 3.1.1.

3.4.2 Interface Requirements - General

Interfaces to be identified and defined in this paragraph are those with systems and/or configuration items external to the given CPCI (i.e., external to the CPCI only, not necessarily to the system). Interfaces internal to the CPCI are matters of computer program design, which will be: determined later, controlled by the responsible developer during the course of CPCI development, and documented eventually in the CPCI Part II specification.

The primary function of interface definitions, here, is to provide the developer with definitive information about all relevant characteristics of equipment and other computer programs with which the given CPCI must be designed to operate. Considered from the point of view of this specification, responsibilites are summarized as follows:

- The Government, not the developer, is responsible for ensuring that the external items will prove to have those characteristics when the CPCI is eventually put into operation.*
- The developer is responsible for designing and developing the CPCI to be fully compatible with all external items as defined.

The style of specifying interfaces is adapted accordingly. A statement at the outset (in paragraph 3.1.1) that the developer shall design the CPCI to be compatible with all interfaces defined in 3.1.1 and its subparagraphs is sufficient to be directive on the developer. The interface identifications and definitions are then provided (in the subparagraphs) in descriptive terms, representing declarations of intent on the part of the Government.

The instructions above for paragraph 3.1.1 provide for specifying interfaces "either directly or by reference". The use of references for specifying interfaces should observe the following rules:

- Reference may be made internally to other portions of the Part I specification itself. Specific practices for the use of internal reference,

---

*With respect to each given CPCI, this principle holds even if the same contractor is also responsible for an interfacing item. If the interfacing item fails to meet a specified interface requirement, the contractor's responsibility to the Government for that failure is a separate matter. In principle, the given CPCI must meet the interface requirement defined in its own specification, subject only to the resolution of problems via formal processing of engineering change proposals (ECPs). When the Part I specifications have been properly prepared, ECPs required to resolve interface problems among items for which a given, single contractor is responsible should normally be processed as compatibility changes (Code C; see MIL-STD-480, paragraph 4.3.2.1.3).

relating interface messages to inputs and outputs, are further discussed in 3.8 below.

- Reference may be made to other documents listed in Section 2 of the specification. These other documents may include: specifications or other descriptions of the interfacing items; the system and/or system segment specification(s); and selected other references which should also normally be cited in the latter (e.g., JCS standards for message formats).

- References to interface control drawings (ICDs) should not be made. They are not consistent with Air Force policy and are not advisable (see: ref. 1, pp. 1-11 and 1-31; and ref. 10, pp. 78-80).

NOTE: It is reported that one or two current programs are finding need for extensive uses of computer program ICDs during middle and late stages of the development period. Information regarding the prospects for cost/schedule success of those programs is not available. Together with the Air Force standards, this guide necessarily assumes that each PO will require properly-prepared Part I specifications as a prerequisite to orderly management of contract development. When that does not occur, however, more extensive use of ICDs (and various other compensatory measures) may well be indicated, even as a development proceeds into its later stages. As indicated, statements made above are based on policy set forth in paragraphs 1-12 and 1-39 of AFSCM/AFLCM 375-7. A few key considerations relating to their meaning, intent, and validity are summarized briefly as follows:

- ICDs are generated by an interface control working group (ICWG), during the full-scale development phase, in the form of technical agreements among interfacing contractors/agencies. Traditionally, the prominent concern during that phase is with equipment interfaces which (a) have been defined functionally in Part I specifications at the outset of the phase and (b) must be defined precisely at the physical level as the development proceeds. The conventional requirement is that those physical interfaces be "finalized" not later than CDR. Most such ICDs are later referenced in the CI product specifications, together with other detail engineering drawings of CI design and construction. Prior to PCA, they are controlled only by the ICWG, not by the PO's configuration control board (CCB).

  When a program does not have a validation phase (which happens often), many ICDs may be generated at the outset of full-scale development to complete definitions of functional interfaces required in the system/system segment and CI development specifications. Theoretically, those should be completed before CI preliminary design efforts are initiated. Practically, it is recognized that some may be delayed until approximately the time of PDR (see paragraph 2-4,b,(1) of AFSCM/AFLCM 375-7). When completed, they are incorporated into the system and CI development specifications, via ECP, for control during development by the CCB.

- All external interfaces of a CPCI are functional. Since they represent essential, contractual requirements to govern computer program development, they must be incorporated into the CPCI Part I specifications. The CPCI Part II specification does not contain an interface requirements paragraph, consistently with its as-built nature and absence of a contractual requirements role.

- It should be noted that AFSCM/AFLCM 375-7 (in the paragraphs referenced above) does not prohibit the preparation and use of computer program ICDs. It requires only that, when interfaces have been documented initially on ICDs, their definitions be incorporated directly into the Part I specifications, rather than by reference to the ICDs. That requirement is fully consistent with the policy of promoting integrity of the Part I specification as an entity in itself, avoiding duplication of requirements across a multiplicity of baseline documents. It is also consistent with the fact that the most prominent interfaces to be defined are message inputs and outputs. To the degree that definitions of those inputs and outputs are <u>not</u> known and incorporated into the Part I specification at the time it is placed on contract, there is a corresponding absence of information for most of the specification's required basic content.

3.4.3 Interface Identifications

The objective for paragraph 3.1.1.1 is to identify all configuration items and/or external systems having functional interfaces with the given CPCI. A detailed definition for each interface identified here is then to be provided in the next paragraph of the specification, 3.1.1.2 (see 3.5 below).

Computer programs tend to involve external relationships which are often subject to question, in certain areas, regarding whether they in fact constitute interfaces to be identified in accordance with the requirements of this paragraph in the CPCI Part I specification. The following comments address a few of those "gray areas".

a. The phrase "man-machine interface", although meaningful in another context, does not imply that relationships to human operators are to be included among interfaces identified here. In general, interface control is confined to relationships among computer programs, equipment, and/or facilities--i.e., the inanimate parts of a system as a whole whose characteristics are created by system designers/developers and documented in specifications. For computer programs:

- Functional characteristics which affect the CPCI's compatibility with human operators are defined as integral parts of the requirements specified elsewhere in Section 3, primarily in paragraph 3.2 (see 3.12 below for a further discussion of requirements for human performance).

- In all cases, the man-machine relationships are mediated through operator console or other manual input and display equipment. Interfaces of the CPCI with those equipment elements are significant interfaces to be identified and defined in the subparagraphs under 3.1.

b. In a system program, CPCIs are involved in only the first two of the four major classes of interfaces listed below; and both of those are always applicable:

→ Software/Software

→ Software/Hardware

(N/A) Hardware/Hardware

(N/A) Hardware/Facilities

c. Although the general approach to specifying and controlling software interfaces is derived from hardware precedents, certain well-established hardware concepts have also proved to be sources of confusion when applied to software. Questions arise regarding: (1) the distinction between "functional" and "physical" interfaces; (2) the related practice of developing interface definitions at lower levels (physical) during the course of CI development; together with (3) the practice of limiting interface identifications to interacting items which share a common point or region of physical contact. Suggested general approaches to handling those questions are summarized in the following comments:

- All interfaces of a CPCI with both hardware and other software are regarded as functional. They should be defined, <u>in the Part I specification</u> (again, either directly or by reference; see 3.4.2 above), at a <u>level</u> of detail which is sufficient for all purposes of (1) guiding or constraining the CPCI developer and (2) maintaining interface control during the life of the item.

- Interfaces to be defined and controlled for a CPCI are not limited to direct interactions with other items in either space or time. From one point of view, for example, an applications CPCI operating in a given computer interfaces "directly" with only the given computer hardware and other software operating at the same time in that computer. The computer and its resident operating system or other utility software do represent prominent interfaces to be identified in the Part I specification and defined (normally by reference to existing documentation). However, the significant operational functions of the CPCI are to receive and process messages input from remote sources, and to generate messages for output to remote destinations. Thus--for the specific purpose of identifying allocations of responsibility in the Part I specification--the role of the CPCI may be regarded as comparable to that of a mail-order question-and-answer service, in that:

--It interfaces directly with the mediating hardware and software (post office) with respect to certain routine matters of format (like addresses, bulk, rates, timing, and other rules for message transportation); but

--most importantly, it must be capable of responding to message content in a manner which is directly compatible with the remote sources/destinations (customers), involving matters for which the mediating services have neither the responsibility nor the capability to handle.

Hence, as summary rules: CPCI interfaces are identified with the computer and other software operating in the same computer. They are also identified with all other items*, both hardware and other software, which affect characteristics of input or output messages in ways over which the computer and its support software have no control. The latter typically include console or other input and display capabilities, recording devices, communications links, and other applications software operating in different computers.

3.4.4 Example

The content of this paragraph may be limited to the interface block diagram called for explicitly in the instructions. One example is illustrated in Figure 6. A specific format for the diagram is not specified. However, it should be noted that the "identification" is accomplished basically by providing (a) the approved nomenclature of each interfacing item, together with (b) a CI or other number(s) to designate the item precisely. If abbreviated identifications are used in the diagram to conserve space, full identifications (in the form of numbers and approved nomenclature) may be provided either in a separate listing keyed to the diagram or in the next paragraph, 3.1.1.2.

*Interfaces with items that are parts of another system should be identified as interfaces with the other system, not with particular configuration items.

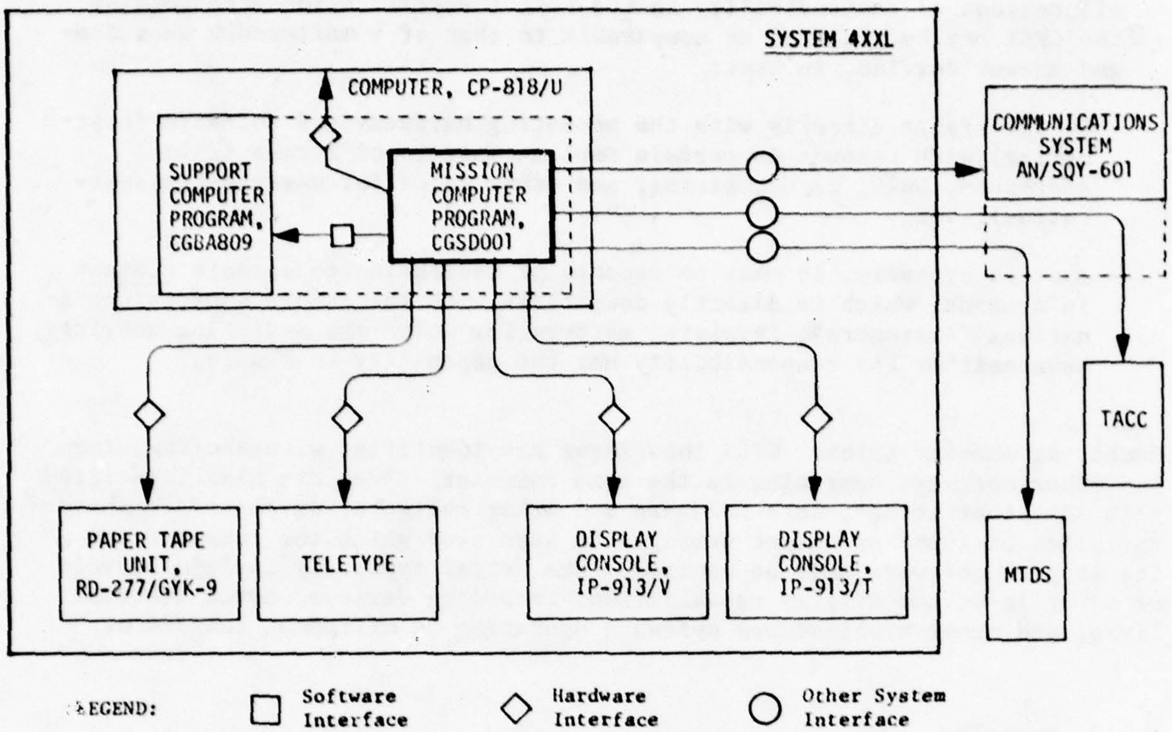

Figure 6 . Sample of an Interface Block Diagram for a Mission CPCI.

NOTE: This example is chosen to illustrate principles discussed in 3.2.3,c above, indicating that:

- **The given applications (mission) CPCI has direct interfaces with the computer in which it operates, and with the resident software "operating system" (O/S; shown as the Support Computer Program).**

- **All interactions of the CPCI with elements external to the computer are mediated by the O/S (as indicated by the surrounding dashed lines).**

- **But effective interfaces exist between the given mission CPCI and other items or systems external to the computer. Those must be identified and defined, in the Part I specification, whenever the design and operation of the given CPCI *will affect, or be affected by*, functional characteristics of those external items or systems.**

3.5 PARAGRAPH 3.1.1.2, DETAILED INTERFACE DEFINITION

> 3.1.1.2 Detailed interface definition. This paragraph shall specify, in subparagraphs as appropriate, the functional relationship of the CPCI to interfacing equipment and computer programs. This information shall be given in quantitative terms with tolerances where applicable to the level of detail necessary to permit design of the CPCI. Functional interfaces shall specify the input/output requirements of the CPCI in terms of data rate, message format, etc. In addition, this paragraph shall specify design requirements imposed upon other equipment/computer programs as a result of the design of this CPCI (eg, operator console equipment, display characters, junction and distribution boxes, terminal boards, etc).

3.5.1 General

This paragraph should be organized into a set of major subparagraphs, each corresponding to one of the other items/systems with which interfaces were identified for this CPCI in the preceding paragraph 3.1.1.1. Suggested general rules for specifying the detailed interface definitions are the following:

- The computer, its standard support (non-functional) software, and peripheral input/output equipment should normally consist of items which already exist prior to the time that development of the given system CPCI is undertaken. Those interface characteristics should be specified in the given CPCI's Part I specification by reference to the applicable documentation of each existing item. In these cases, the major burden of specifying interfaces, here, is on precise identification of the individual items and their documentation, with attention to specific type/model/version designations where applicable.*

- If a given existing item is being modified as a part of this system program in such a way as to affect its interface with this CPCI, the details of that aspect of the interface should be defined directly in this paragraph.

- Interfaces which consist of messages input to the given CPCI from external sources, or output to external destinations, should be defined directly in

---

*Concurrent development of separate CPCIs expected to have complex operating interactions, in the same computer (e.g., interacting applications functions; applications vs. operating system and/or real-time maintenance/diagnostics), is a situation to be avoided if possible. If such elements are assigned to separate contractors, with concurrent phasing of their development during the program, interfaces can easily become unmanageable. If assigned to one contractor, they should normally be combined into a single CPCI.

this paragraph or by reference to other parts of this CPCI specification (see 3.8.2,b). The use of references to other documents, e.g., the system specification, is permitted, but should be held to a minimum (see also 3.4.2 above).

- Interfaces with other items being developed concurrently should be defined at the time the specification is initially approved, as well as they are known; and the nature of any details missing at that time should be described, together with notations that those details are yet to be added --e.g., in the form of "To Be Determined (TBD)" entries. Such TBDs should be replaced by the required detail not later than preliminary design review (PDR). See also the discussions of this point contained in AFSCM/AFLCM 375-7, paragraphs 2-4,b,(1) and 2-5,c,(1).

3.5.2 Examples

As indicated above and further discussed in 3.8.2, message interfaces with external items and systems are also inputs and outputs to the CPCI functions, and the requirements for their detailed definitions are correspondingly similar. For examples of the proper levels of information, see Figures 11 and 15 herein. The minimum information to be specified directly in this paragraph should consist of: (a) identification of the message (within the major subparagraph for that external item/system); and (b) reference to the specific location of its detailed definition.

The following Figures 7 and 8 illustrate portions of the interface definitions that might be provided for the CPCI's interfaces with an operator console. For an existing console not being modified under the given program, this information should be specified by reference to the console documentation. It should normally be specified directly in this paragraph, as illustrated, if the console is being developed or modified concurrently with respect to those manual entry and display features.

3.1.1.2.7.2.1 Interceptor Pilot Simulator (IPS) Console Manual Entry Device. The IPS console manual data entry device consists of two (2) identical input keyboards, each with compatible assemblies for central computer interfaces. The keyboards are located on each side of the Data Display Console. Each keyboard, arranged as shown in Figure 3.8, contains four (4) modules: INTERCEPTOR TRACK NUMBER; ACTION; GENERAL INPUT; and ACTIVATE.

TRACK NUMBER is entered by means of six (6) independent sets of four (4) thumbwheels. Adjacent to each set of thumbwheels is a selector pushbutton to indicate which thumbwheel set is to be read with the action. The GENERAL INPUT module consists of three (3) sets of ten (10) pushbuttons per set. They are used to enter numerical information with the associated action. The ACTION module contains a set of ten (10) pushbuttons for entry of the type of action desired.

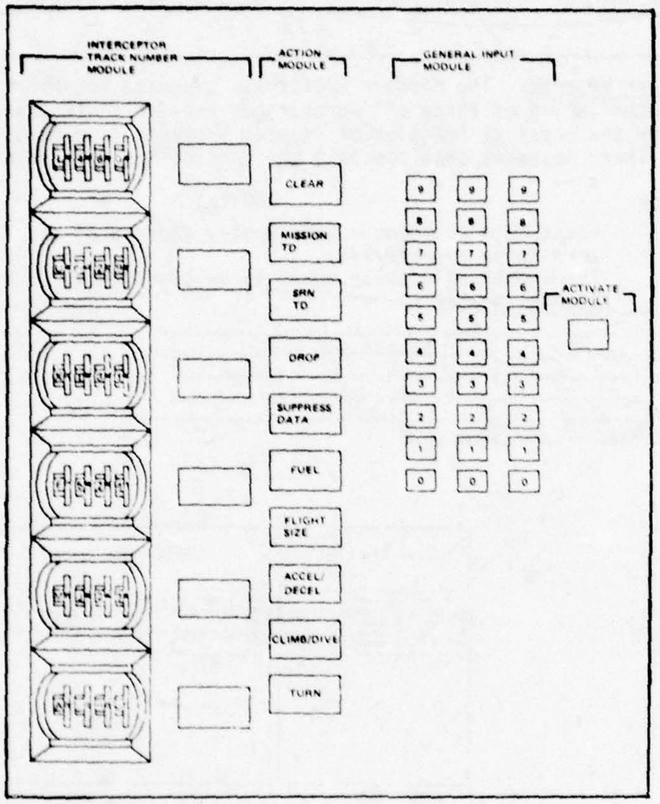

Figure 3.8. IPS Console Keyboard - Module Layout.

3.1.1.2.7.2.2. IPS Console Input Message. When the ACTIVATE pushbutton is depressed, the content of the four (4) thumbwheels of the selected TRACK NUMBER module, the three (3) pushbuttons of the GENERAL INPUT module, and the depressed pushbutton in the ACTION module are assembled and transferred to the central computer, as an interrupt message, in the following format:

	THUMB WHEEL NO. 1	THUMB WHEEL NO. 2	THUMB WHEEL NO. 3	THUMB WHEEL NO. 4	ACTION MODULE	GI MODULE NO. 1	GI MODULE NO. 2	GI MODULE NO. 3
1	2-6	7-12	13-18	19-24	25-30	31-36	37-42	43-48

Figure 7. Sample Manual Entry Device Interface Data.

3.1.1.2.7.1 Situation Displays. Functional relationships of the Mission CPCI to the Situation Display hardware elements, including interactions with the Operating System CPCI which affect this interface, are outlined in Figure 3.10. The figure depicts the information flow, I/O control, and Mission CPCI detailed interface requirements associated with generating and presenting the Situation Display information.

3.1.1.2.7.1.1 Functional Description. The Mission CPCI shall: (a) generate Situation Display information, utilizing stored mission data; (b) structure display messages; and (c) determine the routing of the display messages through the hardware elements, in accordance with the display specification data specified in 3.1.1.2.7.1.4 below. The Operating System CPCI accomplishes the physical transfer of the Mission CPCI display messages to the Display Drum System as specified by the Mission CPCI in the I/O Request Message. The Drum Controller routes the display messages to their assigned Display Drum channels. The Display Drum transfers all display messages to refresh the CRT Displays. The routing of individual display messages to the

3.1.1.2.7.1.2 I/O Request Message. The Mission CPCI shall generate Request Messages for Display Drum outputs, each consisting of three (3) words, that provide instructions to the Operating System CPCI for the physical transfer of display messages to each of the eleven (11) Display Drum channels. These messages shall contain the information defined below:

WORD	BIT LOCATION	CONTENT
1	1-18	First Word Location - Core Memory address of the first display word to be transferred.
1	19-28	The number of display words to be transferred to Display Drum Channel Number 1

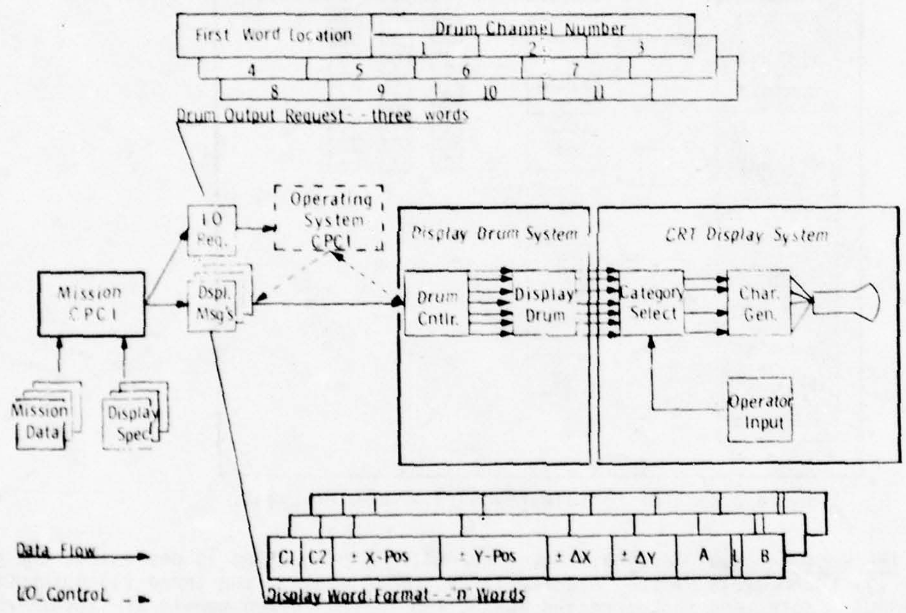

Figure 3.10. Situation Display Interface Diagram.

3.1.1.2.7.1.3 Display Messages. Mission CPCI-generated display messages, consisting of one or more display words (one word is required for each character to be displayed), shall provide the coded instructions necessary to drive the CRT Display System equipments. A standard display word format shall be used which shall contain the following information:

BIT LOCATION	CONTENT
1-3	C1-Display Character Set First Octal Digit
4-6	C2-Display Character Set Second Octal Digit
7-16	±X-POS - Display Coordinate X Component
17-26	±Y-POS - Display Coordinate Y Component

Figure 8. Sample Display Interface Data (cont'd next page).

3.1.1.2.7.1.4 Display Specification Data. The Mission CPCI shall generate Situation Display messages in accordance with the following display requirements:

a. Display Character Set. Character codes for the C1 and C2 elements of the standard display word shall be derived utilizing the Display Character Set matrix illustrated in Figure 3.11.

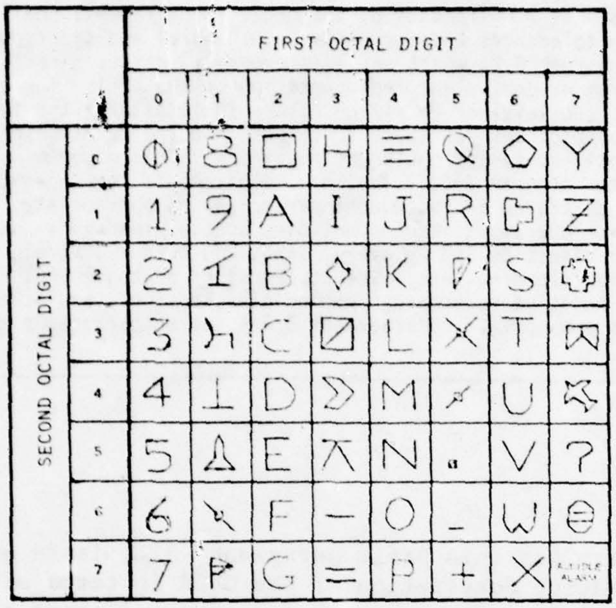

Figure 3.11. Display Character Set

b. Display Routing. Category values for the A and B elements of the standard display shall be derived utilizing Display Routing Category Assignments in accordance with Table XVI.

Table XVI. Display Routing Category Assignments.

Category Name	Category No. (Switch No.) (Decimal)	Category Value: Bits 39-42 of the Display Message (Octal)	Category Value: Bits 44-48 of the Display Message (Octal)
Spare	1	03	XX
Interceptors	2	04	XX
Hostile Class	3	YY	05
Friendly Class Tracks	4	YY	06
Tentative Track Reports	5	YY	07
Forwardtold Tracks	6	YY	01
Exercise Data	7	YY	02
Exercise Tracks	8	YY	03
[illegible] Assigned to SD	9	YY	04

Figure 8 (cont'd). Sample Display Interface Data.

3.6 PARAGRAPH 3.2, DETAILED FUNCTIONAL REQUIREMENTS

> 3.2 Detailed functional requirements. This paragraph shall specify, in subparagraphs defined below, the functional requirements of the CPCI. Requirements shall be stated in quantitative terms, with tolerances where applicable. General and descriptive material may be included in basic paragraph 3.2, which shall incorporate, either directly or by reference, a functional block diagram or equivalent representation of the CPCI. The graphic portrayal shall be accomplished to the level of detail necessary to illustrate the functional operation of the CPCI, the relationships between these functions, and the relationships between the functions and other identified system/equipment functions. This diagram is not intended to be restrictive on computer program detail design. Requirements for separately identified CPCI functions shall be described in subsequent paragraphs as appropriate. A subparagraph shall be included for each operational function, plus special functions such as sequencing control, displays, error detection and recovery, input and output control, real time diagnostics, operational data recording, etc. The descriptions of these CPCI functional requirements shall include the relative sequencing, periodicies, options, and other important relationships of each as appropriate. Paragraphs 3.2.x and subparagraphs shall be repeated for each function above.

3.6.1 General

The important objective for this basic paragraph, 3.2, is to provide an integrated and informative description of the CPCI in terms of its major operational functions. Its purpose is to describe the scope of functions to be performed by the CPCI as a whole, and to explain the sequencing and interrelationships among the major functions for which detailed input, processing, and output requirements are set forth in the ensuing structure of subparagraphs.

The major functions described here should typically correspond with, or be directly traceable to, functions described in the system specification which were allocated to this CPCI. Relationships with other systems/equipment to be included in the description, in accordance with the above instructions, should also correspond with interfaces identified in the system/system segment specification.

Note the instruction to incorporate a functional block diagram or equivalent representation of the CPCI functions and their interrelationships in a manner which "is not intended to be restrictive on computer program detail design". For a complex set of operational requirements, each major function will normally be expanded into its subfunctions and then into successively lower-level subfunctions as may be necessary to arrive at a comprehensive and definitive set of performance requirements for the CPCI. However, the resulting hierarchy of functions does not dictate a corresponding structural organization of the CPCI. Computer program designers, later, will normally create some computer program components (CPCs) or modules to perform control or support functions not explicitly specified, or to perform certain processing operations which may be common among several of the functions/subfunctions defined in the Part I specification. Thus, a description of the manner in which functions delineated in this paragraph are allocated to CPCs is a prominent part of the CPCI design to be examined at preliminary design review (PDR) and later incorporated into the CPCI's Part II specification (cf. instructions in paragraph 60.5.3.1 of MIL-STD-483 for this Part II-level description).

3.6.2 Example

Figure 9 illustrates a type of functional block diagram which has proved to be very useful both in the analysis leading to a Part I specification and to support the type of description called for in this paragraph. Some of the points of information depicted in the diagram are explained in the notes below.

- Major functions allocated to the mission CPCI are identified in the central blocks (stairstep arrangement) of the diagram. In addition, the diagram as a whole shows how those functions relate (a) to other software and hardware elements of the system, (b) to external systems, and (c) to each other.

- Solid lines with arrows are used in this diagram to indicate information flow from or to one or more CPCI functions and external elements. For example:

  --Incoming radar data are input to the Air Surveillance Function, via communications link.

  --Manual inputs may be entered into any major function via the Keyboard Entry Device.

  --Outputs to Interceptors, via communications, are generated by the Weapons Control Function.

  --Six of the major functions generate elements of displays for routing to the Display Device.

- Dashed lines with arrows indicate that, in this example, each major function generates outputs which are used by other functions, and vice versa.

- For purposes of this paragraph, the accompanying narrative should describe those major functions and their interrelationships, identifying the nature of external inputs and required outputs. Obviously, the relationships must be consistent with interface, input, and output identifications provided elsewhere, notably in paragraphs 3.1.1, 3.2.x.1, and 3.2.x.3.

3.2 Detailed Functional Requirements. The Mission Computer Program (MCP) will perform automated processing required at each Sector Control Center (SCC) of the 4XXL System to support the operational air defense functions defined for each SCC in the System Specification, 4XXL-3001A. Requirements are defined herein for the MCP to perform seven (7) major functions depicted in Figure 3.2.1 and described below. Of those, five (5) consist of the primary mission functions of: Air Surveillance, Target Identification, Weapons Direction, Weapons Control, and Sector Command. Two additional functions of Simulation and Recording are defined which will enable MCP to operate in the system exercising mode, in conjunction with the Exercise and Data Reduction CPCIs, without disrupting its processing of live environment data.

As installed at each CC, the MCP will be adapted to operate in the environment peculiar to that SCC. The adaptation will consist of: (a) fixed data pertaining to geographical positions, air

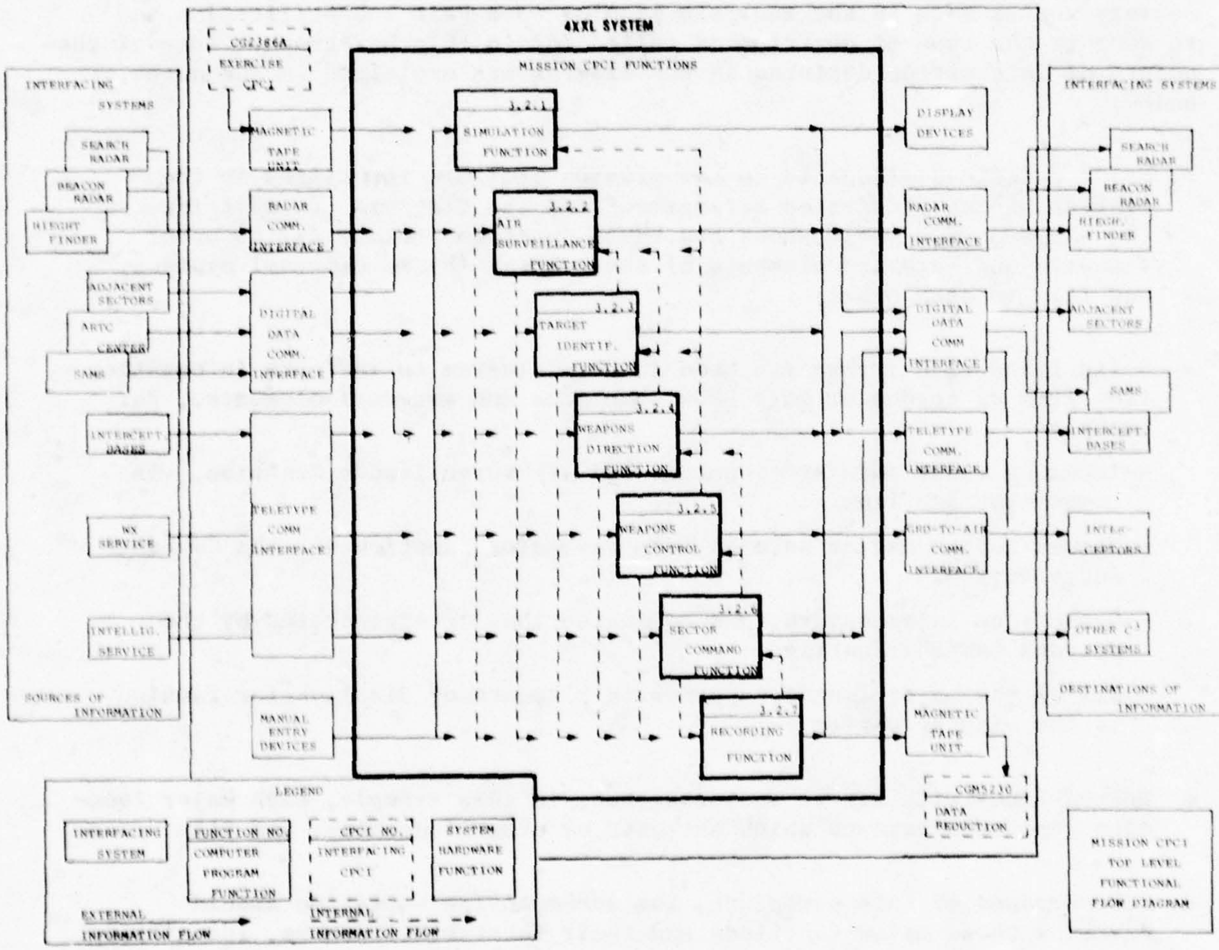

Figure 3.2.1 Top-Level Functional Flow Diagram for the MCP.

Figure 9. Sample Functional Flow Diagram for a CPCI.

3.7 PARAGRAPH 3.2.x, FUNCTION X

> 3.2.X Function X The basic paragraph for each function shall begin with descriptive and introductory material which defines the function and its relationship to other functions. Then, the following three subparagraphs shall specify the quantitative requirements concerning the function.

Like the preceding basic paragraph 3.2, this basic paragraph for each major function calls for an informative description of the function in terms of its component functions (subfunctions) and their processing interrelationships. Together with the preceding description of the CPCI as a whole, the purpose of this description is to clarify relationships which account for, but may be less readily apparent in, much of the subsequent detail specified for lower-level functions.

Figure 10 illustrates how a functional block diagram similar to, and related to, that shown previously in Figure 9 can also be used to support the description at this next level.

It may be noted that paragraph numbers shown in Figure 10 (at the top of each function block) do not correspond with the breakdown into inputs/processing/outputs paragraphs (paragraphs 3.2.x.1/2/3) described in the general instructions. In the specification from which this example is drawn, that breakdown occurred at the next-lower level, resulting in those paragraphs having numbers of the form, 3.2.x.y.1/2/3. For example, inputs and outputs for the Radar Processing function are specified in paragraphs 3.2.2.1.1 and 3.2.2.1.3, respectively; and functions at still lower levels are identified in further subparagraphs and sub-subparagraphs under the processing paragraph, 3.2.2.1.2. Those aspects of the specification structure should be determined by the functional complexity of the given CPCI.

3.2.2 Air Surveillance Function. The Air Surveillance function encompasses the four major functions of Radar Input Processing, Target Tracking, Track Telling, and the Surveillance Display. Figure 3.2.2.1 depicts these functions and the general nature of their processing interactions with other functions both internal and external to Air Surveillance. Detailed requirements for inputs, processing, and outputs are specified in a major subparagraph herein corresponding to each of those functions, as identified in Figure 3.2.2.1.

Requirements specified for the Radar Inputs function include requirements for coordinate conversions, editing of radar data for legality of formats and ranges of values, correlation of radar data from multiple sensors, and the generation of target position data for use in

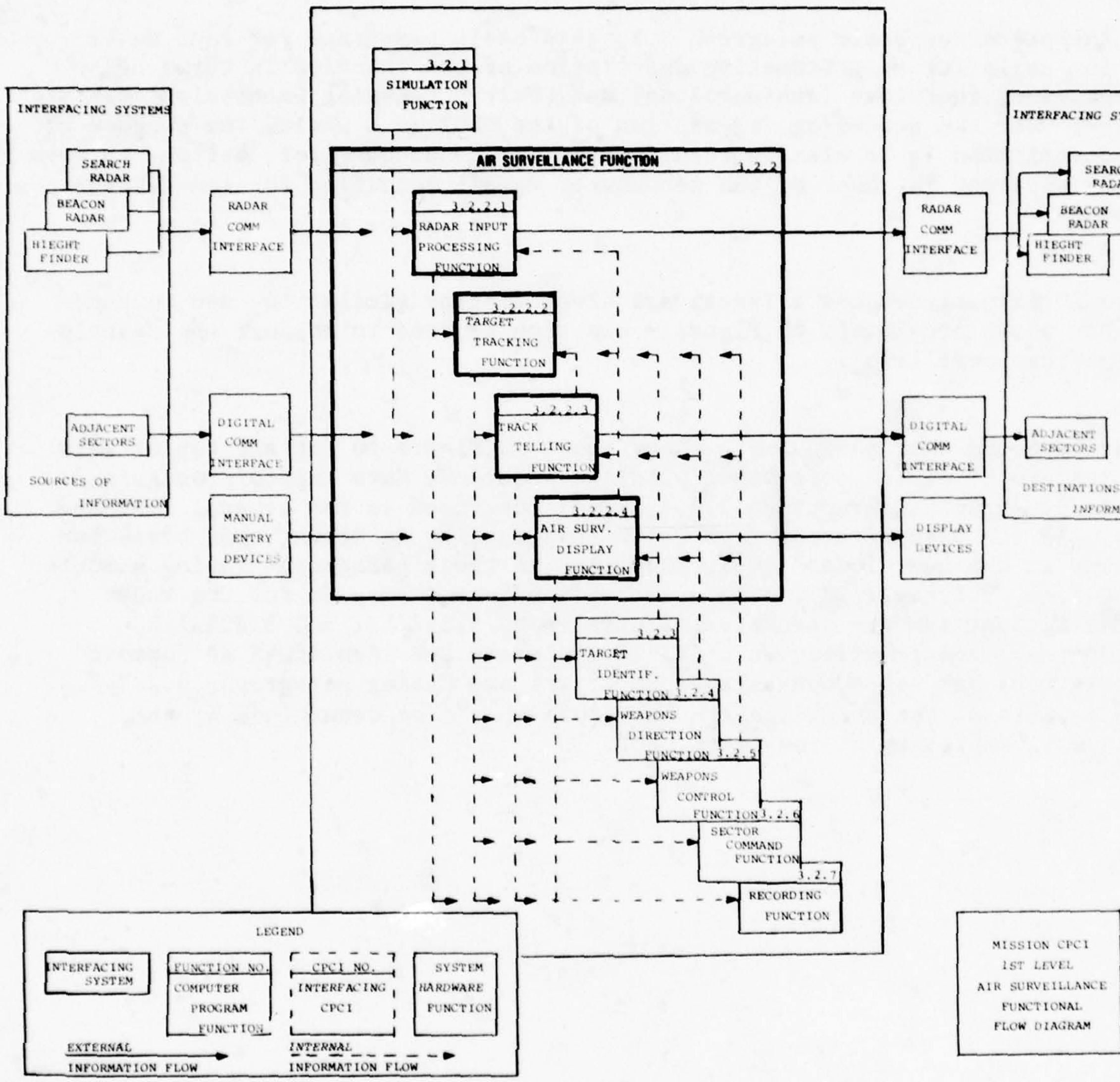

Figure 3.2.2.1. Air Surveillance Function; First-Level Functional Flow.

Figure 10. Sample Expansion of a CPCI Major Function.

3.8 PARAGRAPH 3.2.x.1, INPUTS

> 3.2.x.1 Inputs. This paragraph shall specify either directly or by reference to another part of this specification the source(s) and type(s) of input information associated with a function of the CPCI. This shall include a description of the information, its source(s) and, in quantitative terms, units of measure, limits and/or ranges of units of measures, accuracy/precision requirements, and frequency of input information arrival.

> (Recommended Entry for CDRL Backup Instructions)
>
> 3.2.x.1 Inputs. Add the following two sentences: "Inputs received from other functions, which are wholly internal to this CPCI, shall be specified at a level of detail which is appropriate to clarify those functional relationships. Such inputs are not subject to formal qualification."

3.8.1 General

The emphasis in this paragraph is on providing complete information about inputs to the function, including all relevant detail which will be needed as a basis for CPCI design and coding. The requirement is to identify each input, and to provide for each: identification of its source; a name; a precise definition of the item; and all applicable information pertaining to units of measure, range of possible values, frequency/timing, and accuracy. For non-quantitative inputs, the definition must contain a listing of the possible states of the variable.

Definitions of inputs may be made in prose and tabular form as best fits the nature and quantity of input data for the given function. Information should be grouped into subparagraphs on a meaningful basis. Paragraph/subparagraph numbers and letters, together with input names, should be organized in such a way that each individual input can be identified without ambiguity.

The process of gathering and organizing detailed information about inputs is an essential part of analyses performed prior to and during preparation of the CPCI development specification. This paragraph is the place where that information is recorded and controlled for the life of the CPCI. The information should be specified in meticulous detail, taking into account certain qualifications and points of emphasis noted below.

3.8.2 Notes

a. Role of Inputs as Requirements. As one part of the specification, the role of information specified in the inputs paragraphs for CPCI functions tends to be similar to that of interface information, in that: the developer is constrained to design the CPCI to utilize and be compatible with the inputs as defined, but he may have limited or no responsibility (depending on the source) for ensuring that the inputs will prove to have those specified characteristics when the system is put into operation. This consideration affects (1) the manner in which "requirements" contained in the inputs paragraph are phrased and (2) the subsequent handling of those requirements during the test program:

- The words "shall" and "will" are both used, appropriately to the actual intent; note the phrasing of sample statements illustrated in Figure 11 below. Usually, the appropriate orientation is that: inputs having the specified characteristics <u>will</u> *be provided* (*to* the given function developer); and the developer <u>shall</u> design the CPCI to utilize those inputs as specified.

- All specified inputs should be verified during the qualification process when feasible (see Note <u>c</u> below). However, failures of certain inputs to have their specified characteristics do not necessarily imply failure of the given CPCI to fully qualify.

b. <u>Specification by Internal Reference</u>. The use of specification by reference requires the use of coordinated rules affecting other parts of the specification, considering that each input will be identified elsewhere as also being either: an interface; an element of the data base; or the output of another function. One useful device is to organize major portions of the data into separate appendices, providing (as examples):

- One common appendix containing detailed definitions of all messages--i.e., messages either received by the given CPCI from an external source or transmitted from the CPCI to an external destination (external to the CPCI, not necessarily to the system). Definitions contained in the common appendix are then referenced under appropriate subparagraphs of the interface paragraph (3.1.1.2) and of each affected paragraph concerned with function inputs (3.2.x.1) and outputs (3.2.x.3).

- Another common appendix containing detailed definitions of data base items, which is referenced in its entirety under the data base paragraph (3.3) and appropriately for individual items in the input paragraphs of functions utilizing the data base as a source.

c. <u>Inputs From Other Internal Functions</u>. Special problems have been encountered with those inputs and outputs which are peculiar to relations among functions, as distinguished from inputs/outputs which are either external or associated with a fixed data base. For a variety of reasons, it is intentionally not required that the CPCI be structured into components (CPCs) which correspond one-to-one with functions specified at the Part I level. Thus, during the subsequent process of computer program preliminary design, the designers are free to allocate the functions defined in the Part I specification to some different number of CPCs. As a result, many inter-CPC relationships will exist which do not correspond with inter-function relationships defined in the Part I; and, some of the specified input/output relations peculiar to functions may be handled internally to individual CPCs.

However, those implications are not explicitly recognized in the MIL-STD-483 instructions for this paragraph of the specification, as written. Hence, a set of CDRL backup instructions similar to that illustrated above is generally applicable. Recommended rules for handling that class of inputs/outputs in preparing the specification are:

- Define the characteristics of the item(s) in the output paragraph (3.2.x.3) of the producing function, in sufficient detail to clarify the functional relationship and intent.

- In the input paragraph (3.2.x.1) of the receiving function, identify the producing function as the source and specify the item by reference in that paragraph (3.2.x.3).

- Where the possibility exists that specified input/output relations between internal functions may be altered or lost in the process of computer program design, ensure that the references to those inputs and outputs are properly handled in preparing the Verification Cross Reference Index for Section 4, Quality Assurance Provisions. Generally, the entry "N/A" is appropriate for such inputs and outputs.

3.8.3 Examples

Illustrative data shown in Figure 11 are adapted from a specification which contained complete definitions of inputs directly in the input paragraph. The examples illustrate (a) how all inputs to the function may be identified, and (b) the levels at which complete definitions are provided for a few of the identified inputs.

As noted above, it is often preferable to make use of references to other parts of the specification. Efficient organization of the various elements of data comprising inputs, outputs, data base, and message interfaces is particularly important when the total volume of data is large. From the point of view of this paragraph (3.2.x.1), the detailed definitions of all inputs can be specified by reference. However, referencing applies only to detailed definitions. Information to be provided directly in this paragraph should normally consist, at a minimum, of a listing (or table) of all inputs to the given function, including for each input:

- Name of the input.

- Source of the input.

- Reference to the specific location of its detailed definition.

3.2.3.1 *Source and Type of Inputs*. To perform the Data Transfer function, the MDICP shall be designed to accept and process inputs that are listed in Table IV and defined in subsequent paragraphs.

Table IV. Summary of Input Sources and Types.

TYPE OF INPUT	SOURCE	UNITS	LIMITS	ACCURACY/ PRECISION	FREQUENCY
Zulu Time	Manual Inputs	Hours 0-23 Minutes 0-59 Seconds 0-59	24 Hours	1 Second	During Startup
TADIL R TO Messages	TACC	Refer to 3.2.3.1.2			Maximum of 29 messages per second
Equipment Assignments:	Manual Inputs				
I/O channel assignments		I/O Channels	0-3, None	N/A	During Startup
Display channel assignments		Display Channels	5-9, None	N/A	During Startup
Teletype assignments		Teletype Numbers	0-4, None	N/A	During Startup
Simulation Test	Simulati...	...3.2.3.1.1 through			Ref...

3.2.3.1.2 TADIL R TO Messages. The MDICP shall be capable of accepting and processing TO messages input from TACC at rates up to 29 messages per second. Each message consists of 3 words having the core format and content illustrated in Figure 6 below. Explanations, units, accuracy, precision, and/or logical values of the message data content are provided in Table V.

WORD	BIT POSITION																							
	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	LABEL				TRACK NUMBER												ACTION				IDENTITY		S	P
2	TRACK X COORDINATE												VALIDITY CODE (1)								TRACK QUALITY			p
3	TRACK Y COORDINATE												VALIDITY CODE (2)								I	E	C	P

Figure 6. TADIL R TO Message Format and Content.

Table V. TADIL R TO Data Explanations and Characteristics.

DATA TITLE	EXPLANATION	UNITS, LIMITS, ACCURACY, PRECISION, OR LOGICAL VALUES
LABEL	A number assigned to each TADIL R message to identify the type of message.	1000_2 (equal to 10_8)
TRACK NUMBER	The reference number used to associate information and directions with a particular track.	12 bits (equal to 1000_8)
ACTION	Indicates either (a) the type of action to be carried out by the recipient or (b) action taken by the originator.	0 - Drop track 1 - Change track number 2 - Forward-told to MTDS 3-7 - Reserved
TRACK COORDINATES	The position of the track in X and Y coordinates relative to the common coordinate system center and oriented to grid North from the coordinate center. X and Y coordinates are positive in the East and North directions, respectively.	LIMITS: ±511 3/4 data miles for each coordinate; negative values are one's complement

Figure 11. Sample Identifications and Definitions of Inputs.

3.9 PARAGRAPH 3.2.x.2, PROCESSING

> 3.2.X.2 Processing. This paragraph shall provide a textual and mathematical description of each of the processing requirements of each function. Presentation of the mathematical descriptions under each function shall include:
>
> a. Purpose - This area shall describe the exact intent of the mathematical operation(s). This involves a definition of the specific input and output parameters and the processing required.
>
> b. Approach - This area shall contain a textual description of each mathematical operation specified. The accompanying narrative shall identify accuracies required, sequence and timing of events, and relevant restrictions or limitations. Derived equations shall be shown with appropriate mathematical and control symbols adequately defined.

3.9.1 General

The purpose of information in this paragraph is to define all of the rules by which inputs to the given function (specified in 3.2.x.1) are transformed into its required outputs (specified in 3.2.x.3). Although not amplified in the above instructions, all processing operations must be specified, whether or not they are normally regarded as mathematical computations. For example, they may include requirements to:

- Edit inputs for legality, and to generate specified output messages when inputs are rejected (see 3.12.2).
- Perform routine retrieval, sorting, summarizing, correlating, and/or routing of data.
- Perform complex modeling, evaluation, prediction, or any other data transformations required to produce the necessary outputs.

3.9.2 Notes

a. Relations to Input/Output Paragraphs. In the course of defining processing operations, this paragraph necessarily identifies the individual inputs and outputs, or groups of those, which are specified in the preceding and following paragraphs, respectively. However, the specific function of those other paragraphs is to provide separately-organized listings of all inputs and outputs associated with the given function, together with the detailed definition of each item. Thus, this paragraph need not define those same detailed characteristics of each item, redundantly, in the course of specifying the processing operations.

b. Equations vs. Algorithms. Many processing operations are properly specified by means of mathematical formulas or transformation equations, together with definitions of their symbols and terms. The term "algorithm" normally refers to a particular procedure and sequence of operations through which a given set of computations is performed. An equation, for example, might be solved by using any one of several alternative algorithms. Considered in that light, the use of algorithms to specify processing operations in this paragraph should generally be avoided. They specify design, in effect, and have proved frequently to be in conflict with required CPCI performance. It is particularly important that the content of this paragraph emphasize what processing is to be accomplished, in accordance with specified performance criteria (e.g., of speed, timing, volumes, priorities, end-result accuracies), as opposed to how the data manipulations are to be carried out.

c. Derivations. Note that the "derived equations" are to be specified, not their derivations as such. As a general rule, derivations (or other documentation of system engineering analyses leading to the Part I specification) are not proper content for the body of this specification. See 3.25 herein for a further discussion of derivations and alternate methods.

3.9.3 Examples

Processing operations may be specified in the form of narratives, equations, tables, graphic aids, or combinations of those as appropriate to the type of function being described. Figures 12 and 13 illustrate samples for only two of the many different forms of presentation and varieties of functions to be encountered.

Figure 12 is a sample of requirements specified for a portion of the calculations involved in scoring air intercept exercises, in a flight simulation facility.

Figure 13 is a (modified) sample of switch action processing requirements for an air defense operational computer program. The two tables of switch action lists shown at the bottom of this figure are drawn from similar tables which were organized, in the actual specification, into a separate, 200-page volume.

3.2.9.2.3.4.2 Elevation check. The target shall be considered within the vertical gimbal limits if the elevation of the target with respect to the manned Interceptor (α_e) is less than, or equal to, the Interceptor's radar gimbal limits in elevation (α_{eg}). The angle $\alpha_{eg} = \alpha_{eu}$ if the target is above the Interceptor and $\alpha_{eg} = \alpha_{ed}$ if the target is below the Interceptor (see Figure 2a).

The angle α_e shall be calculated as follows:

$$\alpha_e = \sin^{-1} \frac{|Z_I - Z_T|}{D_{IT}}$$

where:

Z_I = Altitude of the Interceptor

Z_T = Altitude of the Target

The absolute value of $Z_I - Z_T$ is used since the Target can be on either side of the horizontal plane of the Interceptor (the center line of the radar volume).

The determination of the value used for α_{eg} shall be as follows:

a. If the Interceptor is above the Target (i.e., $Z_I - Z_T$ is positive), Interceptor's lower gimbal limit (α_{ed}) will be used.

b. If the Interceptor is below the target (i.e., $Z_I - Z_T$ is negative), the Interceptor's upper gimbal limit (α_{eu}) will be used.

If the Target passes the range and elevation checks, the Target"s position shall be translated (to the X_1, Y_1 axes) and rotated (to the X_2, Y_2 axes) so that the Interceptor's position and heading are the reference instead of the AN/GYK-19 grid center and true north. The determination of the Target's position relative to the manned Interceptor shall be computed as follows:

$$x_2 = x_1 \cos \psi_1 - y_1 \sin \psi_1$$
$$y_2 = x_1 \sin \psi_1 + y_1 \cos \psi_1$$

where:

ψ_1 = Interceptor heading (true) measured in the X,Y axis system

x_1, y_1 = The Interceptor to Target coordinates computed in 3.2.9.2.3.4.1

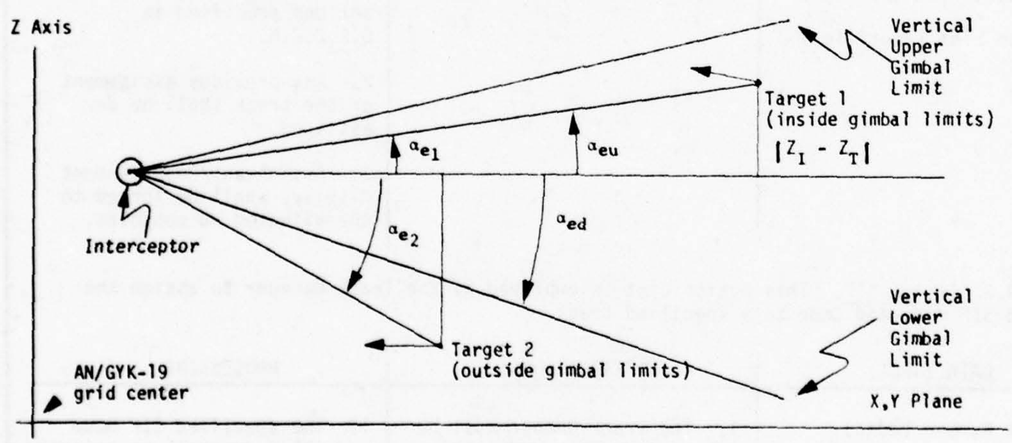

Figure 2a. Elevation Check.

3.2.9.2.3.4.3 Front plane check. The target shall be considered in front of the manned Interceptor if the y_2 position is greater than zero, as illustrated by Target 3 in Figure 3 (Target 4 does not

Figure 12. Sample of Processing Requirements for an Umpiring Function.

3.2.1.2.4 Switch Action Data Processing. The Mission Computer Program (MCP) shall process Switch Action inputs from the Track Management Console in accordance with the criteria defined for each Action List in the following subparagraphs. For purposes of this specification, a Switch Action is defined as a single data input item, such as Track Number, Power Level, etc.; a grouping of Switch Actions which defines a processing requirement for the MCP shall be termed an Action List. Prior to acceptance of Switch Action inputs for further processing, MCP shall check all such inputs for compliance with (a) alphanumeric format/content specified for individual Switch Action inputs in 3.2.1.1 and (b) restrictions applicable to each Action List for which processing is specified in the subparagraphs herein.

a. Console Input Feedback Display. The alphanumeric characters corresponding to each Switch Action shall be displayed on this display in the format specified in 3.2.1.3.27. With the exception of Track Number, all individual Switch Actions shall be displayed in the order of their insertion. Feedback delay for each Switch Action shall not exceed 0.2 seconds.

   (1) Each Action List, whether legal or illegal for the given Unique Track Action Identifier, shall be displayed until a new Action List is initiated.

   (2) When the inserted Action List is legal in its entirety, and MCP initiates the ordered processing on a designated Track, the Track Number shall be output to complete the specified content of this display.

b. ILEG and Alarm 3. The insertion of any illegal element in a track Action List, or the failure to insert a required element, shall (1) cause the action to be rejected for further processing of the ordered action, (2) activate the Illegal Switch Action (ILEG) display, and (3) activate the (audible) Alarm 3. The ILEG display shall continue until a new Action List is initiated. MCP shall clear Alarm 3 after 3(±1) seconds.

3.2.1.2.4.1 Assign/Reassign Tracks. This Action List is employed by the Track Manager to assign control of a specified track to a specified Weapons Director (WD).

DATA INPUT	RESTRICTIONS	PROCESSING
1. Track Number (dddd) 2. WD Designator (WD1-WD4) 3. Action List Identifier	1. The Track Number must be one assigned to an existing track.	1. The specified track shall be assigned to the specified WD for processing of all WD actions specified in 3.2.2.2.6. 2. Any previous assignment of the track shall be de-assigned. 3. Assignment/Reassignment displays shall be forced to the affected WD consoles.

3.2.1.2.4.2 Assign SIF. This Action List is employed by the Track Manager to assign the indicated SIF Mode and Code to a specified track.

DATA INPUT	RESTRICTIONS	PROCESSING
1. Track Number (dddd) 2. SIF Code Mode (1,2,or 3) 3. SIF Code (0000 - 9999) 4. Action List Identifier	1. The Track Number must be one assigned to an existing track. 2. The track must not be a told-in track. 3. The track must not be in process of being dropped.	1. The specified SIF Mode and SIF Code shall be assigned to the indicated track.

Figure 13. Sample Specification of Switch Action Processing.

3.10 PARAGRAPH 3.2.x.3, OUTPUTS

> 3.2.X.3 Outputs. This paragraph shall specify, either directly or by reference to another part of this specification, the destination(s) and type(s) of output information associated with a function of the CPCI as a result of the processing described in paragraph 3.2.2. This shall include a description of the information; its destination(s); and, in quantitative terms, units of measure, accuracy/precision requirements, frequency of output information, etc, where applicable.

3.10.1 General

As viewed by the intended CPCI user, outputs can constitute the most visible and important portions of a Part I specification. Once the CPCI is developed, outputs and their characteristics tend to be the focal criteria for evaluating its performance, both during formal test and later operations. Hence, the analysis and documentation of precise requirements for outputs should be matters of initial and continuing emphasis throughout the course of developing the Part I specification.

Outputs for the given function may be specified directly in this paragraph or by reference to another part of the specification, as indicated in the instructions. Unlike inputs or interfaces with existing external items, outputs should rarely if ever be specified by reference to other applicable documents. The use of internal references for the detailed output definitions, however, may often be justified by considerations of efficient data organization and specification maintenance, as for inputs (see 3.8 above). Again, the minimum information to be provided directly in this paragraph for each function should consist of a listing of the function outputs, identifying for each:

- Name of the output.
- Destination of the output.
- Reference to the specific location of its detailed definition.

Detailed and precise definitions of output characteristics are generally mandatory and indispensable for outputs external to the CPCI. There is normally less need to provide the same degree of precise detail in defining outputs to other functions internal to the CPCI, for reasons discussed earlier in 3.8.2. However, those internal outputs should also be identified here, in the minimum manner described above, since they constitute essential links in tracing the flow of information across functions. For that purpose, such an output can often be defined adequately by reference to the portion of the processing paragraph (3.2.x.2) which describes how it is generated.

3.10.2 Examples

Figure 14 provides an example of the minimum listing of outputs which should appear directly in this paragraph for each function. In this particular example, it happens that all of the detailed definitions are specified by reference, rather than in the listing itself; some of the references are to an appendix (for message formats) and some to subparagraphs of this paragraph.

Samples of the detailed definitions for two of the outputs, marked as "A" and "B" in the figure, are reproduced in Figures 15 and 16, respectively.

3.2.4.3 Outputs. In performing the Track Data Transfer Function, the Interface Computer Program (ICP) shall produce outputs listed in Table XIII and defined in paragraphs referenced therein.

Table XIII. Destinations and Types of Outputs.

TYPE OF OUTPUT	DESTINATION	UNITS	LIMITS	ACCURACY	FREQUENCY
TADIL E messages M2 M3	NATS	Refer to 40.3.4.3.1 A Refer to 40.3.4.3.2			Max. 33 messages per second
TADIL E messages M7 M9	TACC	Refer to 40.3.4.3.3 Refer to 40.3.4.3.4			Every 20(±4) seconds
92-bit MO messages; Position Velocity	NATS	Refer to 40.3.6.3.1 Refer to 40.3.6.3.2			Max. 18 messages per second
IO recording request	Simulation and Recording	Real time, see 3.2.4.3.1 Non-real time, see 3.2.4.3.2			Max. every 2 seconds
Track situation display	Display console	Refer to 3.2.4.3.3 B			Refer to 3.2.4.2.11
Attention situation display	Display console	Refer to 3.2.4.3.4			Refer to 3.2.4.2.11
Console alarms	Display console	Refer to 3.2.4.3.5			Refer to 3.2.4.2.2

Figure 14. Sample Table of Outputs for a Function.

40.3.4.3.1 TADIL E Messages, M2. The ICP shall output M2 TADIL E messages to NATS in the format specified in Figure 19 below. Content of the messages shall comply with definitions, limits, precision, and logical values listed in Table XXIV.

<table>
<tr><th rowspan="2">WORD</th><th colspan="24">BIT POSITION</th></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>1</td><td colspan="12">TRACK NUMBER</td><td colspan="4">LABEL</td><td colspan="8">HEIGHT</td></tr>
<tr><td>2</td><td colspan="8">M VELOCITY - $\dot{X}$ L</td><td colspan="8">M VELOCITY - $\dot{Y}$ L</td><td colspan="3">TRACK QUALITY</td><td colspan="5">SPARES</td></tr>
<tr><td>3</td><td colspan="5">TIME TAG (HOURS)</td><td colspan="7">TIME TAG (MINUTES)</td><td colspan="2">E</td><td colspan="2">RAID SIZE</td><td colspan="8">AMPLIFYING IDENTITY</td></tr>
</table>

M = Most Significant Bit; L = Least Significant Bit.

Figure 19. TADIL E M2 Message Format and Content.

Table XXIV. TADIL E M2 Data Explanations and Characteristics.

DATA TITLE	EXPLANATION	UNITS, LIMITS, ACCURACY, PRECISION, OR LOGICAL VALUES
TRACK NUMBER	The reference number used to associate information and directions with a particular track.	Four decimal digits, 0000 - 4056
LABEL	A number assigned to each TADIL E message to identify the type of message.	1000_2; 0010 = M2 0011 = M3 0111 = M7 1001 = M9
HEIGHT	The height of the track is the contact altitude above mean sea level.	LIMITS: 000 - 127,500 feet UNITS: 500 feet If height is unknown, 11111111_2 is transmitted
VELOCITY	Velocity vector of the track in X and Y components. The most significant digit of each component indicates direction.	LIMITS: ±127/128 data miles/sec UNITS: 1/128 data miles/sec Negative values are reported as one's complement
TRACK QUALITY	The value computed by the reporting source based on the frequency of its data.	0 - 7 (7 = highest quality)
TIME TAG	Time that track position was observed is expressed in hours and minutes, Zulu time.	LIMITS: 0 - 23 hours 0 - 59 minutes UNITS: 1 minute

Figure 15. Sample Definition of a Message Output.

3.2.4.3.3 Track Situation Display. The Track Situation display shall be presented on-screen and shall appear in the central 12" by 12" area of the console display scope. Its position within the 512 by 512 point matrix for the on-screen area shall be determined by the location of the track Identity Symbol representing relative geographic location of the track. The display shall consist of a Track Number, Identity Symbol, Track Age, and Track History. See Figure 26.

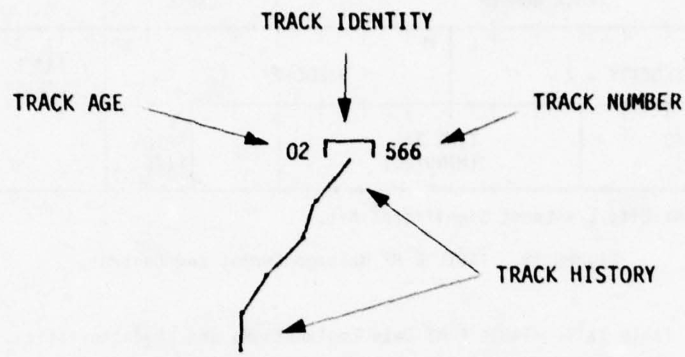

Figure 26. Relative Geographic Location of Tracks.

The displayed elements shall comply with the formats and definitions provided in Table XIV.

Table XIV. Track Display.

PARAMETER	CODE	LEGAL VALUES/EXPLANATION
Track Number	ddd	An octal number representing the three least significant of the MTDS track number.
Track Identity	⌒	Friendly identity
	⊓	Unknown identity
	∧	Hostile identity
	[	Special identity (not expected from MTDS)
Track Age	dd	Time elapsed, in minutes, since the track was at this position. This value represents the difference between current Zulu time, and the time of observation reported for the track in the most recent message received from the MTDS on the track.
Track History	—	A vector line. Up to nine vectors can be displayed, connecting the ten last-reported positions of the track. The number of vectors displayed can be reduced by the Display N Vectors switch action.

3.11 PARAGRAPH 3.2.n, SPECIAL REQUIREMENTS

3.2.n* Special requirements. This paragraph shall specify, in appropriate subparagraphs, requirements which affect the design of the CPCI and are distinguishable from the performance requirements of paragraph 3. These requirements result from general considerations of CPCI usability. These may include, but are not limited to, requirements for:

a. The use of programming standards to assure compatibility among computer program components (CPCs - subprogram or groups of subprograms).

b. Program organization, such as overall program segmentation. In addition, for CPCIs which contain or process classified information, special attention shall be given to the requirements for protecting classified information.

c. Program design resulting from consideration of modifications to the CPCI during operation (e.g., on-site modification requirements and the permissible amount of operational degradation allowed during installation of modification may be specified).

d. Special features, to facilitate the testing of the CPCI. For example, special procedures for the design of CPC interfaces, requirements for intermediate printouts, and commentary on the program listing may be required.

e. Expandability (growth potential) to facilitate modifications and additions to the CPCI.

n* = The next sequential number following the number of the last function.

(Recommended Entry for CDRL Backup Instructions)

3.2.n Special requirements. Delete instructions for this basic paragraph in MIL-STD-483 and replace by the following:

Design requirements for the CPCI shall be specified in this paragraph (e.g.: use of computer programming language or other standards; requirements pertaining to CPC organization or control; limitations in utilization of computer storage). Design requirements specified herein shall be the minimum necessary to meet verified needs of the procuring activity. For CPCIs which support a system, this paragraph shall cite paragraph 3.3.8 of the system specification and incorporate requirements peculiar to this CPCI on an add or delete basis. Design requirements shall be specified to the maximum extent possible by reference to established military specifications and standards.

Changes are suggested to this paragraph largely because some of the examples given do not clearly support the required distinction between design and performance, e.g., protecting classified information, allowable degradation, intermediate printouts. "Commentary on program listing" is a matter to be handled via CDRL backup instructions to the Part II specification, not here.

The principal function of this paragraph is to verify or modify the applicability to this CPCI of design requirements/standards specified for computer programs in paragraph 3.3.8 of the system specification. Unless there are good reasons in a given case to the contrary, this paragraph should normally specify the same requirements/standards, either directly or by reference to the system specification. In both cases, effort should be made to minimize the specified design requirements and constraints, in the interests of reducing (a) costs and (b) the ever-present potential for conflict with specified performance.

3.12 PARAGRAPH 3.2.n.1, HUMAN PERFORMANCE

> 3.2.n.1 Human performance. Human performance/human engineering requirements for the CPCI shall be specified in this paragraph (e.g., minimum times for human decision making, maximum time for program responses, maximum display densities of information, clarity requirements for displays, etc.). For CPCIs which directly support a system(s), this paragraph shall cite the appropriate paragraph(s) of the system/system segment specification which establish the human performance/human engineering requirements for all system equipment, and incorporate requirements peculiar to this CPCI on an add and/or delete basis.

> (Recommended Entry for CDRL Backup Instructions)
>
> 3.2.n.1 Human performance. Delete instructions for this paragraph in MIL-STD-483 and replace by the following: .
>
> For CPCIs which directly support a system, this paragraph shall cite the appropriate paragraphs of the system specification which establish the human performance/human engineering requirements for system segments and equipment, and identify the degree to which requirements peculiar to this CPCI are incorporated in the content of this specification. In addition, this paragraph shall provide an organized list of the human engineering criteria employed in determining the specific requirements documented in other paragraphs of this specification for displays, manual inputs, and special processing to support operator tasks (e.g., the operator shall be able to clear all display alarms; all control inputs shall have a positive indication that the input has been accepted; all geographic displays shall be north-oriented, etc.). Compliance of the CPCI as a whole with these criteria shall be subject to human factors test and evaluation during CPCI qualification.

3.12.1 General.

This paragraph has proved to be a frequent source of confusion and difficulty. Reasons for the CDRL backup instructions shown above are outlined briefly as follows:

- The general intent of requirements in the area of human performance is to assure that the "machine" portions of a man/machine system are designed to be compatible with natural human capabilities and limitations. The MIL-STD-483 instructions for this paragraph are based on equipment practice, in which human engineering aspects of design are largely incorporated, by the engineering designers, in response to requirements of the equipment CI development (Part I) specification. Requirements are often stated by reference to human engineering standards (MIL-STD-1472) covering such considerations as sizes and shapes of displays, locations of controls, workspace illumination, etc.

- In a data processing system, the human operator's direct interaction is with operator consoles, or other input/output devices, whose man/machine compatibility is a joint function of (a) the equipment design and (b) functional characteristics determined largely by the computer programs. Thus, the human engineering aspects are matters of computer program/equipment interfaces, as well as of the separate characteristics of those two "machine" elements of the system. From the point of view of the computer program, the characteristics which can facilitate or hamper efficient operator performance are totally functional--i.e., display formats and content, timing, responses to manual inputs, and special processing capabilities to support operator tasks.

- All of those characteristics, however, including detailed interfaces with the equipment displays and manual input devices, are characteristics to be analyzed, verified, and defined for the computer program directly in its Part I specification. They represent portions of the Part I specification which are particularly significant to its evaluation, approval, and eventual control by operational personnel of the using command. If the system engineering tasks of developing the Part I specification are properly conducted, they will be incorporated throughout the input, processing, and output requirements (paragraphs 3.2.x.1 through 3.2.x.3) specified for each affected function and subfunction.

Thus, useful functions of the information called for in the recommended backup instructions are to provide (a) a summary record of human engineering criteria employed to derive requirements contained in the Part I specification itself, and (b) a direct basis for human factors portions of the CPCI test program. The content of this paragraph may also help computer program designers to understand the reasons for, and intent of, various requirements documented elsewhere in the specification. However, it should not normally impose needs for any additional computer programming or human engineering design work during the CPCI development.

Statements to be listed in this paragraph should consist of criteria which apply generally across the CPCI's various functions. Hence, they will not cover all human engineering requirements for the CPCI, since many of those should have been derived through considerations specific to individual functions. Guidelines to be observed include the following:

- The statements should reflect, in fact, systematic attention to human performance aspects of operations that are required in the given system.

- They should be concise and directed towards functions performed by the CPCI, as distinct from equipment or the human operator--although clearly related to those when indicated. They should not include, for example, statements such as "The functions and locations of controls shall ensure ease, speed, and accuracy of operation...", which are ambiguous with respect to equipment vs. the CPCI.

- General human engineering principles should be included if they can be phrased in terms that are (a) clearly related to the given CPCI operation and (b) verifiable during CPCI qualification. Non-testable, "motherhood" statements should be avoided.

3.12.2 Example

Figure 17 lists a few statements to illustrate the types and levels of content appropriate for this paragraph.

3.2.5.1 Human Performance. The Mission Computer Program (MCP) shall be designed to comply with all detailed requirements specified in 3.2.1 through 3.2.4 herein for manual inputs, display, and related special processing capabilities required for efficient and error-free performance by the system console operators. Verification of the MCP design and performance for compliance with those requirements shall include verification of compliance with the general criteria listed below. These criteria, together with specific requirements incorporated in the referenced paragraphs for individual MCP functions, implement human performance/human engineering requirements allocated to the Mission Data Processing System Segment in paragraph 3.7.4.6.1 of the system specification.

a. All manual entries from the operator console shall be error-checked and verified for legality before being accepted by the functioning computer program. Positive feedback shall be provided for both accepted and rejected manual entries, in a manner appropriate to criticality of the event to system operations. In providing that feedback, use shall be made of equipment capabilities provided in operator console design requirements for: digital, situation, and input displays, including input display clearance; computer-controlled keyboard interlock; hardcopy output; and console light indicators.

b. The lighting of Alarm 1 or Alarm 2 at the operator console shall be used only to inform the operator of a situation requiring immediate action. Displays and/or printouts shall be provided in all cases when necessary to (1) identify the reason for the alarm and (2) facilitate corrective action.

c. Abbreviations used in digital displays, including digital elements of the situation display, shall comply with standards set forth in TAC Manual 302-15. Non-standard abbreviations shall be used only when (1) display space is critical and (2) their meaning is obvious to the operator.

⋮

n. Each symbol used to provide aircraft identification/status information in the situation display shall have only one meaning, regardless of track identity, console operator position, or mode of system operation.

Figure 17. Sample Content for the Human Performance Paragraph.

3.13 PARAGRAPH 3.2.n.2, GOVERNMENT-FURNISHED PROPERTY LIST

> 3.2.n.2 Government-furnished property list. This paragraph shall list the Government-furnished computer programs which the CPCI must be designed to incorporate. The list shall identify the program by nomenclature; specification number; model number, if appropriate; and associated documentation.

If there is a requirement for the contractor to physically incorporate existing, Government-furnished computer programs into the given CPCI, this paragraph is used to identify those items. The above instructions are based directly on established practice in specifying equipment CIs (see paragraph 20.3.1.4 of MIL-STD-490), in which the requirement to incorporate Government-furnished components tends to be frequent. However, it does not readily apply to CPCIs with the same meaning and implications. While it is difficult to rule out the possibility that it might prove useful in certain special cases, there are also some alternatives and potential problems which should be explored and resolved. As examples:

- Existing computer programs which can operate in the same computer, and are compatible with the same support software, are usually better identified as interfacing items and so specified in accordance with requirements for paragraph 3.1.1. This approach should be considered even if the Government-furnished item requires modification, and by the same contractor.

- There is a multiplicity of questions that can be raised regarding how, or whether, the documentation of existing items can be reconciled with the new item's specification, at both development and product levels. In the (unlikely) event that the existing item is properly specified at both levels fully in accordance with MIL-STD-483, much of that documentation might also be incorporated into the new CPCI specification (and the procuring activity presumably regards the CPCI as already being qualified in those respects). In the event that performance requirements are explicitly specified for the first time in the new CPCI's development specification, responsibilities for the CPCI qualification can easily become matters of debate.

- If computer programs exist which perform the desired functions, but require complete redesign and coding to be compatible with the new CPCI and its operating environment, their "incorporation" into the new CPCI is not properly specified in this paragraph. Selected algorithms, if fully verified, might be specified as design requirements (in basic paragraph 3.2.n, directly or by reference). Preferably, their documentation might be made available to the contractor, separately from the specification, for use at his own discretion (or risk) in designing the new item.

3.14 PARAGRAPH 3.3, DATA BASE

> 3.3 Adaptation. These paragraphs shall specify, in descriptive and quantitative terms, the data base requirements which affect the design of the CPCI. In addition, where applicable, these paragraphs shall specify the methods necessary to convert these parameters into a form suitable for use by the computer program. These requirements are divided into three classes: general environment, system/equipment parameters, and system/equipment capacities and shall be presented as follows:
>
> 3.3.1 General environment. This paragraph shall contain a description of environmental data detailing characteristics anticipated for all particular installations. Each installation will select and set the required data and value for operational use. Examples of such data are: grid limits, radar ranges and areas of coverage, prescribed safety limits, etc.
>
> 3.3.2 System parameters. This paragraph shall contain a description of constants required by one or more subprograms that may change from time to time incrementally within a specified range according to operational needs. Such data consists of allowable trajectory deviations, missile performance characteristics, ranges of possible values, accuracy/precision and quantities, etc.

> (Recommended entry for CDRL backup instructions)
>
> 3.3 Adaptation; 3.3.1 General environment; 3.3.2 System parameters. Delete instructions for these three paragraphs and replace by the following:
>
> 3.3 Data base. Data base requirements which affect the design of the CPCI shall be defined fully in subparagraphs herein, including precise definitions of all data items, together with units of measure, ranges of values, and accuracies/precision where applicable. The data base encompasses all data to be coded and inserted into the system prior to operation of the CPCI. Data definitions contained herein may be organized into categories which are meaningful and appropriate to the given CPCI, e.g., of general environment, variable parameters, or others.
>
> Where the CPCI is intended for operation at multiple sites, or in various mission modes, and is to be adapted for those uses by the insertion of data values specific to each site or mission, this paragraph shall identify and define all such adaptation data separately for each site or mission. When so indicated, data definitions may be provided in separate volumes or appendices.

Purposes of the change recommended in the sample CDRL backup entry provided above are:

- **To alleviate problems which have resulted from a conflict in the meaning of "adaptation", as the title for paragraph 3.3, vs. its established Air Force meaning as defined in the backup instruction.**
- **To provide greater flexibility in tailoring the specification of data base requirements to different individual CPCIs.**
- **To separate the specification of data base requirements from the specification of system capacities (see 3.15).**

3.14.1 **Notes**

The statement in the above backup instructions that the data base "encompasses all data to be coded and inserted into the system prior to operation of the CPCI" is somewhat deceptive in its simplicity. In practice, situations tend to arise which indicate that no single, simple rule applies in the same way to all CPCIs. The comments below identify a few additional rules, together with a few sources of potential confusion, to be considered in determining what the data base should consist of, for purposes of the Part I specification.

a. System Requirements vs. CPCI Design. Here, "data base" is to be interpreted strictly in the layman's language. It refers to files or tables of data whose contents are of interest to the intended operational user, which can be accessed by the operating CPCI for purposes of retrieval and display, summarizing, or other processing uses specified for the Part I specification functions.

It is a source of frequent and significant confusion that data specified here will become (when coded) only one portion of the data base (software language) to be later specified in the CPCI Part II specification. The latter is always more extensive, primarily because it includes all transient or intermediate data being stored temporarily during CPCI operation. The computer program designers may also construct and use some tables of data to perform certain internal CPCI control functions. However, to the extent that those considerations are matters of computer program design techniques, they are not addressed in the Part I specification.

b. Data Base vs. Other Specified Data. Data to be specified here as part of the data base are data which are needed as inputs to one or more of the functions specified in paragraphs 3.2.x, but which are not specified elsewhere in the Part I specification as any one of the following:

(1) messages input from sources external to the CPCI;

(2) data output by another internal CPCI function (specified in one of the output paragraphs, 3.2.x.3); or

(3) data for which values are specified as part of specifying the processing requirements, directly in 3.2.x.2 for an affected function (e.g., mathematical constants used in equations).

For purposes of setting forth requirements in the Part I specification, it is often optional whether certain data values are specified in the last-mentioned category (i.e., directly in a given paragraph 3.2.x.2) or as data base. That determination should be made by the Part I specification writer on the basis of such factors as convenience, bulk of the data definitions, considerations of specification maintenance, whether the data apply to more than one Part I specification function, and--for some data and some CPCIs--whether it is anticipated that some of the values will later be systematically inserted or changed for desired operation of the CPCI at different times or places.

c. Fixed vs. Variable Values. The requirement that the data be coded and inserted into the system prior to operation of the CPCI is subject to certain further qualifications:

(1) It may not apply literally to the entire CPCI in all cases. As in many batch processing or "data base management" capabilities, some system CPCIs may include functions which generate and maintain the data base, as well as other functions which make processing uses of the current data base at the time of their operation. The reference in the instructions is primarily to the latter type of function.

(2) Even when actual values are specified in the Part I specification, to be later included in the CPCI code at the time of its first delivery, data are as subject to change as any other elements of the CPCI--and often more so. When it is desired to specify an initial value, but to provide for future modifiability, the data may be specified as "variable parameters". The specific purpose of specifying a variable parameter is to require that the CPCI be designed to accommodate later changes of the initial value in specified increments, and over a specified range, without affecting the computer program logic or design.

(3) For some classes of data, the actual values may not be known in advance of developing the computer program, and the intent is to provide capabilities for the user to insert the values after the CPCI becomes operational. Tables of security data to be employed for control of data base access by multiple users would be one example. This situation might apply to all of the adaptation data (see below) in some systems. In a mobile, tactical system, for example, it may be desired to provide capabilities for the user to insert regularly various elements of environmental data suitable to a given location or mission.

d. Adaptation Data. Adaptation data are defined as data whose values are fixed for a given site or mission but may vary for different sites or missions. This concept was initially formalized for Air Force use in an early air defense system, which was to be installed for operation at 20 or more site locations. The requirement was for a mission CPCI having the same basic configuration at all sites, with the exception that it be "adapted" to each site at the time of (or prior to) installation by inserting fixed values for geographical coordinates, airbase designators, and numerous other environmental data appropriate to the location. Requirements pertaining to adaptation data in paragraph 80.12.1.2,e of MIL-STD-483 for the version description document, and in paragraph 60.5.3.3.1 for the Part II specification, refer specifically to the class of data described here.

In that application, the concept of adaptation data was the same as the current concept of CPCI "types" (see 3.1.2), in that it was employed basically as a device to avoid having a multiplicity of CPCIs when a single CPCI would suffice with only minor alterations to its configuration. In that particular case, adaptation data are specified in two ways in the Part I specification:

(1) For the basic CPCI, data items are specified as variables--i.e., providing data definitions, units of measure, ranges of values, and tolerances, as applicable (such aspects as frequencies and timing, which are significant to specify for inputs from external sources, do not normally apply). This set of requirements will later affect the computer program design, which should be accomplished to accommodate the specified ranges of values without requiring changes in the basic computer instructions.

(2) For each "type" (location), the actual value is provided for either all or a selected subset of the data items--i.e., in the form of alphabetic characters, names or abbreviations, and decimal numbers. These requirements later affect the computer program code, in that each "type" delivered with a given CPCI version contains the coded actual values for that site.

e. <u>Configuration Management Considerations</u>. In the example described above, it happened that the variations among the various site locations were confined to adaptation data. However, the association between "types" and adaptation data established by that precedent is somewhat incidental:

- Differences in adaptation data are likely to be characteristic of types, but differences in computer instructions should not necessarily be ruled out if they are minor and can be accomplished in a manageable way (see ref. 10, para. 2.3).

- As indicated in c(3) above, provisions for the CPCI to accommodate adaptation data changes may be indicated even if the CPCI is not classified into more than one type.

Suggested answers to a few questions which arise regarding that distinction are summarized briefly as follows:

(1) What difference does it make in specifying adaptation data in the Part I specification whether CPCI types are identified or not?

(a) If types are identified, as in the air defense system example, at least some of the adaptation data will be specified at the level of fixed but different values for different sites, in addition to being specified as variables for the basic CPCI [see d(2) above].

(b) If types are not identified, all adaptation data items are specified as variables.* In addition, capabilities for the user to insert the actual values later, either prior to or during CPCI operation (e.g., via manual inputs), are also specified. --In this case, the adaptation data values do not affect either the CPCI types or versions (see below).

(2) What is the relationship between a CPCI type and a version? --Multiple types are indicated when the intent is to require--in the Part I specification--that one CPCI be developed for initial delivery in some number of different configurations. The intent is to use and support that given number of separate types, concurrently, for the life of the item. Versions of the CPCI will occur as changes are proposed, approved, and implemented to the CPCI's basic configuration as defined in its specification. Each new version or interim version applies to the CPCI as a whole.

(3) How do configuration control procedures apply to adaptation data? --All requirements specified in the Part I specification for adaptation (or any other) data require formal Class I change processing and approval by the central configuration control board (CCB) for the given CPCI. Hence, individual data items should be specified only as variables [as described in Note d(1) above] when it is intended that control of actual values be exercised by the operational user or a local support activity.

3.14.2 *Examples*

Samples of ways in which elements of the data base were specified for an air defense system are presented in Figues 18 and 19. They illustrate: (Figure 18) tables of actual values for data to be coded and incorporated into the CPCI for use in interceptor guidance calculations; and (Figure 19) adaptation data specified in the form of variables, as discussed in 3.14.1,d(1) above. In the latter case, actual values unique to each site were contained in a number of separate volumes supplied by the using command and incorporated into the specification by reference.

---

*Adaptation data may consist of data which could also be classified as environmental data, security data, or other. The point is that the "adaptation" label is a management concept, referring to <u>any</u> class of data which is intended to be changeable for purposes of adapting the CPCI to a given set of operating circumstances. The specific intent may be either: (a) to provide for the future insertion of actual data values, under local controls, without having to change the specification (in which case, the data are specified only as variables); or (b) to permit different actual data values to be specified for different locations or use, without having to classify each of those different configurations as a separate CPCI.

3.3.2.2 Manned Interceptor Characteristics. MCP shall utilize the data values recorded in tabular form below for the performance predictions and guidance calculations unique to each interceptor that are specified for Weapons Control processing in 3.2.5.4.2 herein. Definitions of symbols and terms specified for the Weapons Control processing requirements apply to the symbols and terms used in these tables. [illegible] interceptor, identified in [illegible]

Table VIII. F9 Level Flight Speeds (in TMN).

ALTITUDE (feet)	V_{MIN}	V_{MC}	V_{MAXP}
1,000	.33	.43	.83
5,000	.33	.48	.83
10,000	.38	.53	.83
15,000	.38	.59	1.03
20,000	.43	.66	1.03
25,000	.48	.68	1.18
30,000	.48	.76	1.38
35,000	.58	.82	1.48
40,000	.58	.90	1.43
45,000	.63	1.00	1.28

Table IX. Stabilized Level Fuel Consumption (in Pounds per Minute).

ALTITUDE (feet)	V_{MC}		V_{P2}		V_{MAXP}	
	CLEAN	TANKS	CLEAN	TANKS	CLEAN	TANKS
5,000	98	103	215	235	1210	1295
15,000	84	96	178	195	1280	1130
25,000	79	93	140	148	918	970
35,000	94	100	305	552	637	671
40,000	99	106	370	506	531	633
45,000	281	380	336	459	425	572

Table X. F9 Performance.

SPEED and ALTITUDE PARAMETERS		CLEAN	TANKS
ALTITUDES:			
Optimum Cruise Altitude (Z_O)	feet	20,000	15,000
Optimum Acceleration Altitude (Z_A)	feet	20,000	20,000
Maximum Maneuverable Cruise Altitude (Z_{MC})	feet	35,000	35,000
Coaltitude Reference (Z_C)	feet	40,000	40,000
SPEEDS:			
Optimum Cruise Speeds at Z_O (V_O at Z_O)	IMN	.81	.76
Maximum Maneuverable Cruise Speed at Z_A (V_{MAX} at Z_A)	IMN	1.20	1.11
Minimum Snapup Speeds V_{MS})	IMN	1.18	1.00
PROFILES:			
Profile 1 (V_{MAXP} at Cruise Altitude)			
Prime Cruise Altitude (Z_{PI})	feet	35,000	35,000
V_{MAXP} at Z_{PI}	IMN	1.18	1.00
Profile 3 (V_O/V_{MC} at Cruise Altitude)			
Prime Cruise Altitude (Z_O)	feet	25,000	20,000
V_O at Z_O	IMN	.81	.76
Profile 4 (1.2V_T at Cruise Altitude)			
Prime Cruise Altitude (Z_O)	feet	25,000	20,000
PERFORMANCE TIMES: (brake release to Z_O)	minutes		
Buster Climb		14.0	14.6

Figure 18. Sample Data Base Specification - Interceptor Characteristics.

3.3.5 Adaptation Data. The Mission Computer Program (MCP) shall be designed as a common, adaptable computer program that will operate in any one of the ten Sector Control Centers (SCCs) of the 4XXL System. The parameters defined in Table XXXII herein shall apply to all processing specified for MCP in 3.2 above wherein adaptation data values are identified as required inputs to the processing operations. Actual values to be coded and incorporated into MCP prior to installation at each SCC are those contained in the identically-structured tables of adaptation data values provided for the ten individual SCCs in Appendixes XI through XX of this specification.

Table XXXII. Sector Environmental Data Parameters.

ADAPTATION ITEMS	UNITS	LIMITS	ACCURACY/ PRECISION	REMARKS
SECTOR COORDINATE CENTER - GEOGRAPHIC COORDINATE CENTER OF THE LOCAL CENTER				
ℓ_0 Conformal Latitude	Radians	$0 \le \ell_0 \le 157.0770$	4.84814×10^{-6}	Positive North
λ_0 Longitude	Radians	$0 \le \lambda_0 \le 314.1540$	4.84814×10^{-6}	Positive West
R_E Earth Radius	N.Miles	$R_E \le 5000$	1/16	At Coord. Center
SECTOR DISPLAY CENTER - LOCAL SECTOR'S DISPLAY CENTER ON STEREOGRAPHIC PLANE				
X_D, Y_D Components	N.Miles	$0 \le X_D, Y_D \le 1024$	1/4	Sector Center Ref.
RADAR SITE - PARAMETERS FOR EACH RADAR SITE INTERFACING WITH THE LOCAL SECTOR				
Site ID	LDDD	-- Unique designator for each radar site --		
ℓ_R Conformal Latitude	Radians	$0 \le \ell_R \le 157.0770$	4.84814×10^{-6}	Positive North
λ_R Longitude	Radians	$0 \le \lambda_R \le 314.1540$	4.84814×10^{-6}	Positive West
R_E Earth Radius	N.Miles	$R_E \le 5000$	1/16	At Radar Site
A_R Altitude	Feet	$0 \le A_R \le 12000$	100	Above Sea Level
X_R, Y_R Components	N. Miles	$0 \le X_R, Y_R \le 1024$	1/16	Sector Center Ref.
HEIGHT FINDER SHADOW AREAS - PARAMETERS FOR EACH HF AT THE RADAR SITE				
A_Z Azimuth	Degrees	$0 \le A_Z \le 360$	1.0	Sector North Ref.
ΔA_Z Limits	Degrees	$\Delta A_Z \le 10$	1.0	
COMPUTER I/O ASSIGNMENTS - RADAR COMMUNICATION CONNECTIONS TO I/O CHANNELS				
P_I Primary Input	Chan No.	00-64	DNA	
P_O Primary Output	Chan No.	00-64	DNA	
S_I Secondary Input	Chan No.	00-64	DNA	
S_O Secondary Output	Chan No.	00-64	DNA	
GATR SITE - PARAMETERS FOR EACH GATR SITE INTERFACING WITH THE LOCAL SECTOR				
Site ID	LDDD	-- Unique designator for each GATR site --		
X_G, Y_G Components	N. Miles	$0 \le X_G, Y_G \le 1024$	1/16	Sector Center Ref.
R_G Range Radius	N. Miles	$0 \le X_G, Y_G \le 1024$	1.0	

Figure 19. Sample Data Base Specification - Adaptation Data (Variables).

3.15 PARAGRAPH 3.4, SYSTEM CAPACITIES

> 3.3.3 <u>System capacities</u>. This paragraph shall contain a description of the capacity requirements for the computer program. Items such as compatibility for total simultaneous target handling, total number of simultaneous missile trajectory controls, total number of simultaneous displays and operator station requests, track capacity, number and types of inputs processed, etc, shall be described. The system capacities are directly related to computer storage capacities, interfacing subsystem timing rates, and interfacing equipment capabilities.

> (Recommended entry for CDRL backup instructions)
>
> 3.3.3 <u>System capacities</u>. Delete this paragraph.

The renumbering of this paragraph as paragraph 3.4 corresponds with changes to the "Adaptation" paragraph, noted in the preceding section herein (3.14), based on a draft version of MIL-STD-490A. The recommended CDRL backup instruction to delete the paragraph entirely derives from the failure of the instructions to identify how or whether any requirements should be specified here which (a) should not already have been specified in preceding paragraphs of the specification, or (b) are really the proper level of requirements to specify in the Part I CPCI specification. Specifically:

- The statement that "system capacities are directly related to computer storage capacities, ..." appears to be included for explanatory purposes. However, it suggests an orientation more towards computer program design than towards requirements; and it provides no clue to how those capacities, rates, and capabilities differ from requirements to be specified as design requirements (in para. 3.2.n) or as interfaces (in para. 3.1.1).

- "Number and types of inputs processed" is directly redundant with the requirements to be specified in the various inputs paragraphs, 3.2.x.1.

- The reference to "compatibility for total" simultaneous targets, missile trajectory controls, etc. are subject to various interpretations. They <u>might</u> refer to general requirements set forth in the system specification. However, the responsibility to be compatible with the system specification rests primarily with the developers of the Part I specification itself. The repetition of system requirements here is merely redundant, at best, if those have been properly allocated, analyzed, and converted into detailed requirements for the CPCI interfaces, inputs, outputs, processing, design constraints, and data base specified in preceding paragraphs. If not, the repetition is unlikely to alleviate resulting deficiencies and/or direct conflicts among Part I specification requirements.

3.16 SECTION 4, QUALITY ASSURANCE PROVISIONS

<u>Section 4 Quality assurance provisions.</u> Requirements for formal verification of the performance of the CPCI in accordance with the requirements of section 3 of this specification shall be specified in this paragraph. Formal verification of performance of the CPCI shall determine acceptance of the CPCI. This paragraph shall specify formal verification requirements to a level of detail which:

a. Designates verification requirements and methods in section 4 for performance and design requirements in section 3. The methods of verification to be specified herein may include inspection of the CPCI, review of analytical data, demonstration tests, and review of test data.

b. Specifies requirements for verification to the level of detail necessary to clearly establish the scope and accuracy of the test method.

c. Permits ready identification of each verification requirement specified in section 4 with the appropriate performance/design requirement paragraph in section 3.

d. Allocates verification requirements to the subparagraphs included herein.

NOTE: This section shall not incorporate, either directly or by reference, detail test planning documentation and operating instructions. Requirements specified herein shall be the basis for preparation and validation of such documents.

3.16.1 General

Note that the above instructions summarize requirements for Section 4 as a whole; they do not specify content to be provided immediately under the title of the section. In a single-volume specification, no content is necessarily indicated. When the specification is prepared as a document series, statements should be provided here along the following lines:

- In Volume 1 - "Quality assurance provisions are specified in this section for the (title) CPCI as a whole, and for all of its functions."

- In each other volume - "Quality assurance provisions for the (title) function(s) are specified in Volume 1 of this specification."

As stated in the MIL-STD-483 instructions, the real emphasis of this section is on specifying requirements for <u>formal</u> tests/verifications which will establish, to the procuring activity's satisfaction, that the CPCI meets performance and design requirements specified in Section 3. Thus, Section 4 should not attempt to include requirements for testing to be carried out by the developer as an integral part of the CPCI development process. The latter ("informal testing", or CPT&E) is included only to the limited extent outlined in 3.19 below.

As indicated in the NOTE contained in the instructions, planning information as such (schedules, numbers and sequencing of tests, support needs, and procedures) is not to be documented in Section 4 of the development specification. This section contains the contractually-applicable test objectives/criteria to govern the formal test program for the CPCI, against which the adequacy of later-approved test plans and procedures can be evaluated. At the same time:

- **Planning information for CPCI qualification should normally be developed concurrently with, and in a mutually-supporting relationship to, development of the Section 4 requirements (see 3.23). But it shoud be documented separately in the form of a preliminary version of the formal test plan (DI-T-3703) for the CPCI.**

- Planning of the developer's informal test program should be accomplished and described in the computer program development plan (CPDP). This planning provides the basis for the developer's "requirements" documented in paragraph 4.1.2 of the specification for test equipment, facilities, and/or other Government support of his CPT&E, including (when necessary) time for installation and checkout of a mission CPCI at the system DT&E site.

3.16.2 Notes on Test Terms and Concepts.

a. Verification vs. Test. "Verification" is used here in its accepted English-language meaning--i.e., referring to the determination that something exists or is true. As used in the above instructions specifically, it is a more appropriate term than "test", since it implies that the determination can be accomplished by various methods. Note that "test" is used at times with that same broad meaning (equivalent to verification), and at times in the more limited sense of one particular method--namely, an experimental exercise yielding performance data--by which the verification is accomplished.

b. Qualification vs. Acceptance Testing. "Qualification testing" refers to the process of verifying (by all applicable methods) that a newly-developed item meets the requirements of its development specification. Successful completion of qualification is the primary basis for procuring activity acceptance (approval, and physical transfer of possession) of the item. "Acceptance testing" refers to routine tests/verifications conducted on production units of an equipment item against requirements specified in Section 4 of its product (Part II) specification; hence, it never applies to a CPCI. (This distinction between qualification and acceptance testing is firmly established only in the Air Force; it is not uniformly observed by other Government agencies.)

3.17 PARAGRAPH 4.1, INTRODUCTION

> 4.1 Introduction. This paragraph shall establish the requirements which provide the basis for development of a test plan and test procedures for the subject program. All test/verification requirements shall be specified within the subparagraphs included herein.

General or introductory statements may be made in this basic paragraph to explain the organization of requirements or, when relevant, to identify any significant objectives or policies which apply to the given CPCI or system program. Examples:

- "Test subcategories and methods for verifying individual performance/design requirements specified in Section 3 are summarized in the Verification Cross Reference Index, Table (number). Narrative data pertaining to each test subcategory are specified in the subparagraphs below. Amplified descriptions of test methods applicable to individual Section 3 requirements are specified in 4.2 and subparagraphs thereto."

- "Verification of (title or abbreviation of the CPCI) compliance with basic performance requirements set forth in Section 3 of this specification shall be accomplished using the CPCI configuration adapted to the (name or number) site location. Adaptation of the CPCI to remaining site locations will be accomplished by the Government, subsequent to completing system DT&E."

- "Modified compiling capabilities specified for this CPCI in 3.2.x.1 through 3.2.x.3 shall be verified through formal qualification testing specified in 4.1.4 below. Other functions of this CPCI shall be qualified through its successful use in supporting the development and qualification of the (titles) CPCIs."

3.18 PARAGRAPH 4.1.1, CATEGORY I TEST

> 4.1.1 Category I test. The term "category I test" as used herein is defined to include all testing of the CPCI other than that accomplished during the formal category II (or equivalent) system/configuration item test programs. (See paragraph 4.1.5 below.) Category I testing is subdivided into the following broad types:
>
> a. Computer programming test and evaluation - Tests conducted prior to and in parallel with preliminary or formal qualification tests. These tests are oriented primarily to support the design and development process.
>
> b. Preliminary qualification tests - Formal tests oriented primarily towards verifying portions of the CPCI prior to integrated testing/formal qualification tests of the complete CPCI (see paragraph 4.1.3 below). These tests will typically be conducted at the contractor's design and development facilities.
>
> c. Formal qualification tests - Formal tests oriented primarily towards testing of the integrated CPCI, normally using operationally configured equipment at the category II site prior to the beginning of category II testing. This testing will emphasize those aspects of the CPCI performance which were not verified by preliminary tests. The testing requirements which cannot be verified during category I test shall be specified in paragraph 4.1.5.
>
> NOTE: Requirements for verification included in the system/system segment specification, which are directly related to requirements specified herein, may be incorporated herein by reference to avoid redundant establishment of the requirements.

The "category test" terms contained in the above instructions have been outdated as a result of revisions which appeared first in the 12 May 1972 issue of AFR 80-14. In the specification, they should be replaced (via authority provided in CDRL backup instructions, pending the issuance of MIL-STD-490A) by the current terms in accordance with the following simple conversion:

Category I Test → CPCI Development Test & Evaluation (CPCI DT&E)

Category II Test → System Development Test & Evaluation (System DT&E)

Other terms used to distinguish the three types (subcategories) of Category I test--namely, CPT&E, PQT, and FQT--and their definitions as provided above, are unaffected by those changes.

Again, note that the instructions provide information only, defining the types of testing under CPCI DT&E to be covered in remaining subparagraphs of 4.1. They do not specify the content to be provided in this basic paragraph. Depending on applicability to the given CPCI, statements may be provided here to clarify the test policy with respect to defined subcategories of tests along lines suggested by the following examples:

- **"Qualification of this CPCI shall be accomplished during CPCI DT&E to the maximum extent possible, as a result of PQTs and FQTs conducted by the developer and witnessed/verified by the procuring activity."**

- **"Qualification of selected requirements during CPT&E shall be limited to tests conducted under controlled conditions, using documented procedures. Results of such tests shall also be documented, and the developer shall make such documents available as evidence of verification to representatives of the procuring activity during functional configuration audit (FCA) of the CPCI."**

- **"Qualification of requirements during system DT&E shall be limited to requirements which cannot be demonstrated satisfactorily during CPCI DT&E due to the absence of the full equipment configuration and operating intersystem communications."**

3.19 PARAGRAPH 4.1.2, COMPUTER PROGRAMMING TEST AND EVALUATION

> 4.1.2 Computer programming test and evaluation. This paragraph shall contain the following:
>
> a. Programming test and evaluation which satisfy one or both of the criteria listed below shall be included herein. (Routine tests accomplished in support of design and development, which do not satisfy one or both of these criteria, shall not be specified herein.)
>
> (1) They are intended to be the only source of data to qualify specific requirements in section 3.
>
> (2) They must be accomplished as part of an integrated test program involving other systems/equipment/computer programs (eg, verification of requirements in paragraph 3.1.1.)

"CPT&E" is the label applied to the developer's internal (i.e, informal) testing accomplished as a part of his CPCI development effort, in contrast to formal testing for purposes of qualification. It is assumed* that a developer has internal plans and procedures for conducting CPT&E, ranging from code checking and debugging through CPC and performance testing at successively higher levels, and that those are adequate to meet the technical needs of computer program development. The conduct of the internal test program as a whole, however, is largely irrelevant to the purposes of this section (Quality Assurance Provisions) of the development specification. As stated in the instructions, material should not be specified here about CPT&E as such (or elsewhere in Section 4) except where Government recognition is specifically needed for the two reasons stated. Those two types of "requirements", and the ways in which they should be specified, are discussed separately in the two subparagraphs below.

3.19.1 Test Requirements.

A "test requirement", in Section 4, is generally a statement that a given performance or design requirement set forth for the CPCI in Section 3 is to be verified during a given subcategory (type) of DT&E, by an identified method (see 3.21 below). Successful verification constitutes qualification of the CPCI with respect to the given requirement.

Although CPT&E is not basically a part of the formal test program, it is

*Or judged independently of this specification. For example, descriptions of planning for internal testing are required as a portion of information to be supplied by a developer in his computer program development plan (CPDP).

recognized as one possible source of data by which the formal qualification can be accomplished for some Section 3 requirements. For CPCIs, the judicious use of CPT&E for that purpose should be considered seriously from the point of view of reducing the burden on formal tests/demonstrations, in which the number of requirements that can be demonstrated, in depth, is typically limited. Requirements of a detailed nature which are relatively independent of interfacing operations, or certain requirements whose verification entails lengthy compilation and/or analysis of test data, are likely candidates.

When CPT&E is identified as the source of qualification, however, those selected tests must be conducted and documented by the developer in a manner which is more formal than may be typical of his internal test program as a whole: The tests must be conducted strictly in accordance with documented test procedures, including listings of input data; and the test results must be fully recorded, including hardcopy printouts or comparable records of test performance and outputs. Formal qualification of the individual requirements occurs at a subsequent functional configuration audit (FCA) on the basis of the evidence provided by that documentation, together with any supporting evidence which may be derived from CPCI performance during formal tests/demonstrations.

Section 3 requirements selected for qualification via CPT&E may be identified directly in this paragraph (4.1.2) of the specification. However, the preferred approach to specifying requirements in this (and following) paragraphs is to: (a) construct a summary matrix (the Verification Cross Reference Index; see Figure 20) which lists all Section 3 requirements for the CPCI, identifying the test subcategory and verification method for each; and (b) limit the narrative information provided directly in this paragraph to the following statements:

- A statement that requirements to be qualified through this test subcategory are identified in Column (number) of the Verification Cross Reference Index.

- Additional narrative statements that may be necessary to further define, clarify, or delimit aspects of the requirements to be verified. These should include, but are not necessarily limited to, clarifications of specific intent for individual requirements that are listed in more than one test subcategory, in the Verification Cross Reference Index. (The possibility of multiple listings is discussed further in 3.20 and 3.21, below.)

In effect, the basic content of Section 4 as a whole is provided by the Verification Cross Reference Index. Additional information is contained directly in (a) each of the paragraphs, 4.1.2 through 4.1.5, as necessary to clarify the nature or scope of individual Section 3 requirements to be verified in the given test subcategory, and (b) paragraph 4.2 to further clarify the specific applicability to identified requirements of the general test methods defined in paragraph 4.1.4 (see 3.21 below).

TABLE XXXIV. VERIFICATION CROSS REFERENCE INDEX

<u>Method Legend:</u>

NA - Not Applicable
1 - Inspection
2 - Analysis
3 - Demonstration
4 - Review of Test Data

<u>Test Legend:</u>

A - Computer Programming Test & Evaluation
B - Preliminary Qualification Test
C - Formal Qualification Test
D - System Test

NOTE: The verification category, test category, and test requirements specified for each paragraph shall apply to all subparagraphs therein which are not separately listed in this index.

Section 3 Requirement Reference	Verification Method					Test Subcategory				Test Requirement
	N/A	1	2	3	4	A	B	C	D	
3.1										4.2.1
3.2	X									
3.2.1	X									
3.2.1.1	X									
3.2.1.1.1		X						X		4.2.2
3.2.1.1.2		X						X		4.2.2
3.2.1.1.3	X									4.2.3
3.2.1.2	X									
3.2.1.2.1	X									
3.2.1.2.1.1				X				X		4.2.4
3.2.1.2.1.2				X				X		4.2.6
3.2.1.2.1.3			X				X			4.2.5
3.2.1.2.1.4				X				X		4.2.4
3.2.1.2.1.5					X	X				4.2.7, 4.2.9
3.2.1.3	X									
3.2.1.3.1				X				X		4.2.11
3.2.1.3.2				X				X		4.2.11
3.2.1.3.3				X				X		4.2.11, 4.2.12
3.2.1.3.4				X					X	4.2.13
3.2.2	X									
3.2.2.1	X									
3.2.2.1.1		X						X		4.2.2

Figure 20. Sample Verification Cross Reference Index. This matrix is prepared as a table, normally consisting of several pages, which should be located in the specification at the end of Section 4.

3.19.2 Support Requirements.

A response to the phrase "...integrated test program involving other systems/equipment/computer programs..." is always pertinent when a Government agency or contractor other than the given CPCI developer controls the availability of computer equipment, facilities, interfacing CPCIs, or trained operational personnel that may be needed to perform the development or to prepare for upcoming formal test sessions. Hence, statements made here should specify the developer's needs (requirements), as applicable, in such areas as:

- Availability of the computing equipment, including consoles and other input or display devices, specifying minimum equipment configuration vs. phasing if appropriate.

- Availability of the system test facility(ies) for purposes of computer program installation and checkout.

- Availability of other support to the CPT&E program if needed--e.g., interfacing CPCIs or trained user personnel.

Note that similar information about the developer's needs for Government support in those areas is not mentioned elsewhere in the instructions, pertaining to formal tests. Additional coverage would normally be redundant, since CPT&E includes internal testing in preparation for conducting the formal tests. It is generally desirable to carry out PQTs and FQTs as efficiently-conducted formal demonstrations. If the developer has "done his homework" properly, he will have verified the capabilities to be demonstrated (via tests, corrections, and retests as needed) in advance of each formal session.*

---

*Questions have been raised resulting from comparing the interpretations presented above with (a) requirements to be included in the CPCI test plans/procedures and (b) statements made about these support requirements in Volume II of AFR 800-14:

(a) It is true that these requirements are to be included in the CPCI test plan and procedures. However, instructions for the latter state that they "will normally correspond to requirements set forth in paragraph 4.1.1 of the Part I CI specification". See DI-T-3703, paragraphs 1,f,(6) and 2,f; note that the first sentence of paragraph 1,f,(6) confirms the interpretation presented here.

(b) AFR 800-14 (Volume II, paragraph 5-5,e) also confirms that the requirements described here are to be contained in the specification, but does not clearly interpret them as being covered by the "integrated test program..." phrase. It (1) adds a (redundant) statement to that effect, and (2) omits reference to the use of informal test results for qualification.

3.20 PARAGRAPH 4.1.3, PRELIMINARY QUALIFICATION TESTS

> 4.1.3 Preliminary qualification tests. This paragraph shall specify only those preliminary qualification test requirements which require formal recognition by the Air Force and are oriented toward verifying proper performance of portions of the CPCI prior to integrated testing of the complete CPCI. Testing accomplished by the contractor in support of design and development which does not require recognition by the Air Force, other than it is within the general terms and conditions of a contract, shall not be specified herein. Requirements for preliminary qualifications specified herein shall reference requirements in section 3.

PQTs are parts of the formal test program, in that, like FQTs, they: (a) are scheduled in the CPCI test plan, preceded by delivered test procedures, and followed by deliverable test reports; (b) are attended and witnessed by the procuring activity; and (c) are conducted to demonstrate identified Section 3 requirements of the development specification.

PQTs differ from FQTs, basically, in that they are conducted on "portions of the CPCI" before formal testing of the integrated CPCI is initiated (cf. 3.18 above). Other differences, and rules for their use in actually accomplishing qualification, are not clearly established in current Air Force policy. The comments below suggest a number of considerations to be examined and resolved in the course of tailoring this part of Section 4 to the needs of a given system program and CPCI.

- The instructions in MIL-STD-483 for paragraph 4.1.1 regarding different locations for PQT (contractor's plant) and FQT (system test site) are subject to some freedom of interpretation. While the full system environment is typically essential to completing FQTs of the system CPCIs, it is generally recognized that: (a) most support CPCIs should be qualified earlier; and (b) as much of the FQT for mission CPCIs should also be conducted at the contractor's plant as can be validly accomplished at that location. Purposes of the latter are to promote confidence in satisfactory completion of their development before committing them to system test, and to reduce demands for extensive CI-level testing at the system test site.

- The basic intent of PQTs is to provide formal points of visibility to the procuring activity, between CDR and FQT, of the developer's interim progress towards achieving an acceptable end product--i.e., a "confidence-building" function. Nominally, a PQT is intended (based on hardware precedent), not to qualify in itself, but to demonstrate that the portion(s) being tested operate well enough to justify their inclusion in formal qualification testing of the integrated item. Thus, "preliminary qualification" really means preliminary to qualification, assuming that characteristics demonstrated during PQT will also be demonstrated (either explicitly or implicitly) during formal qualification. From that point of view, many of the Section 3 requirements identified for verification during PQT may also appear in FQT.

- However, the use of both PQTs and selected results of CPT&E (see 3.19 above) for qualification should be considered in the light of objectives for the test program as a whole. Available time and resources do not normally permit fully verifying the compliance of a complex CPCI with the totality of its Section 3 requirements, even if maximum use is made of all available testing opportunities. While PQTs are necessarily limited to requirements that can be demonstrated satisfactorily through operation of the given portions (i.e., groups of CPCs), they also lend themselves to greater attention to requirements of a detailed nature than may be appropriate or feasible during FQT. Hence, when PQTs are scheduled, the attempt should be made to include as many requirements as can legitimately be verified in this test subcategory. The necessity to repeat verification of those same requirements during FQT should then be judged on the basis of such factors as criticality, effects of integrated operation with portions of the CPCI not tested, and the likelihood of the tested requirements being altered during subsequent CPCI development.

- Experience suggests that the total number of PQTs should not generally exceed 3 or 4, even for a very large and complex CPCI, and that those should be spaced appropriately to their primary purpose of providing confidence-building visibility during the development period (ref. 11, Ch. VI). As noted in 3.16 above, some test planning should be accomplished concurrently with the preparation of Section 4. While the requirements specified in Section 4 itself do not identify, for example, how many PQTs (or FQTs) will be conducted, they should be formulated in conjunction with at least a minimum of advance planning along those lines. In the case of PQTs, the requirements that can be properly identified in Section 4 depend on the planned allocations of Section 3 functions to CPCs, together with the planned sequencing of individual CPC development and assembly.

Requirements identified for verification during PQT should be specified in the same manner as outlined above (3.19.1) for CPT&E.

3.21 PARAGRAPH 4.1.4, FORMAL QUALIFICATION TESTS

> 4.1.4 Formal qualification tests. This paragraph shall specify requirements for formal qualification tests of the integrated CPCI to demonstrate and/or verify that the requirements established in section 3 have been satisfied. This paragraph shall, in subparagraphs as appropriate, specify the requirements and method of verification for the requirements specified in section 3, with the following exceptions:
>
> a. The requirement in section 3 has been identified, and verification that it has been satisfied by one of the tests included in paragraphs 4.1.2 and 4.1.3.
>
> b. The requirement is section 3 is peculiar to category II type system testing and will be identified in paragraph 4.1.5.
>
> Verification of the requirements may be accomplished by inspection, demonstration, test, and review of test data, or combinations of these. This paragraph shall contain a subparagraph for each of the principal methods of verification, and shall specify therein the requirements of section 3 to be verified by the method.

The general aim of testing at the CI level is to qualify all Section 3 requirements during FQT. For reasons indicated above, it is usually desirable to shift some of the burden to CPT&E and/or PQTs, however, particularly for individual requirements involving numerous minor details or time-consuming techniques to accomplish their verification. Hence, the practical emphasis in FQT is normally placed on requirements which are critical, and on those which can only be verified during operation of the integrated CPCI.

The specification of requirements is accomplished, in this paragraph, in the same manner as outlined above (3.19) for paragraph 4.1.2. In accordance with the above instructions, this paragraph should cover all Section 3 requirements which are not specified in paragraphs 4.1.2, 4.1.3, or 4.1.5. It should also include (a) requirements which are specified in 4.1.2 and 4.1.3, but for which full qualification via those tests is not intended, and (b) requirements specified in 4.1.5 which are to be verified partially, or in a preliminary way, during FQT. When a given requirement appears in more than one test subcategory in the Verification Cross Reference Index, the intent with respect to full qualification is clarified in narrative statements contained directly in the affected paragraphs (i.e., 4.1.2 through 4.1.5).

In addition, the principal methods of verification should be explained in a separate subparagraph under this paragraph, along lines of the sample content illustrated in Figure 21.

4.1.4.2 Verification Methods. The four methods identified in the Verification Cross Reference Index for verifying individual Section 3 requirements are explained as follows:

a. Inspection - Formal verification of compliance with a requirement by examination of the assembled CPCI and its design documentation at the time and place of qualification testing. Inspection is the principal method by which design requirements specified in paragraph 3.2.n of the development specification are verified. It may also apply to selected requirements in other areas, for example: to verify adaptation data through comparison of data base documentation with sample printouts of coded data contained in the CPCI.

b. Analysis - Formal verification of a performance requirement by examination and study of the computer program design and coding. For example, verification of compliance with a weapons guidance equation and specified tolerances may be accomplished through analysis of algorithms and the flow of input data through successive stages of processing. This method is typically tedious and time-consuming.

c. Demonstration - Formal verification of performance characteristics through observation of functions being performed by the operating computer program. This is the basic method by which most qualification of a CPCI is normally accomplished with respect to its Section 3 performance requirements. Examples include: ability of the CPCI to accept specified inputs; performance of specified control actions; and format, content, and timing characteristics of display or other CPCI outputs. Verification is accomplished at the time and place of the demonstration (test).

d. Review of Test Data - Review of test records for tests/demonstrations accomplished at an earlier time. This method is typical of requirements tested during CPT&E, but may also apply to other requirements which depend on a series of tests over more than one test occasion or under varied conditions of operation. Verification is typically accomplished by review of (1) detailed test procedures, including input data, and (2) hardcopy printouts of CPCI test outputs.

Figure 21. Sample Statements Explaining Verification Methods.

3.22 PARAGRAPH 4.1.5, CATEGORY II SYSTEM TEST PROGRAM

> 4.1.5 Category II system test program. This paragraph shall identify requirements specified in section 3 which cannot be verified until category II testing (or equivalent) and must be listed as a category II test requirement.

For support CPCIs which are relatively independent of system mission functions, this paragraph is normally "not applicable".

For mission CPCIs, the basic requirements of Section 3 should normally be specified, throughout, as they pertain to operation of the item in the full system operational environment. However, qualification testing during PQTs and FQTs is typically accomplished utilizing something less than the full configuration of system equipment or personnel, and/or utilizing simulated rather than live inputs from external sources. Thus, for individual Section 3 requirements which cannot be satisfactorily verified under those conditions, this paragraph provides for completing their qualification during system DT&E.

Note that the reference is not merely to testing at the system DT&E location, but to verification during the course of actual system-level DT&E. An alternative to be considered, when appropriate and necessary, is to conduct all or portions of FQT for the mission CPCI during a period of CI/subsystem DT&E which may be held at the system test site, immediately prior to the beginning of full system DT&E.

Requirements identified for verification during system DT&E are specified in the manner outlined above (1.19.1) for CPT&E.

3.23 PARAGRAPH 4.2, TEST REQUIREMENTS

> 4.2 Test requirements. This paragraph shall specify the requirements for each type of testing. The requirements shall include test formulas, algorithms, techniques and acceptable tolerance limits, as applicable.

This paragraph should consist of statements which further clarify the specific intent with respect to how general verification methods defined in paragraph 4.1.4 (see 3.21) apply to individual Section 3 requirements.

The material to be provided in this paragraph is of particular importance to the specification as a whole, in that its function is to delineate techniques --including limitations--of verification in such a way as to provide a feasible and realistic basis for the formal CPCI test program. It has been observed that full verification of a complex CPCI is often a practical impossibility, in view of the endless permutations and combinations that can occur under operational conditions. Considered in that light, objectives of these statements are to define and delimit each verification requirement so that (a) it will provide adequate assurance to the procuring activity that the CPCI complies with its required performance, and (b) it constitutes a requirement which can reasonably be met by the developer, within applicable constraints of time and resources.

Hence, the statements listed in this paragraph should be derived from a careful consideration of each Section 3 requirement from the point of view of how it can be verified in a manner which is not only adequate, but also feasible and attainable during the DT&E program. In a sense, they largely represent assessments of the practical "testablility" of various Section 3 requirements. They should be judged for their adequacy in the light of such factors as: effects on costs and complexity of the formal test program; expected stability of the CPCI configuration following its initial qualification; and criticality of the given Section 3 requirement(s) to system operations. Those factors tend to vary widely for different requirements, CPCIs, and systems. A few examples are provided in Figure 22. Note that:

- Each statement is provided in a separate, numbered subparagraph.
- Statements are referenced in the "Test Requirements" column of the Verification Cross Reference Index (see Figure 20), for applicable Section 3 requirements.
- A given statement may apply to more than one Section 3 requirement, and vice versa.

4.2 <u>Test Requirements</u>. The following test requirements shall apply to Section 3 performance and design requirements as specified in the Verification Cross Reference Index, Table XXXIV, within the intent of verification methods defined in 4.1.4.

4.2.1 Interface requirements specified in subparagraphs of 3.1 shall be verified through verification of requirements affected by those interfaces in other paragraphs of Section 3.

4.2.2 Data base inputs and simulated external message inputs shall be verified by inspection of hardcopy printouts at the time and place of qualification testing. CPCI acceptance of all inputs, simulated or live, shall be verified implicitly by proper performance of the functions using those inputs.

4.2.3 Inputs from other functions internal to this CPCI shall be verified implicitly by proper operation of the specified function with its interfacing functions.

4.2.4 Direct verification of this function shall be limited to demonstrating proper detection and processing of error conditions that can be generated by modifying jumper wires, PC card removal, or removal of power from isolated components.

⋮

4.2.n This function shall be verified through verification of specified output displays resulting from manual inputs at the console, following a script of manual actions.

Figure 22. Examples of Test Requirements to be Specified in Paragraph 4.2.

3.24 SECTION 6, NOTES

<u>Section 6 Notes.</u> This section shall include information which is stated here for administrative convenience only, and is not a part of the specification for the CPCI in the contractual sense (i.e., it shall not include requirements which constrain design, development, and qualification of the CPCI and require compliance by the contractor. The text may be preceded with the statement "Administrative Information Only - Not Contractually Binding." This section of the specification shall include information of particular importance to the procuring activity in using this particular specification as a contractual instrument for acquisition of the CPCI either initially or for follow-on procurement.

Background information or rationale which will be of assistance in understanding the specification itself or using the CPCI it specified, may be included herein (e.g., technical data ordering instructions).

As may be inferred from the above instructions (and similar general policies for this section set forth in MIL-STD-490), there is no positive requirement that any material be included in this section at all. Effectively, it is the one place in the specification where the writers may include, for background or explanatory purposes, certain types of information which are not properly contained in other sections or in appendices (see 3.25 below). The significant point is that information in this section is not really a part of the specification, in that it does not add to or otherwise qualify any requirements stated in the basic specification and is not directive on the developer of the CPCI.

The most prominent function of this section, in a CPCI Part I specification, is to provide certain key items of background or explanatory information to aid in understanding selected requirements set forth in the text (i.e., in other sections). Likely candidates are: information about operator procedures; functions of related CIs or systems; notes referencing relevant system engineering or design trade studies; or derivations of selected equations (cf. 3.9.2,b).

MIL-STD-490 (paragraph 3.2.13) provides for the possibility that Section 6 may contain an organized list of definitions, to which parenthetical reference may be made when terms are used in the text. Considering the policy governing Section 6 in general, that appears to be an option which should be employed sparingly, either to consolidate definitions which are also provided directly in the text or for terms defined for information only. Terms whose definitions are necessary to make the specified requirements precise should be defined in the text; each term may be defined when it is first used, and/or by reference to an organized list provided elsewhere, e.g., in an appendix.

3.25 SECTION 10, APPENDIX I

> Section 10 Appendix I. *This section of the specification* shall contain requirements which are contractually a part of the specification but which, for convenience in specification maintenance, are incorporated herein (e.g., requirements of a temporary nature or for limited effectivity). Appendixes may be bound as separate documents for convenience in handling (e.g., when only a few parameters of the program are classified, an appendix containing only *the classified material* may be established). Where parameters are placed in an appendix, the paragraph of section 10 shall be referenced in the main body of the program specification in the place where the parameter would normally have been specified. Typical data that may be included in computer program development specification appendixes include:
>
> a. Mathematical derivations
>
> b. Alternate method
>
> c. Summary of equations
>
> d. Definitions of terms

> (Recommended Entry for CDRL Backup Instructions)
>
> Section 10 Appendix I. Delete the examples: a. Mathematical derivations; and b. Alternate methods.

Suggested rules for use of the appendix form, and a few examples, are included in various preceding discussions (see especially: 2.3.2, Figure 4, Figure 5, and 3.8.2,b).

The important point of the basic instructions for this section is that an appendix, unlike the Notes section (Section 6), is an integral part of the specification in that its content is contractually binding on the developer. Hence, *it is recommended* that the first two examples listed below the instructions be deleted, since they both appear to be in direct conflict with that policy.

Questions are occasionally raised about the reasons for excluding information in such categories as alternative methods and mathematical derivations, based on the observation that information in those areas is often helpful to the computer program developer (*and to those who have to* evaluate a completed specification). The point is made, for example, that errors in mathematical equations are notoriously frequent, and the availability of their derivations can often aid in detecting and correcting those errors.

As noted above (3.24), Section 6 is the one place in the specification where information of that nature may be recorded. It is obviously limited, in that Section 6 should be brief relative to the specification as a whole. However, it should be recognized that those restrictions placed on content of the

specification are based soundly on its intended functions, and they do not imply at all that the information should not be available:

- System engineering and computer program design study documentation resulting from the Part I Specification development process should also be available to the developer, in addition to the specification. In a normal system program, he would have generated much of that information himself--including the mathematical derivations, trade studies, detailed operator task analysis data, etc.

- The requirements and constraints placed on specifications stem from a variety of considerations, many of which have been mentioned in preceding portions of this guidebook. For example, the computer program development specification should be written as a directive document, delineating as precisely as possible the CPCI's required characteristics in order to best serve its primary function as a legal instrument governing the procurement of that item.

It should be noted that the specification is constrained not only with respect to the types of system engineering data mentioned above, but also to exclude many other levels of related information which are contained in such documents as: development, configuration management, and quality assurance plans; test plans, procedures, and reports; positional handbooks and user manuals. In a system program, it is the normal assumption that the CPCI Part I specification will constitute a key element, but will be properly integrated with--and supplemented by--those other elements of the "software documentation" structure as a whole.

SECTION 4. REFERENCES

1. AFSCM/AFLCM 375-7, Configuration Management for Systems, Equipment, Munitions, and Computer Programs. 31 March 1971.*

2. MIL-STD-483 (USAF), Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs. 12 April 1971.

3. MIL-STD-1521A (USAF), Technical Reviews and Audits for Systems, Equipment, and Computer Programs. 1 June 1976.

4. MIL-STD-480, Configuration Control - Engineering Changes, Deviations, and Waivers. 30 October 1968.

5. MIL-STD-490, Specification Practices. 30 October 1968.

6. DoD 5220.22-M, Industrial Security Manual for Safeguarding Classified Information.

7. AFSCP 800-3, A Guide for Program Management. 9 April 1976.

8. DI-E-3119A, Computer Program Development Specification. 17 April 1972.

9. Glore, J. B. Software Acquisition Management Guidebook: Life Cycle Events. ESD-TR-77-22; Electronic Systems Division, AFSC, Hanscom AFB, Mass. February 1977. (NTIS Accession No. AD-A037115)

10. Searle, L. V. An Air Force Guide to Computer Program Configuration Management. ESD-TR-77-254; Electronic Systems Division, AFSC, Hanscom AFB, Mass. August 1977.

11. Searle, L. V., Rosove, P. E., and Sydow, E. H. Systems Management Applied to Large Computer Programs in BUIC III; Review of Experience. ESD-TR-69-302; Electronic Systems Division, AFSC, Hanscom AFB, Mass. June 1969. (DDC Accession No. AD 699-585)

---

*Now superseded by AFSCP 800-7, Configuration Management, 1 December 1977. Changes which may have been introduced in AFSCP 800-7 are not reflected in the text of this guidebook.

SECTION 5. ABBREVIATIONS

AFLC	Air Force Logistics Command
AFSC	Air Force Systems Command
C^3	Command, Control, and Communications
CDRL	Contract Data Requirements List
CI	Configuration Item
CPC	Computer Program Component
CPCI	Computer Program Configuration Item
CPDP	Computer Program Development Plan
CPT&E	Computer Programming Test and Evaluation
DID	Data Item Description
DoD	Department of Defense
DT&E	Development Test and Evaluation
ECP	Engineering Change Proposal
ESD	Electronic Systems Division
FQT	Formal Qualification Test(ing)
JCS	Joint Chiefs of Staff
N/A	Not Applicable
O/S	Operating System
OT&E	Operational Test and Evaluation
PDR	Preliminary Design Review
PQT	Preliminary Qualification Test(ing)
RFP	Request For Proposal
SAM	Software Acquisition Management
SDC	System Development Corporation
SDR	System Design Review
SRR	System Requirements Review
TBD	To Be Determined