

DEPARTMENT OF THE NAVY  
NAVAL AIR SYSTEMS COMMAND

①

AD-A273 630



ADVANCED AVIONICS ARCHITECTURE  
AND TECHNOLOGY REVIEW  
FINAL REPORT

EXECUTIVE SUMMARY  
&  
VOLUME 1 - AVIONICS TECHNOLOGY  
VOLUME 2 - AVIONICS SYSTEMS ENGINEERING

6 August 1993

Under the Direction of  
Avionics Systems Engineering Division, AIR-546

Government Participants

NAVAL AIR SYSTEM COMMAND  
SPACE AND NAVAL WARFARE SYSTEMS COMMAND  
NAVAL RESEARCH LABORATORY  
NAVAL AIR WARFARE CENTER, AIRCRAFT DIVISION-  
INDIANAPOLIS & WARMINSTER  
NAVAL AIR WARFARE CENTER, WEAPONS DIVISION-CHINA LAKE  
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY  
NATIONAL INSTITUTE FOR STANDARDS AND TECHNOLOGY

DTIC  
ELECTE  
DEC 10 1993  
S E D

Industry Participants

AAIA • Alliant Techsystems • Analog Devices • Apple Computers • ARINC  
AT&T • Boeing Aircraft • C3 • Computing Devices Intl • Convex Computer  
Douglas Aircraft • Draper Laboratories • DY 4 Systems • Raytheon  
GEC Avionics • General Dynamics • Grumman Aerospace  
Harris Corporation • Honeywell Corporation • Hughes Aircraft • IBM  
IEEE Electronics Integration Committee • Intel • Litton Guidance Systems  
Lockheed Aircraft • Lockheed Sanders • Loral Defense Systems  
McDonnell Aircraft • Microlithics • Mitre Corp • Motorola  
National Semiconductor • Northrop • Paramax • Rockwell Collins • SAE  
SBS Engineering • SUN • Tamarack Storage Devices • Texas Instruments  
TRW • TimePlus • Westinghouse • Zycad

93-30034



244pb

Approved for public release  
Distribution

93 12 9 00 6

# REVIEW TEAM MEMBERSHIP

## CHAIRMAN

CAPTAIN Frederick G. Schobert  
AIR-546  
Naval Air Systems Command  
Arlington, VA

Mr Joseph J. Carr  
AIR-546M  
Naval Air Systems Command  
Arlington, VA

Mr Jay Crawford  
Naval Air Warfare Center,  
Weapon Division  
China Lake, CA

Mr Andrew S. Glista Jr.  
AIR-546TE  
Naval Air Systems Command  
Arlington, VA

Mr Louis Neri  
AIR-546E6E  
Naval Air Systems Command  
Arlington, VA

Mr Ralph Lachenmaier  
Naval Air Warfare Center,  
Aircraft Division,  
Warminster, Pennsylvania

Mr Jerry Murdock  
NGCR Program,  
Space & Naval Warfare Systems  
Command, Arlington, VA

Mr Victor Skullman  
AIR-546  
Naval Air Systems Command  
Arlington, VA

Mr David Kaplan  
Naval Research Laboratory  
Washington, D.C.

## VICE CHAIRMAN

COMMANDER Charles E. Jewett  
AIR-546E  
Naval Air Systems Command,  
Arlington, VA

Mr Rodney S. Katz  
Naval Air Warfare Center,  
Aircraft Division,  
Indianapolis, Indiana

Mr Chuck Caposell  
AIR-546TE  
Naval Air Systems Command  
Arlington, VA

Mr Jules P. Letellier  
Naval Research Laboratory  
Washington, D.C.

Mr Eric Alfonsi  
Naval Air Warfare Center,  
Aircraft Division,  
Warminster, Pennsylvania

Mr Brian Hardman  
Naval Air Warfare Center,  
Aircraft Division,  
Indianapolis, Indiana

Dr. John Newport  
Naval Air Warfare Center,  
Aircraft Division,  
Indianapolis, Indiana

Mr William Schibler  
AIR-546E2  
Naval Air Systems Command  
Arlington, VA

Mr Robert Westbrook  
Naval Air Warfare Center,  
Weapon Division  
China Lake, CA

**REPORT DOCUMENTATION PAGE**

1a. REPORT SECURITY CLASSIFICATION <b>UNCLASSIFIED</b>		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION NAVAL AIR SYSTEMS COMMAND	6b. OFFICE SYMBOL (if applicable) AIR-546M	7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State, and ZIP Code) COMMANDER, NAVAL AIR SYSTEMS COMMAND AIR-546M 1421 JEFFERSON DAVIS HWY. ARLINGTON, CA 22243-5460		7b. ADDRESS (City, State, and ZIP Code)	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO.	PROJECT NO.
		TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) ADVANCED AVIONICS ARCHITECTURE AND TECHNOLOGY REVIEW FINAL REPORT, INCLUDES: EXECUTIVE SUMMARY; VOLUME 1; AVIONICS TECHNOLOGY; VOLUME 2; AVIONICS SYSTEMS ENGINEERING			
12. PERSONAL AUTHOR(S)			
13a. TYPE OF REPORT FINAL	13b. TIME COVERED FROM 7-92 TO 8-93	14. DATE OF REPORT (Year, Month, Day) 1993 AUGUST 6	15. PAGE COUNT 244
16. SUPPLEMENTARY NOTATION INDUSTRY PROVIDED A SIGNIFICANT AND ENTHUSIASTIC RESPONSE BY PARTICIPATING IN A TECHNICAL SURVEY THAT CONTRIBUTED GREATLY TO THIS EFFORT. HIGH COST OF AIRCRAFT MODIFICATIONS INTEGRATION COSTS ARE EXCESSIVE AND RESULT IN PROTRACTED PROGRAM EXECUTION AND CONCEALED (over)			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
		AVIONICS, ARCHITECTURE, TECHNOLOGY (JIAWG) JOINT INTEGRATION AVIONICS WORKING GROUP; (NGCR) NEXT GENERATION COMPUTER RESOURCES, INTEGRATED, JOINTNESS, COMMANAVITY, (OVER)	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) THE PURPOSE OF THIS STUDY WAS TO REVIEW THE CURRENT TECHNOLOGY BASE, DEVELOPING TECHNOLOGIES, AND THE POTENTIAL TO APPLY THEM TO MEET MILITARY AVIONICS REQUIREMENTS. THE REVIEW ALSO EXAMINED THE TECHNOLOGY INTEGRATION METHODS USED IN THE AVIONICS COMMUNITY IN ORDER TO DEVELOP A MORE EFFICIENT AND MORE AFFORDABLE AVIONICS ARCHITECTURAL STRATEGY FOR THE FUTURE. IT PROVIDES A BOLD VISION FOR THE APPLICATION OF ADVANCED TECHNOLOGY TO NAVAL AVIONICS. THIS STUDY WAS ACCOMPLISHED AS A COOPERATIVE VENTURE BY MEMBERS OF THE NAVAL AVIATION SYSTEMS TEAM AND OVER 50 LEADING FIRMS FROM THE ELECTRONICS AND AEROSPACE INDUSTRIES. THE STUDY TEAM ALSO INCLUDED REPRESENTATIVES FROM THE AIRFORCE, ADVANCED RESEARCH PROJECTS AGENCY, NATIONAL INSTITUTE FOR STANDARDS AND TECHNOLOGY, AND THE DEFENSE SYSTEMS MANAGEMENT COLLEGE. THEIR GOAL WAS THE REVIEW EMERGING TECHNOLOGIES, INVESTIGATE THEIR POTENTIAL COST SAVINGS. THE JOINT AERONAUTICAL COMMANDERS GROUP (JACG) WILL NOW USE THIS STUDY AS A BASIS FOR DEVELOPING AND IMPLEMENTING A BETTER ACQUISITION PARTNERSHIP BETWEEN THE MILITARY SERVICES AND INDUSTRY. THIS REPORT IS THE ONLY COMPREHENSIVE COMPILATION OF TODAY'S AVIONICS TECHNOLOGY (OVER)			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input checked="" type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL CHARLES E. JEWETT, CDR, USN		22b. TELEPHONE (Include Area Code) (703) 692-5789 ex 2807	22c. OFFICE SYMBOL AIR-546M

BLOCK 16 (continued)

MODIFICATION PROPOSEALS. THESE PROBLEMS ARE PARTIALLY DUE TO A LACK OF A COMMON AIRCRAFT ARCHITECTURE.

BLOCK 19 (continued)

IT IS A GREAT REFERENCE BOOK THAT CONTAINS EASILY UNDERSTANDABLE DESCRIPTIONS OF TODAY'S TECHNOLOGIES, INCLUDING BUS ARCHITECTURES, BACK PLANES, PROCESSORS, SOFTWARE, AND JUST ABOUT EVERY ASPECT OF AVIONICS DESIGN AND DEVELOPMENT. IT IS A REFERENCE BOOK THAT SHOULD BE RETAINED.

BLOCK 18 (continued)

STANDARDIZATION, SYSTEMS ENGINEERING, CONCURRENT ENGINEERING, MODULAR, PI-BUS FUTUREBUS ±, OPEN SYSTEM, INTEGRATION, COMMERCIAL - OFF - THE - SHELF (COTS) NON-DEVELOPMENTAL ITEM (NDI), PACICAGING, SOFTWARE, PEOCRATED, ADA, MODULARITY



DEPARTMENT OF THE NAVY  
 NAVAL AIR SYSTEMS COMMAND  
 NAVAL AIR SYSTEMS COMMAND HEADQUARTERS  
 WASHINGTON, DC 20361

IN REPLY REFER TO

5230  
 Ser AIR-546/NP495/0661  
 13 Sep 93

MEMORANDUM FOR THE DISTRIBUTION LIST

From: Commander, Naval Air Systems Command

Subj: ADVANCED AVIONICS ARCHITECTURE AND TECHNOLOGY REVIEW  
 FINAL REPORT

1. It is with great pleasure that I forward the Advanced Avionics Architecture and Technology Review Final Report. It provides a bold vision for the application of advanced technology to naval avionics.
2. This study was accomplished as a cooperative venture by members of the Naval Aviation Systems Team and over 50 leading firms from the electronics and aerospace industries. The study team also included representatives from the Air Force, Advanced Research Projects Agency, National Institute for Standards and Technology, and the Defense Systems Management College. Their goal was to review emerging technologies, investigate their potential for application to naval avionics, and identify potential cost savings. The Joint Aeronautical Commanders Group (JACG) will now use this study as a basis for developing and implementing a better acquisition partnership between the military services and industry.
3. This report is the only comprehensive compilation of today's avionics technology that I have seen. It is a great reference book that contains easily understandable descriptions of today's technologies, including bus architectures, backplanes, processors, software, and just about every aspect of avionics design and development. I recommend it as a reference book that you shall want to retain.

W. C. BOWES  
 Commander

Distribution:  
 (see next page)

DTIC QUALITY INSPECTED 3

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification .....	
By .....	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	



Approved for public release  
 Distribution unlimited

Subj: ADVANCED AVIONICS ARCHITECTURE AND TECHNOLOGY REVIEW  
FINAL REPORT

Distribution:

USD (R&E)  
ASN (RDA)  
ASN (RD&A/AW)  
CNO (N6)  
AFMC (CC)  
AFMC (ASC/CC)  
ATCOM (CG)  
HQMC (AP)  
DLA (DD)  
FAA (ASU-300)  
CG (G-Ad)  
NASA (R)

(additional distribution to be accomplished separately)



# Advanced Avionics Architecture & Technology Review

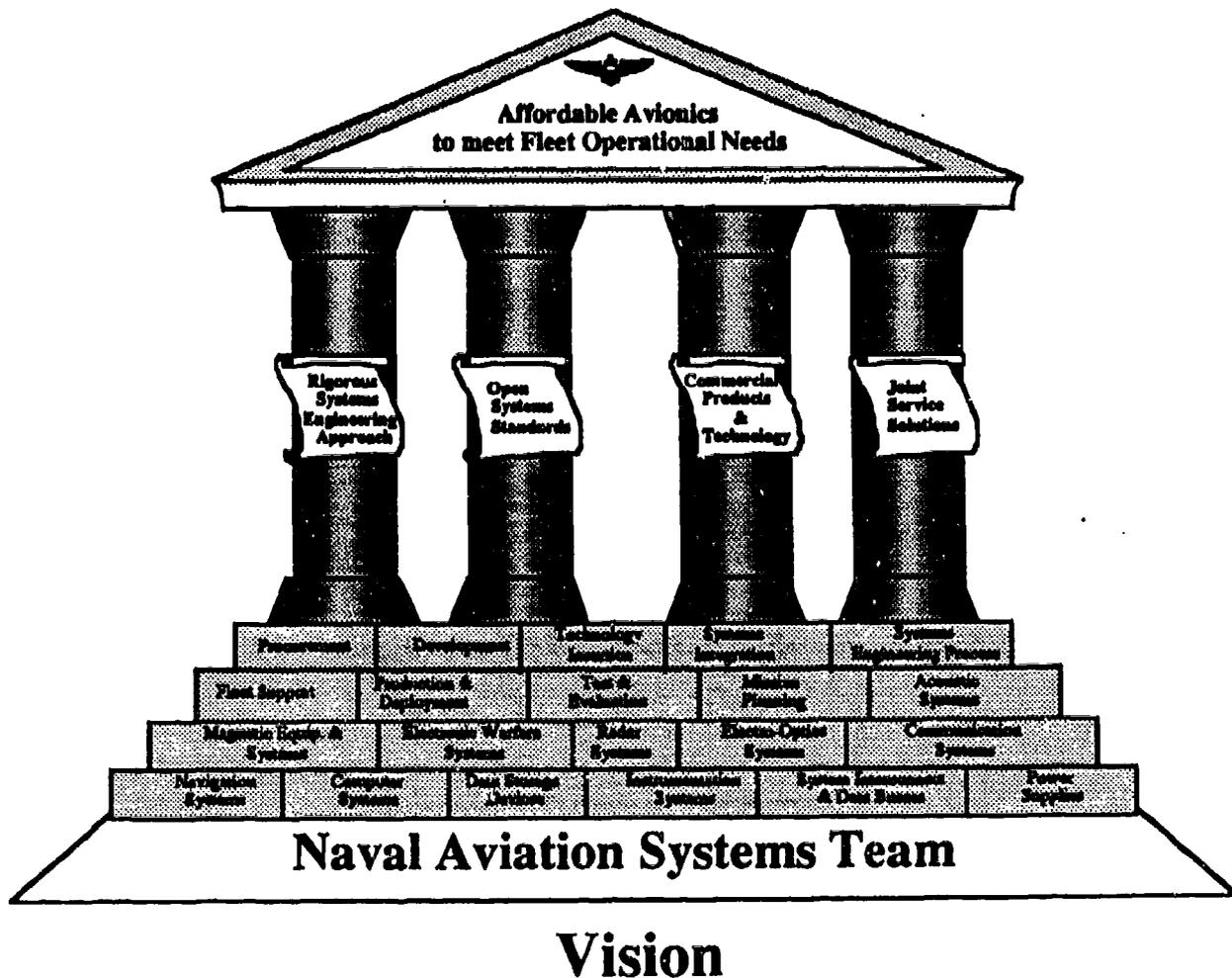
August 1993



Naval Air Systems Command  
Department of the Navy  
1421 Jefferson Davis Highway  
Arlington, Virginia 22243-5460

## Advanced Avionics Architecture & Technology Review Achievement of Success

This review was tasked to assess currently available and emerging technologies that are relevant to Naval avionics acquisition. The review fosters a Naval Aviation "vision" for delivering the best avionics capability to the Fleet. The vision encompasses the key enabling technologies to be used as building blocks and the four principal thrusts that become the "pillars" for the successful implementation by trained personnel of the Naval Aviation Systems Team and industry. The report fulfills the tasking and provides the vision, findings, and recommendations as summarized below.



## Findings

- No single architecture will meet all avionics requirements.
- System complexity demands the use of a rigorous systems engineering methodology.
- Industry supports DOD's use of an open systems approach.
- Use of published standards will increase contract competition and lower cost.
- Modular avionics will lower systems development and support cost.
- A transition to commercial technologies and products will save government development and support dollars.
- High payoffs are projected if sensor design standards are updated and followed.
- Affordable system development will depend on an improved software development process.
- Software reuse capability will enhance systems affordability only if matured.
- Ada should remain DOD's software language of choice.
- Joint service acquisition makes sense.
- Technology insertion can be made affordable.

These findings can be consolidated into a single conclusion:

The best methodology for designing naval aviation's avionics systems is a well defined standards based approach, applied through a rigorous systems engineering methodology.

## Recommendations

Based on the findings of this study, the following recommendations are made. It is recommended that the Navy:

- Transition from the use of multiple unique avionics point designs to an architectural approach that capitalizes on open system opportunities and joint initiatives.
- Become more active in the standards organizations that are responsible for applicable open system standards.
- Establish a limited number of architectural choices through a managed program of technology tracking, standards selection, risk assessment and technical trade-off studies.
- Continue to promote the use of COTS technology and products in military systems.
- Adopt a strategy for managing optimal COTS use.
- Enhance our commitment to joint service programs.
- Support the Avionics Engineering Sub-board of the JACG
- Invest in the software development infrastructure now — to establish long term software affordability.
- Establish a formal system engineering process to guide architectural choice, engineering standards selection, and implementation of cost-effective technological solutions for Navy systems.

**This page intentionally left blank.**

## Master Table of Contents

- Executive Summary - Presents the top level findings and recommendations of the study.  
*(Intended for all levels of decision makers involved with avionics acquisition).*
- Volume One: - Avionics Technology provides a review of key avionics technologies, processes and development programs.  
*(Intended for technical decision makers and to serve as an executive reference for avionics technologies).*
- Volume Two: - Avionics Systems Engineering presents an overview of Systems Engineering and discusses its application to avionics acquisition.  
*(Intended for avionics managers, systems integrators, and systems design engineers)*
- Volume Three: - Industry Survey Responses presents industry's response to an intensive avionics development questionnaire.

Note: *Volume Three is published separately and is limited to U.S. Government Distribution Only due to extensive use of proprietary/competition sensitive information.*

**This page intentionally left blank.**

---

# EXECUTIVE SUMMARY

---

---

## Purpose

"... high cost of aircraft modifications ... integration costs are excessive and result in protracted program execution and canceled modification proposals. These problems are partially due to a lack of a common aircraft architecture. ..."

VADM J.O. Tuttle, USN  
3/2/92

The purpose of this study was to review the current technology base, developing technologies, and the potential to apply them to meet military avionics requirements. The review also examined the technology integration methods used in the avionics community in order to develop a more efficient and more affordable avionics architectural strategy for the future.

---

## Background

The Navy continues to make a significant investment in complex avionics systems that are developed uniquely for Navy aircraft. It must be determined whether the Navy can transfer more of this investment burden to industry and still meet requirements. Controlling costs, while still acquiring leading-edge technologies, requires a well thought out plan for managing future investments. Help was needed to understand emerging technologies and predict their usefulness to, and availability for, naval avionics applications.

To begin formulating this plan, a team of government avionics experts was assembled. The team developed and executed a research and information gathering strategy which polled industry to determine present-day and future technology trends in advanced avionics. The second phase required a structured analysis to identify important findings and consensus positions. During this phase, issues such as multi-mission platforms, extending life capability, and compatibility with other Navy systems and other services were addressed. The third phase involved formulating individual findings and recommendations which are the principle products of this report.

Industry provided a significant and enthusiastic response by participating in a technical survey that contributed greatly to this effort.

# Findings

## No Single Architecture Will Meet All Avionics Requirements

Architectures: incorporate the complete set of factors that contribute to making an avionics suite into a system that works as a unified, functional whole to accomplish the mission of the aircraft.

- No single avionics architecture can be both cost-effective and functionally efficient when forced to perform all missions or fit all applications.
- A single architecture, rigidly applied, limits necessary design flexibility among platforms and discourages timely technology insertion.
- Architecture choice is driven by operational requirements, environmental requirements, and the legacy of older aircraft.
- Industry supports specifying a limited number of open architectures from which to work.

It was found that the range of functional needs among platforms is so diverse that no known architecture would satisfy all requirements. The goal of a single architecture to meet the most demanding user requirements would necessarily add unneeded capabilities to other user's applications, driving costs prohibitively upward. Additionally, a single architecture hinders design optimization and flexibility. On the other hand, too many architectural choices can obscure standardization, also driving costs upward. To achieve cost-effective and efficient solutions to avionics applications, a balance must be struck.

The Navy must be knowledgeable on all the various building blocks that become available for avionics applications. These building block elements usually represent today's state-of-the-art technologies that continue to evolve. At the same time, the Navy has a unique challenge.

The Navy must balance affordability, diverse and changing requirements, emerging threats and the fact that industry focuses on technology trends that have the greatest potential benefit in commercial rather than military markets.

## System Complexity Demands The Use Of A Rigorous Systems Engineering Methodology

**Systems Engineering:**  
 "An interdisciplinary approach to evolve and verify an integrated and life-cycle balanced set of systems product and process solutions that satisfies customer needs."  
 (MIL-STD-499B)

For a detailed discussion of systems engineering, see Volume 2 of this report.

For a detailed discussion of Systems Engineering tools see Volume 1, sections 9-1 & 9-3.5 of this report.

- Most industry respondents suggested a formal, disciplined systems engineering process to manage a more standards based approach to complex systems design.
- An Integrated Product Development Team (IPDT) approach involving the customer (Navy), the prime contractor, and subcontractors was recommended by industry.
- Current acquisition regulations are a stumbling block to an effective integrated product development team approach.
- Systems engineering ensures that software development, reuse, and metrics are part of the total systems design and development effort.
- Encourage computer based modeling and simulation, as part of the systems engineering process, to reduce cost and risk.

The Systems engineering process provides the means for effective selection and application of standards. Figure 1 depicts the role of the systems engineering process to manage and evolve both products and standards. An architecture is designed by carefully considering requirements and constraints such as cost targets. Just as it was with hardware, software development is undergoing transition from an art to an engineering science which must be embraced as a key, cost-savings part of the systems engineering process.

Systems engineering environments provide tools used in each aspect of the acquisition process from simulation programs used in the conceptual phase to life cycle cost models to compare the overall cost throughout the expected service life of the system. Computerized tools do not replace good systems analysis.

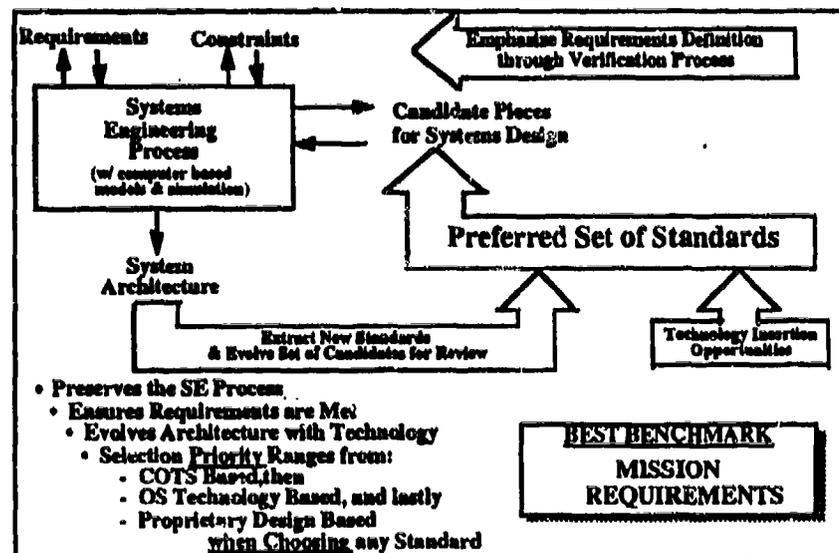


Figure 1: Overview of a Standards-based Systems Engineering Process for Avionics Architecture Development

## Industry Supports DoD's Use Of An Open Systems Approach

### Open System:

One that implements sufficient publicly available specifications for interfaces, services, and supporting formats to enable properly engineered components to inter operate with other components on local and remote systems, to interact with users in a style which facilitates portability, and to be utilized across a wide range of systems with minimal change.

For a detailed discussion of open systems standards architecture approaches, see Volume 1, section 3-5.

- Industry supports an open systems standards-based approach as the best potential method to reduce cost, risk, and development time, while increasing inter operability opportunities.
- Industry supports standard interfaces for common avionics modules, standard data buses and switched networks.
- Open systems standards support a modular building block approach.
- An open systems standards-based approach allows the use of commercially-available products for rapid prototyping.

Technical advances can be more readily accommodated, without adversely affecting existing overall systems design, by adopting an open systems standards approach. The Navy will realize cost savings because an open systems approach allows us to take advantage of commercial standards and their associated products and technology.

The open systems standards approach is not without problems, however; it is not a panacea. For example, in many open standards there are user-defined features, certain functions, memory locations or connectors pins that can be applied differently by different users of these standards. This design flexibility can undercut the sought-after attributes of interoperability and interchangeability.

Blindly following open systems standards without a plan will not save the Navy money. We must track the selected open standards to ensure that Navy requirements and needs are addressed. Promoting the use of a complementary and limited set of open standards will prevent an excessive number of choices and promote rational standardization among avionics systems.

## Proper Use Of Published Standards Will Increase Contract Competition and Lower Cost

**Interface Standards:**  
Standards which define and control how architectural building blocks connect electronically, thermally, mechanically, etc., among the systems that we develop.

**Systems Standards:**  
Standards that address global issues such as systems security, power distribution, system diagnostics and those for system level supportability, e.g., MIL-STD 704.

**Process Standards:**  
Standards that address and define the methods and procedures employed as systems are developed. Examples include those that govern software development and system engineering process.

The most effective way to obtain commonality among the architectural building blocks is to select architectures based on a family of interface, system and process standards. These standards build a framework for an integrated avionics suite. They should cover the areas of data processors, signal processors, RF electronics, sensors, and the various buses and networks that interconnect these building blocks. Interface standards must deal with software as well as hardware. In addition, selected standards should be diverse enough to meet Navy needs, limited enough to support commonality, and flexible enough to permit designers freedom to be creative (yet still be logistically supportable). The recommended order of precedence to be used when applying engineering standards to both new and existing systems is:

- a) Open Systems Standards controlled by non-government standards bodies
- b) Widely-used standards under government control
- c) Standards controlled by special purpose working groups
- d) Proprietary standards for unique interfaces

where the selection process is the result of an engineering analysis that has considered the overall merits and shortcomings of each for the application.

Use of standards lowers cost by increasing contract competition. By breaking the overall system into modular building blocks, the system integrator (or the Navy) is able to competitively buy the component blocks from a variety of different vendors. By establishing interface and systems standards, these blocks, built by different vendors, can be quickly integrated to form the overall system. With well defined standards and reasonably small building blocks, a medium sized company can find a niche need, develop a module at company expense to fill that need, and market that module for use in a number of different systems. Commercial industry feels strongly that future module reliability, and the eventual ability to build modules to standards with software that is instruction set architecture independent, will lead to the military's ability to significantly reduce logistics costs. A failed module built by one vendor can then be replaced by a module built by a second vendor, possibly using a more powerful next generation processor, built to be form, fit, and interface compatible through the use of standards.

Another factor which will help lower cost is dual commercial and military use of the standards. Dual use has been difficult in the past because items built to commercial standards have often not satisfied Navy requirements. This problem can be partially alleviated by early Navy participation in commercial standards development in order to ensure that Navy requirements are considered. The Next Generation Computer Resource (NGCR) program followed this approach.

## **Modular Avionics Will Lower Systems Development And Support Cost**

For more information on programs examining modular integrated architectures, see Volume 1, chapters 8.

For a detailed discussion on use of Modular Avionics see Volume 2, section 17.

A significant trend in avionics design is a move from current federated systems (interconnected black boxes) to highly modular, integrated systems with a high degree of sensor data sharing. Modular packaging has become the preferred approach, where practical, because it provides for effective use of powerful high density microelectronics, increased maintainability, effective cooling techniques, and offers the design feature of backplane communications using standard data paths and data buses.

- Modular integrated architectures can reduce overall avionics weight, volume and power requirements while permitting cost-effective fault tolerance.
- Modular integrated architectures can lower recurring procurement costs by allowing hardware and software asset sharing.
- Modularity provides a cost-effective means of selective technology insertion.

Early modeling and simulation of systems will validate design assumptions, predict system performance against standard benchmarks, and test changes before committing to a specific design. Modularity makes it easier to create and maintain usable systems models.

The attributes of military modular avionics systems (building blocks) are being explored by organizations such as the Navy Standard Hardware and Reliability Program (SHARP).

## A Transition To Commercial Technologies and Products Will Save Government Development And Support Dollars

### Commercial off-the-shelf (COTS):

Hardware and software products, technology and designs purchased through commercial retail or wholesale distributors as is; modified to meet specified functional requirements; or ruggedized to meet service requirements.

- The majority of industry respondents believe that COTS avionics should provide significant cost savings and risk reduction for functional areas that are common to both commercial and military aircraft.
- DoD should cooperate with industry in establishing standards.
- Use of appropriate commercial technology and products provide for incorporation of continual technical advancements at minimal cost to the government.
- Use of COTS allows for early prototyping, early system debug, and earlier software development to reduce total system development time, risk and cost.
- Satisfying stringent avionics environmental and unique system-level requirements is a significant challenge for COTS, but the challenge can be met.

Industry stressed the need to leverage commercial advancements in the electronics and information technology industries. Judicious use of COTS hardware (including ruggedized and militarized variants) and software offers several benefits: a) a larger qualified suppliers list, b) a greater ability to "test drive" new technologies before committing huge scarce financial resources to a full-scale program, and c) the ability to leverage a strong commercial user base to accommodate changes requested by fleet users at lower cost.

But there are also disadvantages to COTS in military applications. For example, few commercial systems require multi-level security or need to maintain real-time operations in a rigorous environment. Addition of these features can lead to a significant level of redesign, which reduces the economic advantages. COTS is not usually designed specifically for environments equivalent to those seen by military avionics. To compensate for this, rigorous testing of COTS must include environmental, inter operability and integration testing to predict integrated performance and reliability necessary for life-cycle planning. COTS solutions might actually require more testing than unique military solutions because lack of detailed documentation and Navy monitoring during the design phase reduces our knowledge of the system.

Figure 2 shows the relative levels of potential cost avoidance at several different levels of COTS technology.

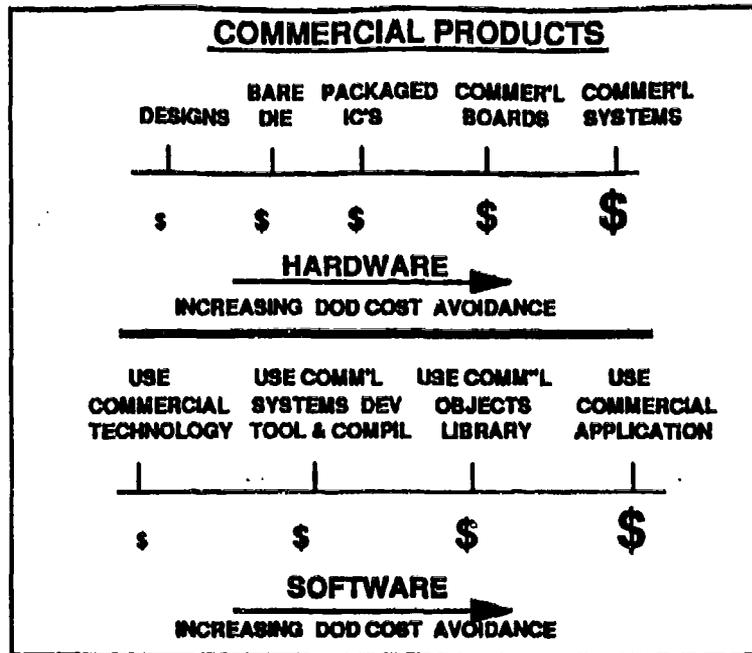


Figure 2: Inverse Relationship Between Cost Avoidance and COTS Utilization

For a detailed discussion of the pros and cons of COTS use in military avionics, see Volume 1, section 3-5.4.

To achieve standardization and required performance. Effective selection criteria and engineering guidelines for the use of COTS in military avionics are needed. Differences between commercial standards and military needs have the greatest chance of reconciliation if the military actively participates in commercial standards development working groups.

---

**High Payoffs Are  
Projected If  
Sensor Design  
Standards Are  
Updated And  
Followed**

- Industry concurs that a method of RF interface standardization should be applied to the sensors domain.
- State-of-the-art avionics is heading towards greater miniaturization, high speed networks, distributed processing and integrated sensors.
- Application of circuit technology from the MIMIC program can provide both performance gains and cost savings at the sensor head (front end) for RF sensor systems.
- Functions such as sensor fusion will be more easily performed in integrated systems.
- Aircraft sensor technology and design has been driven by DoD needs.

Industry reported that between sixty and seventy percent of avionics cost is attributed to sensor systems. This claim is based on historical analyses of federated systems. Federated systems require autonomous capability to provide functionality, driving up sensor system cost and complexity. In newer integrated avionics systems, modular avionics techniques can potentially reduce sensor costs by performing much of the signal processing using standard digital signal processing modules located in avionics compartments remote from the analog sensor heads.

The retrofit of newer sensor technology subsystems into older aircraft requires careful consideration. The benefits of the technology gains could be outweighed by other factors such as the need for advanced cooling techniques and/or increased data processing requirements in excess of what is available in that specific model aircraft.

Although DoD has driven sensor technology in the past, this situation is changing. Increased commercial sensor applications in areas such as meteorology, medical imaging, video teleconferencing, geological and geographical mapping, and communications are now stimulating commercial development of sensor technologies. Because of a common need for developing these technologies, cooperative development and cost sharing may soon be a reality.

**Affordable System  
Development Will  
Depend On  
Improved Software  
Development  
Processes**

- The scale and scope of software development is often underestimated at the start of full-scale development for major programs.<sup>1</sup>
- Systems functionality is moving from hardware to software. This movement is accelerated as systems move from federated to integrated architectures.

Modern Naval aircraft weapon systems incorporate complex avionics systems whose core functions are implemented with software-intensive subsystems that are dependent on the real-time processing of information and data. The development of the software is often more of a cost, schedule, and risk driver at the program level than is the hardware. Although software is perceived to be distinct from hardware, the software engineering process must be part of the overall systems engineering process.

The maturity of an organization's software engineering capability can be measured in terms of the degree to which the success of the next software development can be predicted. To be successful, organizations must be able to accurately predict the amount of time, resources, and cost required to develop software. One measure of an organization's software engineering capability, which the review team and industry both endorsed, is the Capability Maturity-Model (CMM) developed by the DoD Software Engineering Institute (SEI).

For a detailed discussion of the Software Engineering Institute's Capability Maturity Model see Volume 1, section 9-4.1

The CMM addresses the disciplines and processes which should be in place for an organization to produce reliable software in a timely and cost effective manner. The model rests on the premise that software process maturity is a credible indicator of capability and that the productivity and quality resulting from an organization's software process can be improved over time.

The CMM has five levels of maturity and describes Key Process Areas (KPAs) which must be in place to reach the next higher maturity level. The five levels are Initial (Level 1), Repeatable (Level 2), Defined (Level 3), Managed (Level 4), and Optimizing (Level 5). Achieving the next higher level of process maturity indicating both greater control of an organization's software process and greater consistency with which the process is applied in projects throughout the organization. The higher the level achieved, the more predictable is an organization's software process. A well defined and planned software measurement program is required to progress through the levels of the CMM. A noted weakness of the CMM is its failure to account for domain knowledge and personnel skill which both contribute to capability.

<sup>1</sup> A General Accounting Office Audit Report, specifically, GAO/IMTEC-92-48 titled as EMBEDDED COMPUTER SYSTEMS, A Significant Software Problem On C-17 Aircraft Software Must Be Addressed, Washington DC: The General Printing Office: May 1992.

## TOOLS

For a detailed discussion of the Software Engineering Tools see Volume 1, section 9-3.5

An important part of the software development process is the application of computerized management tools, Computer Aided Software Engineering (CASE) tools, and analysis tools. Depending on the particular avionics system being developed, a broad range of software tools, ranging from database management to software design tools to project management tools, may be required to support the effort.

Software tools can be grouped into general types such as management tools, development tools, and laboratory tools. Management tools include tools to perform planning, scheduling, requirements traceability, configuration control, and documentation. Development tools include structured analyzers, editors, compilers, code generators, debuggers, and emulators. Laboratory tools include automated testers, data manipulators, simulators, and real-time non-intrusive testers. Many of the tools required for developing avionics system software exist today; however, there is little integration between and among these tools. Three tools which have been applied successfully at NAVAIR are *Document Director*<sup>™</sup>, *Statemate*<sup>™</sup>, and *SES/Workbench*<sup>™</sup>. *Document Director*<sup>™</sup> is a full-featured requirements management tool that combines all the features of a word processor with the power of a database management system to provide for the tracking of requirements across complex specifications, procedures, tests plans, and other documents. *Statemate*<sup>™</sup> is a powerful, graphical, system specification/development tool for the validation of complete, consistent, and accurate requirements. *SES/Workbench*<sup>™</sup> is a multilevel design environment for system modeling and evaluation. These tools are being used or are planned for use on the following programs: ALFS, AX, CAINS, E-2C, GPWS, LAMPS Mk3 Blk II, MH-53, and T-45TS.

**Software Reuse  
Capability Will  
Enhance Systems  
Affordability Only  
If Matured**

- Software reuse becomes more important as system functionality moves from hardware to software.
- The infrastructure required for large scale software reuse does not exist today.
- Adoption of an operating systems interface promotes reuse.
- Object-Oriented Design is being developed to support reuse.

For more information on software reuse, refer to Volume 1, sections 9-2.3.7, and 9-3.8.

A potential major cost savings could be realized if a high percentage of software is reused. Software reuse is currently practiced in an *ad hoc* fashion, and there is no tangible commitment from government (or incentive to industry) to implement and apply reusable software.

One reason for the lack of motivation for reuse is the legal issues of responsibility and use. Legal issues such as certification, data rights, repository support, performance liabilities, and integration liabilities are all unknowns that need resolution before software developers can comfortably embrace software reuse.

For more information on Object-Oriented Design (OOD) refer to Volume 1, section 5-3.1.2.

Affordable software reuse is expected to become more and more achievable as the industry transitions to, and solves some of the problems associated with, Object-Oriented Design (OOD). This new design methodology is a major departure from the structured design methodology used to develop nearly all programs written to date. The major benefits of OOD are:

- (1) a high level of modularity,
- (2) an encapsulation mechanism to support information hiding,
- (3) data abstraction to support creation of objects that refer to a single basic data type and operation,
- (4) a classification system for these objects, and
- (5) an inheritance mechanism that provides reusability and extendibility.

These benefits are most desirable, however, the standardization and consistency that the industry achieves with this new design approach will determine its true benefit to both affordable software development and reuse. This transition will not be either quick nor easy because Object-Oriented Design (OOD), Object-Oriented Analysis (OOA), and Object-Oriented Programming (OOP) are difficult to mix with structured design, analysis, and programming. This effectively makes it an "all or nothing" corporate commitment adding both cost and risk for the designer.

One highly significant fallout of the evolving nature of OOP is the debate over software language standard selection between C++ and Ada. The basic constructs of C++ support a greater degree of inheritance and polymorphism than Ada, but at the expense of increased complexity and the real-time operation needed for many aircraft applications. Today, these problems are most pronounced when designing large, distributed real-time systems. Ada is attempting to accommodate these factors because DoD has a demanding requirement for these types of systems.<sup>2</sup>

While the infrastructure for large scale software reuse is not fully available today, both industry and the government are trying to hasten its evolution.

### Ada Should Remain DoD's Software Language Of Choice

Ada has been evolved to ensure methodical, well-documented software development, targeted for large complex systems. The Ada development process provides checks and balances needed for software program error detection and correction better than any other higher order language.

For more information on Ada as a Higher Order Language (HOL) refer to Volume 1, section 9-3.4.

- There is strong industry support for specifying Ada for custom applications, systems software, and any large scale avionics systems.
- Industry concurred that DoD's use of a standard language is needed to promote more efficient software development and reuse.
- Ada is appropriate for aircraft because it accommodates the large, complex systems found in aircraft.
- A significant segment of industry recommended that DoD alter policy regarding the use of other languages in order to optimize the opportunity to leverage COTS software products.

Industry believes that standardizing on a common software language promotes more efficient software development and reuse. Currently, the primary alternative to Ada under consideration by industry is C++. Figure 3 presents our teams analysis of the pros and cons for both languages. While this analysis suggests that Ada should be the language of choice today, DoD needs to continuously assess the technical trade-offs and requirements as Ada 9X, C++, and other higher order languages (HOL) continue to mature.

<sup>2</sup> Nielsen, Kjell. Object-Oriented Design with Ada . Maximizing Reusability for Real-Time Systems. New York, NY: Bantam Books, 1992.

<u>Ada</u>	<u>C++</u>
<ul style="list-style-type: none"> <li>• <u>ADVANTAGES:</u> <ul style="list-style-type: none"> <li>• Stable, tightly controlled</li> <li>• ANSI Standard</li> <li>• No dialects or versions</li> <li>• Clear, readable, supportable</li> <li>• Successful in realtime deterministic application</li> <li>• Cross compilers available</li> <li>• Ada 9X evolving</li> <li>• Good for large size builds (greater than 100K) with critical mission, high reliability requirements. Designed for maintainability.</li> </ul> </li> <li>• <u>DISADVANTAGES:</u> <ul style="list-style-type: none"> <li>• Lacks some features for OOD</li> <li>• Commercial industry use is not widespread</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• <u>ADVANTAGES:</u> <ul style="list-style-type: none"> <li>• Accommodates Object Orient development</li> <li>• Strong commercial use C → C++</li> <li>• Good for small size, not life critical, w/low maintenance planned</li> <li>• Language for an "artist"</li> </ul> </li> <li>• <u>DISADVANTAGES:</u> <ul style="list-style-type: none"> <li>• Dynamic</li> <li>• 18 versions or dialects</li> <li>• No formal industry standard</li> <li>• Harder to maintain</li> <li>• No cross compilers available</li> <li>• Not proven in large embedded systems</li> <li>• Not proven for real-time, deterministic</li> <li>• Additional training required for Aerospace Industry programmers</li> </ul> </li> </ul>

Figure 3: Advantages and Disadvantages of Ada and C++.

A potential problem for long term military use of Ada is that little commercial software is written in Ada. While the Ada development process provides the attributes listed above, as well as important checks and balances needed for software program error detection and correction, it is initially more costly to write than many of the higher order languages used commercial applications. A balance must be struck between the opportunities afforded by commercial software (especially in the realm of testing and software development tools) and the need to maintain both software and software processes not supported by commercial industry.

---

**Joint Service  
Acquisition Makes  
Sense**

- The Joint Integrated Avionics Working Group (JIAWG) baseline provides a framework for a modular integrated avionics architecture
- The Navy-led Standard Hardware and Reliability Program (SHARP) is leading in the application of advanced technologies to standardize modular packaging for the military

In this period of lean budgets it is important that the Navy and the other services share avionics standardization approaches in order to broaden our applications base and to make use of the products of research and standards developed by all of the services.

Programs such as JIAWG, NGCR and SHARP have been created to establish, in their own way and by different approaches, architectural standardization. It is important that the avionics community utilize these programs as appropriate to contribute to avionics standardization. For instance, JIAWG standards provide near-term solutions and should be considered and used where applicable to meet requirements.

In an effort to provide a common focus to tri-service avionics standardization efforts the Joint Aeronautical Commanders Group (JACG) Engineering Board is now forming an Avionics Engineering Sub-board (AESB). The Army, Air Force, Navy, Defense Logistics Agency, National Aeronautics and Space Administration, and Federal Aviation Administration, will be co-equal members in this effort to create more compelling opportunities for avionics standardization. The precepts of the AESB will be to:

- (1) Standardize on best engineering practices.
- (2) Facilitate the joint specification development process.
- (3) Interface with industry standards bodies.
- (4) Establish common standard/specification verification/qualification and configuration control methodology.
- (5) Implementation using Systems Engineering.

---

## Technology Insertion Can Be Made Affordable

- Effective technology insertion can provide major life-cycle cost benefits.
- An open systems standards approach makes technology insertion easier.
- Industry wants to remove obstacles that limit innovation, creativity and insertion of newer technology.
- DoD can best achieve technology insertion through standardized architectural requirements and use of COTS/NDI.

The rate of new generation electronic technology advancement is 18 to 24 months. Technologies that are presently in development, but not yet ready for exploitation, are considered moderate to high risk and require careful planning to ensure the technology will be ready for insertion.

Emerging technology is considered very high risk, but can be tracked to determine when it will be mature enough for insertion. Open systems will facilitate technology insertion since components and modules can be changed almost at will as long as the interfaces remain compliant with the standards. The use of an open systems standards-based approach allows increased numbers of technology sources, the ability to upgrade without redesign and provide flexibility in system design through integration capability and utilization of P<sup>3</sup>I/evolutionary acquisition. Systems that adopt popular commercial products as components may be able to also leverage their continuous enhancement. This allows the military to avoid the high risks associated with productionizing new technologies, by using state-of-the-art application ready products developed at the commercial consumer's expense.

---

## Conclusions

While it is impossible to predict what avionics architectures will look like 20 to 30 years in the future, it is possible to evaluate present-day technology trends and select those technologies that can have the greatest potential impact on future avionics.

Although no universal architecture will satisfy all of the requirements of naval avionics systems, a standards-based approach to systems development will establish a basis for consistency among platforms that provides for cost-effective technology insertion and performance upgrade. Endorsing an open system standards-based approach allows the Navy to move into the mainstream of modern technology, and provides a broader industrial base to support our systems applications.

The Navy should limit the number of candidate standards to just a few, so that diverse requirements can be met, commonality achieved and cost savings realized. Changes should be made so that policy regarding architectural choices encourages the use of COTS and NDI hardware and software, and encourages Navy participation in standards development groups. Changes in the procurement and acquisition process (DAR and FAR regulations) to eliminate stumbling blocks to an effective systems engineering integrated product team approach must be addressed.

The Navy should encourage the use of a commercially supported, open systems standards based approach under a disciplined, systems engineering process.

There should be a limited number of standards selected from various sources. The criteria to determine the preferred standards should be part of the systems engineering methodology.

The recommended order of precedence of engineering standards for application to new and existing systems, within the systems engineering process, is:

- a) Open Systems Standards controlled by non-government standards bodies
- b) Widely-used standards under Government control
- c) Standards controlled by special purpose working groups
- d) Proprietary standards for unique interfaces

---

# Recommendations

Based on the findings of this study, the following broad-based recommendations are made.

---

## Use an Open Systems Standards-Based Approach to Avionics Systems Development

It is recommended that the Navy begin the transition from multiple avionics design architectures that are unique to military aircraft to an approach that capitalizes on open system opportunities. It was determined that no universal architecture will satisfy the diversity of requirements for Navy Avionics systems. But just as a single architecture can not be applied efficiently, a limit must be placed on the number of architectural choices to achieve reasonable standardization and commonality. An open systems-based approach provides for cost-effective development through the use of widely-accepted standards with broad industry support. Therefore, transition to an open systems standards-based approach is recommended for future avionics systems development. The use of open standards facilitates the application of COTS and NDI products which will allow NAVAIR to leverage trends in the commercial market in the future.

To utilize open system standards fully, it is recommended that NAVAIR be an active participant in the standards organizations that are responsible for applicable open system standards. This activity will become familiar with the detailed technical elements of these standards ensure that naval avionics needs are considered and incorporated as the standards are developed.

In addition, it is recommended that the Navy establish a limited number of architectural choices through a managed program of technology tracking, standards selection, risk assessment and technical trade-off studies.

---

## Leverage Commercial Technology and Products

It is recommended that the Navy continue to promote the use of COTS technology and products in military systems as appropriate. Towards this end, the Navy should investigate common areas between commercial and military avionics systems such as communications, navigation, and displays sensors, to name a few, in which COTS might be readily applied to realize immediate cost savings.

In addition, the Navy should adopt a strategy for managing optimal COTS use. This strategy will allow commercial software tools and development systems to be used, where appropriate, at a minimal cost to the government. It will also ensure that COTS is not the *de facto* choice, but that conscious selections are made during the systems engineering process.

---

## Support Joint Service Programs

It is recommended that the Navy aggressively and actively participate in joint service programs. Coordinated DoD efforts will allow more efficient management of scarce DoD resources and focus industry's efforts on consolidated military requirements. Joint service programs allow the Navy to take advantage of other service efforts and to share the cost of developments. Additional cost advantages can be derived from larger production buys, shared maintenance facilities, and common support systems. Further, the participating services can use each others' research in defining and developing their systems. The goal for these efforts should be equipment and software commonality between and among platforms, otherwise the advantages are significantly reduced.

It is recommended that Navy continue to develop the tri-service Avionics Engineering Sub-board (AESB) of the Joint Aeronautical Commanders Group (JACG) Engineering Board. Joint service cooperation and resource contributions will be required to increase weapon systems affordability by developing a higher level of commonality in avionics systems and subsystems. This commonality may be achieved through the management of avionics architectures, processes, standards, and specifications.

It is recommended that the Navy continue its support to Joint and Tri-Service efforts such as JIAWG, NGCR, MASA and SHARP. It is important that the Navy, together with the other Services, share avionics standardization approaches to broaden our applications base and utilize the research and standards developed by all the services when applicable.

---

## Invest In A Software Development Infrastructure Now To Establish Long Term Software Affordability

It is recommended that more corporate attention focus on software affordability. Applying systems engineering principles to software development and its management is fundamental. Software cost management is also closely linked to effective requirements analysis since stabilizing requirements early in the process can provide extensive cost savings. For example, process analysis using software metrics and corporate commitment to correct identified deficiencies will pay dividends in achieving quality, cost-effective software. It is recommended that efforts should be taken to track the viability of commercial software languages for opportunities. It is also recommended that software reuse be encouraged and supported as a cooperative effort between the government and industry.

---

**Ensure that a  
Rigorous Systems  
Engineering  
Approach is Used**

It is recommended that the Navy establish a formal system engineering approach to guide architectural choice, engineering standards selection, and implementation and integration of cost-effective technological solutions to naval avionics problems. Systems Engineering is regarded with such special significance by the review team that a complete volume of this report, Volume 2, has been devoted to the systems engineering process. The systems engineering approach will synthesize effective avionics systems solutions utilizing COTS where appropriate, while satisfying all constraints of cost, schedule and performance. The application of COTS and NDI products, which includes joint service program products, into avionics systems provides more affordable systems. However, leveraging commercially-based and existing military products places a significant burden on systems engineering and must be managed.

To assist with both the management and implementation of systems engineering, an investment will be needed in modeling and development software tools. While we should leverage the tools used by industry, investment in state of the art tools will greatly increase productivity and cost savings if we are selective and properly train for their use.

A systems engineering approach is already in place in the naval avionics community. It is recommended that the systems engineering process of MIL-STD- 499B be tailored for avionics. In addition, it is recommended that the benefits of closer integration with industry throughout the systems engineering process be examined, especially for large scale development or upgrade programs. It is further recommended that to manage investment of high-cost technology integration, a systematic method of tracking technology and standardization be established.

DEPARTMENT OF THE NAVY  
NAVAL AIR SYSTEMS COMMAND

ADVANCED AVIONICS ARCHITECTURE  
AND TECHNOLOGY REVIEW  
FINAL REPORT

**\*\* VOLUME 1: AVIONICS TECHNOLOGY \*\***

6 August 1993

NAVAL AIR SYSTEM COMMAND  
SPACE AND NAVAL WARFARE SYSTEMS COMMAND  
NAVAL RESEARCH LABORATORY  
NAVAL AIR WARFARE CENTER, AIRCRAFT DIVISION-  
INDIANAPOLIS & WARMINSTER  
NAVAL AIR WARFARE CENTER, WEAPONS DIVISION-CHINA LAKE  
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY  
NATIONAL INSTITUTE FOR STANDARDS AND TECHNOLOGY

**This page intentionally left blank.**

# Table Of Contents

- 1-0 A QUESTION OF SCOPE: WHAT IS AVIONICS?..... 1
- 2-0 WHY STUDY SYSTEMS ARCHITECTURE?..... 2
- 3-0 SYSTEMS AND SYSTEMS ARCHITECTURE..... 3
  - 3-1 Systems ..... 3
  - 3-2 Standards ..... 3
  - 3-3 Architecture ..... 4
  - 3-4 Types of Avionics Architectures ..... 5
    - 3-4.1 Independent Architecture ..... 5
    - 3-4.2 Federated Architectures ..... 5
    - 3-4.3 Integrated Architectures ..... 6
    - 3-4.4 Distributed Architectures ..... 6
  - 3-5 Architecture Standardization ..... 7
    - 3-5.1 Closed Systems Architectures ..... 7
    - 3-5.2 Open Systems Architectures ..... 7
    - 3-5.3 U.S. Government Standards and Openness ..... 8
    - 3-5.4 Advantages and Disadvantages of Open Systems Architectures ..... 8
  - 3-6 Buses Provide Architecture Communications ..... 11
    - 3-6.1 Box-to-Box Interconnect Buses ..... 11
    - 3-6.2 Peripherals Interface Buses ..... 11
    - 3-6.3 Backplane Buses ..... 12
    - 3-6.4 Sub-Buses Within The Backplane Bus ..... 12
- 4-0 MAJOR AVIONICS ARCHITECTURES ..... 13
  - 4-1 Joint Integrated Avionics Working Group (JIAWG)..... 13
    - 4-1.1 Technology Base/Related Programs ..... 13
    - 4-1.2 Core Avionics ..... 15
    - 4-1.3 Mission Avionics ..... 15
    - 4-1.4 Supportability ..... 15
    - 4-1.5 Software ..... 16
    - 4-1.6 JIAWG Standards & Specification Descriptions for Core Avionics ..... 16
      - 4-1.6.1 Architecture ..... 16
      - 4-1.6.2 Processors ..... 17
      - 4-1.6.3 Data Buses ..... 18
      - 4-1.6.4 Backplane ..... 18
      - 4-1.6.5 Packaging ..... 19
      - 4-1.6.6 Power supplies ..... 20
    - 4-1.7 JIAWG Standards & Specification Descriptions for Mission Avionics ..... 20

Table of Contents (continued)

4-1.8	JIAWG Standards & Specification Descriptions for Supportability .....	20
4-1.9	JIAWG Standards & Specification Descriptions for Software .....	21
4-2	Next Generation Computer Resources (NGCR) .....	22
4-2.1	NGCR Standards Descriptions .....	23
4-2.2	Multiprocessor Interconnects .....	23
4-2.3	The Baseline Backplane and the NGCR Modular Open Architecture .....	23
4-2.4	The High Speed Data Transfer Network (HSDTN) .....	24
4-2.5	High Performance Backplane (HPB) Standard .....	25
4-2.6	NGCR Modular Open Architecture Features .....	25
4-2.7	Standards .....	28
4-2.7.1	High Performance Network Standard .....	28
4-2.7.2	Software Interfaces .....	28
4-2.7.3	Operating System Interface (OSIF) Standard .....	28
4-2.7.4	Project Support Environment (PSE) Standard .....	28
4-2.7.5	Data Base Management System (DBMS) Interface Standard .....	29
4-2.7.6	Graphics Language Interface Standard .....	29
4-2.7.7	NGCR Standards Applied to an Avionics System .....	29
4-2.8	Multi System Interconnects .....	30
4-2.9	Common Operating Environment (COE) .....	30
4-3	VMEbus .....	30
4-4	Commercially Supported Avionics Architecture Efforts .....	31
4-4.1	AEEC and ARINC .....	31
5-0	AVIONICS TECHNOLOGY DEVELOPMENT & REVIEW EFFORTS .....	35
5-1	U.S. Air Force PAVE PILLAR and PAVE PACE Programs .....	35
5-1.1	PAVE PILLAR .....	35
5-1.2	PAVE PACE .....	35
5-2	Integrated Communications, Navigation, Identification, Avionics (ICNIA) .....	35
5-3	Integrated Electronic Warfare System (INEWS) .....	37
5-4	Advanced Avionics Subsystems and Technology (AAS&T) .....	38
5-4.1	Documented OPNAV Requirements .....	38
5-4.2	Description of Project .....	38
5-4.3	Rationale for The Project .....	39
5-4.4	Tactical Utility .....	39
5-4.5	Resource Savings - Manpower, Logistics Support, Etc. ....	40
5-4.6	Countermeasure Resistance .....	40
5-4.7	Major Tasks and Subsystems .....	40

## Table of Contents (continued)

5-4.7.1	Shared Aperture and Multi Function Systems.....	40
5-4.7.2	Digital Technologies.....	41
5-4.7.3	Avionics and Photonics.....	41
5-4.7.4	Avionics Packaging.....	42
5-4.7.5	Situation Assessment and Awareness.....	42
5-4.8	Key Technologies.....	42
5-5	Integrated Avionics Development Program (IADP).....	42
5-6	Ultra Reliable Digital Avionics (URDA).....	43
6-0	SYSTEM BUSES, NETWORKS, AND INTERCONNECTIONS.....	44
6-1	Interconnect Buses.....	44
6-1.1	High Speed Data Bus (HSDB).....	44
6-1.2	MIL-STD-1553 and MIL-STD-1773 Serial Linear Control Buses.....	45
6-1.3	Fiber Distributed Data Interface (FDDI) / SAFENET.....	46
6-1.3.1	FDDI in Avionics.....	46
6-1.4	Asynchronous Transfer Mode (ATM) Local Area Network (LAN).....	47
6-1.5	Fibre Channel.....	47
6-1.6	Other Network Schemes.....	48
6-1.7	Electronics Industries Association (EIA).....	48
6-2	Backplane Buses and Networks.....	48
6-2.1	Pi Bus.....	49
6-2.2	Test & Maintenance (TM) Bus.....	50
6-2.3	IEEE-896 Futurebus+.....	50
6-2.4	IEEE-1394 High Speed Serial Bus.....	51
6-2.5	VME Bus, extension to VME64 (Commercial VME Products, IEEE).....	51
6-2.6	IEEE-1596 Scalable Coherent Interface (SCI).....	52
7-0	KEY TECHNOLOGIES.....	54
7-1	Generic Technologies.....	55
7-1.1	Computerized Engineering Tools.....	55
7-1.2	Emerging Technologies.....	56
7-2	Microelectronics Technologies.....	56
7-2.1	Trends in High Density Microelectronics.....	56
7-2.1.1	What Does the Microelectronics Trend Mean?.....	58
7-2.1.2	What About The Threats?.....	60
7-2.2	Random Access Memory Chips.....	60
7-2.3	Gate Array Technology.....	62
7-2.4	Analog-to-Digital Converters.....	62
7-3	Microwave/Millimeter-wave Monolithic Integrated Circuit (MIMIC).....	63
7-3.1	MIMIC Phase 1 Accomplishments.....	64
7-3.2	MIMIC Phase 2 Directions.....	66

Table of Contents (continued)

7-3.3	Tri-Service MIMIC Reliability And Radiation Effects Program .....	68
7-3.4	MMIC Section Summary .....	69
7-3.5	Beyond MMIC .....	70
7-4	Advanced Processor Technology .....	70
7-4.1	Microprocessor Technology .....	70
7-4.2	Examples of Commercial RISC, CISC and DSP Chips .....	71
7-4.2.1	RISC .....	71
7-4.2.2	CISC .....	71
7-4.2.3	DSP .....	72
7-4.3	Technology Trends and Time Lines .....	72
7-4.4	Key Developments For Advanced Processing .....	73
7-4.4.1	Hewlett-Packard Processor .....	73
7-4.4.2	Aladdin .....	74
7-4.4.3	Intel's Evolutionary Processor Growth .....	75
7-4.4.4	Alpha .....	76
7-4.4.5	Touchstone (DARPA) .....	77
7-5	Commercial Trends in Processor Architecture .....	79
7-5.1	Microprocessor Chip Progress .....	80
8-0	<b>ADVANCED PACKAGING TECHNOLOGIES AND ADVANCED ELECTRONIC PACKAGING .....</b>	<b>81</b>
8-1	Avionics Environments and Packaging .....	81
8-2	Standardization and Packaging .....	82
8-3	Commercial Avionics Package Standards .....	82
8-4	Standard Hardware Acquisition and Reliability Program (SHARP) .....	83
8-5	Hardware Interconnects and Connectors .....	84
8-5.1	Computer Backplanes .....	84
8-5.2	Electrical and Digital Connections .....	85
8-5.3	Technology Trends .....	86
8-5.4	Optical Interconnects .....	86
8-5.5	Radio Frequency (RF) Elements .....	88
8-6	Thermal Management .....	88
8-6.1	Component and Module Level .....	89
8-6.2	Standard Computer Modules .....	89
8-6.3	Die and Component Level .....	89
8-6.4	Circuit Card and Module Level .....	90
8-6.5	Common Cooling Methods .....	90
8-6.6	Other Cooling Schemes .....	91
8-6.7	New Cooling Technologies .....	91
8-6.8	Systems Level (Enclosure/Integrated Rack) .....	92
8-7	Structural .....	92
8-8	Advanced Power Systems .....	93
8-8.1	Power Supplies and Power Distribution .....	93
8-8.2	Modular Distributed Power Systems Architecture .....	93

Table of Contents (continued)

8-8.3	On-module Distributed Architecture .....	94
8-8.4	On-Module vs Modular .....	95
8-8.5	Technology Required for On-module Power Converters .....	96
8-9	DC Power Supplies .....	96
9-0	<b>EFFECTIVE SOFTWARE DEVELOPMENT AND MANAGEMENT .....</b>	<b>98</b>
9-1	Real-time Computer Resources Development Dependency on the Systems Engineering Process .....	99
9-2	Computer Resources Open Systems Architecture .....	100
9-2.1	Understanding Open Systems .....	100
9-2.2	Hardware Architectures .....	101
9-2.3	Software Architectures .....	101
9-2.3.1	Software Control/Partitioning .....	102
9-2.3.2	Flow of Data .....	102
9-2.3.3	Timing and Throughput .....	102
9-2.3.4	Interfacing Layering and Protocol Standards .....	102
9-2.3.5	Software Security and Safety .....	103
9-2.3.6	Software Portability .....	103
9-2.3.7	Reuse .....	103
9-2.3.8	Verification and Validation Test .....	103
9-2.3.9	Technology Insertion .....	103
9-2.3.10	Growth .....	103
9-2.3.11	Embedded Training .....	104
9-3	Computer Software Development .....	104
9-3.1	Software Design Methodologies .....	104
9-3.1.1	Structured Design .....	104
9-3.1.2	Object Oriented Design .....	104
9-3.2	Software Standards and Practices .....	106
9-3.3	Commercial-Off-The-Shelf(COTS)/Non- Development Items (NDI) .....	107
9-3.4	Languages .....	108
9-3.5	Software Engineering Environments/Software Tools .....	110
9-3.6	Operating Systems .....	110
9-3.7	Testing .....	111
9-3.8	Software Reusability .....	112
9-3.9	Rapid Prototyping .....	114
9-4	Computer Resources Management .....	114
9-4.1	Process .....	114
9-4.2	Metrics .....	116
9-4.3	Software Experience Database .....	117
9-5	Software Science and Technology Thrusts .....	117
10-0	<b>REFERENCES .....</b>	<b>118</b>

**This page intentionally left blank.**

# Advanced Avionics Architecture and Technology Review

## Final Report

>> Volume 1 - Avionics Technology <<

### 1-0 A QUESTION OF SCOPE: WHAT IS AVIONICS?

Avionics is commonly understood to be electronics used aboard aircraft, i.e., as an aircraft subsystem. But for purposes of this study the range of the common definition was more focused. For example, because of the traditional way of partitioning aircraft subsystems, the flight control computers and related electronics are normally under the cognizance of the aeronautical engineering divisions, even though they fit the broad definition of "avionics." For purposes of this report, therefore, avionics means all on-board aircraft electronics except the flight and engine control electronics. Subdivisions of the avionics category include flight avionics (e.g., common with all aircraft, i.e., navigation, communications), tactical sensors (radar and electro-optical), and computer resources. This subdivision is somewhat arbitrary, but nonetheless useful.

One of the purposes of this study is to identify areas and means of cost savings with respect to avionics. When the different sub categories of avionics systems are examined, it becomes apparent that the two main cost drivers are tactical sensors (e.g., radar and electro-optics) and computer resources. The sensors cost factors are considerably higher than the computer resources cost factors (even though both are large), so by the normal rules of Pareto analysis, a discipline used extensively by practitioners of Total Quality Leadership, one might assume that the Advanced Avionics Architecture and Technology Review Team(AAART) would put the main emphasis on tactical sensors rather than computer resources. However, after close analysis it was seen that much of the cost burden of tactical sensors is due to computer hardware and software embedded within these sensor subsystems, so the two main problem areas overlap. Therefore, paying close attention to computer resources problems deals with both major cost drivers in an effective way. In addition, among all Navy aircraft there is a much broader based common area of technology in the computer resources (both mission computers and embedded subsystem computers) than in the non-computer areas of sensor technology. There is also the fact that computer technology is now used extensively in non sensor avionics subsystems. It was determined that a major contribution to Navy avionics could be made by addressing the study primarily to the computer resources, but with some attention paid to non computer tactical sensor technology. Avionics that do not fit into these categories are important, but are not discussed herein because their overall effect on possible cost reductions is low compared with computer and sensor technologies.

## 2-0 WHY STUDY SYSTEMS ARCHITECTURE?

This study focused on avionics systems architectures because the fundamental architecture underlying any system predetermines initial development costs, future flexibility to meet changing requirements, and the ease with which new technology can be incorporated into the system. Architecture also determines both the degree to which the Navy can leverage commercial research and development funding, and the level at which the products of commercial R&D is useful to the Navy. Certain architectures are inherently more amenable to the use of Non-Developmental Items (NDI) and commercial off the shelf (COTS) hardware and software than others, so it becomes necessary to understand the implications of architectures on our future capabilities.

Major architectural decisions tend to be made early in the design of a system, and those decisions — once firm — considerably reduce the ability to change course in the future. In the present state of affairs, for example, if the wrong architecture is selected very early in a program, then the ability to effect fundamental design changes is reduced, even as early as the Preliminary Design Review (PDR). As a result, new technologies and commercial R&D cannot easily be exploited for Navy systems if the wrong decisions are made early in the process. Nor can new or evolving threats be met without either very costly and time consuming redesign of an old system, or procurement of an entirely new system.

Selecting viable architectures, and acquiring sufficient useful information about them, substantially improves our ability to respond to a dynamic environment, even in the face of high complexity. The greatest leverage on a system's life cycle is found in the early architectural decisions, and is maintained by keeping as many options as possible open to the latest possible time.

This report describes many of the technologies available to designers of Naval air weapons systems. Understanding these technologies aids the systems architect in creating designs that allow the exploitation of emerging technologies even in an environment of reduced funding.

## 3-0 SYSTEMS AND SYSTEMS ARCHITECTURE

Modern weapons systems, of which avionics represents a major sub category, are very complex compared to systems of just a decade or two ago. As a result, design of systems has become much more complex than before. It is well recognized that the problems of designing very complex systems grows much faster than the complexity of the system itself. Therefore, small changes in system complexity can create much larger changes in the problems associated with designing and fielding the system. Before the problems of designing and managing complex systems can be addressed, some definitions must be understood.

### 3-1 Systems

According to *Webster's New Collegiate Dictionary*, a system is "...a regularly interacting or interdependent group ... forming a unified whole," or "...[a body of parts] considered as a functional whole [in a] harmonious arrangement or pattern [emphasis added]." Reichtin (1991) goes further and points out that a system possesses what are called *emergent properties*, i.e., properties or attributes that are not evident by examining the component elements apart from the system. Checkland (1981) asserts that the very concept of 'system' deals with these emergent properties rather than the properties of the components. The properties of a human body, for example, cannot be fully ascertained from understanding the properties of the various parts of the body. When a system is broken apart, emergent properties disappear. For example, cutting an adult human body into two equal pieces does not result in a pair of twin children, but rather a dead adult. The naval aircraft and its avionics suite form a system that cannot be understood apart from the aggregate. At lower levels, the avionics suite forms a system, which is in turn made of subsystems, all of which interact in such a way as to form a whole that is greater than the sum of its parts (i.e., synergism). If an aggregation fails the tests of interconnectedness and synergy, then it is not a system. But if it passes these tests, then it is a system and must be dealt with in the whole, as a system, and not simply as a collection of parts. In the past, we often dealt with systems isolated from their larger context...a luxury that we can no longer afford. As complexity grows, and resources dwindle, the old way of doing things becomes untenable.

### 3-2 Standards

A term used a lot in discussions of avionics architectures is *standards*. A principal finding of the AART was that the most viable methodology for acquisition of naval avionics is a well-defined standards-based approach modulated by a rigorous systems engineering process. In the general sense, a standard is a written "...means of determining what a thing should be..." that is regularly and widely available, accepted and used. Related words include *criterion*, *gauge*, *yardstick* and *touchstone*. A standard is a written document that can be used to compare a thing against. In the more specific sense, a standard is a document produced by an authority and widely promulgated. Private standards (company proprietary), government standards, and industry standards can qualify, but in the limited sense used in this report it is widely used industry consensus standards that are meant. These standards are put together by groups that contain representation from a large segment of the potential user group. Such standards are typically given a number and quoted as such (e.g., IEEE-896, ANSI 91, ISO-9000, etc.)

### 3-3 Architecture

The concept of *architecture*, as it applies to an avionics system, is related to the same concept in building construction. The word architecture incorporates the complete set of factors that contribute to making the avionics suite into a system that works as a unified functional whole to accomplish the mission of the aircraft.

Avionics architecture, like its building counterpart, is hierarchical (i.e., layered) and multifaceted. A glimpse of some of these aspects of architecture can be seen in a list of typical attributes of an avionics architecture:

#### Communications:

- Information flow and communications pathways (e.g., "wiring diagram")
- Backplane bus or buses used
- Interfaces within the system
- Interfaces to the "outside world"

#### Computations:

- Speed, throughput, latency, and redundancy of computer processors and data communications
- Processors used in various functions (common or multiple types)
- Type, location, size and arrangement of memory devices
- Software used

#### Organization:

- Type and arrangement of sensors
- Type and arrangement of mission avionics
- Style of arrangement of major components (e.g., federated, distributed, centralized, etc.)
- Number and location of computer processors
- Physical arrangement of hardware in the system

The concept of "architecture" incorporates the complete range of factors that make the system perform as a functional whole. Avionics architectures describe the form, structure and interrelationships between the avionics system and the aircraft, and amongst the elements of the avionics system itself. The selection of backplane bus within the computers, the types of Electro-Optic (EO) and radar sensors used, and the switches used on the control panel are all part of "architecture." Because the task of architecture definition is hierarchical however, different aspects of the architecture are apportioned to different tasks in the process of creating the system. For example, the selection of switches for the control panel is properly delegated to a detail designer, and not the

system engineering architect. The partitioning of functions, with due regard to which layer is appropriate, is a systems engineering task that must be performed very early in the process.

### 3-4 Types of Avionics Architectures

In this report, the terms "architecture" is used mostly for the higher levels of the hierarchy tiers where functions are partitioned and the structure of their respective interconnections can be dealt with most effectively.

Several different generic types of avionics architecture have evolved over the years, and these are described in the sections below. The categories overlap, and have fuzzy boundaries between them, and should be taken as general guidelines only when the various types of architecture are discussed.

#### 3-4.1 Independent Architecture

The independent architecture is the oldest avionics architecture, and dates from the late 1920s when radios were first installed on aircraft. In this type of system, all subsystems are completely independent except for sharing a common power source and common environmental control system. A radar or radio, for example, could be installed aboard an early World War II era aircraft with only minimal concern for other avionics devices on board. These systems were common into the early 1960s, but are now regarded as obsolete.

#### 3-4.2 Federated Architectures

Independent architectures were replaced with federated architectures starting in the late 1940s, but accelerating in 1960s. The key enabling factor in the rapid ascendancy of the federated architectures in the 1960s was the rise of computers and data communications technologies. In the federated system, overall control and a certain amount of functionality of each avionics subsystem, is delegated to a central mission computer. As computer capability increased, so did the percentage of total functionality apportioned to software. Subsystems (e.g., radar, navigation, communications, etc.) talk to each other over a standardized data bus (e.g., MIL-STD-1553) using a common data format. The data bus might be local (within a subsystem), dedicated (between subsystems) or global (to all — or a major block of — aircraft subsystems).

An advantage of the federated system, over the older independent system, is that it permits better interaction of several avionics subsystems to form a synergistic whole. For example, radar and navigation systems can interact with each other to provide updates to positional knowledge in the navigational system and better ground maps for the radar. Both systems are made better because they communicate with each other through the software of the central mission computer. Similarly, new tactical advantages emerge when radar, E-O and Electronic Warfare (EW) sensors interact in a coherent manner.

Federated architecture does not imply that the subsystems do not have any computer processing of their own, but rather that central coordination and control is delegated to the mission computer. Forward looking infrared (FLIR) E-O and radar subsystems, for example, might each possess very powerful signal processing computers and data computers within the subsystem, but control, display and interactions with other systems is handled by the central mission computer.

Between the 1960s and the 1980s aircraft design became firmly entrenched in the federated model, and the F/A-18 provides a very good example of an advanced federated avionics system.

### 3-4.3 Integrated Architectures

An integrated architecture is one in which all, or nearly all, of the electronics elements of the avionics system are packaged in standard modules, and installed in several (but very few) co-located common racks. Because the computer resources of a modern avionics suite are so numerous, the term "integrated" could also mean a system in which the computing elements, and supporting modules, are co-located in a common rack. The Joint Integrated Avionics Working Group (JIAWG) architecture is an example of this concept.

In other cases, elements such as radio frequency (RF) modules, for use in radar, electronic warfare, communications and navigation systems could also be integrated into a single rack structure. The Integrated Communications, Navigation, Identification Avionics (ICNIA) system and the INtegrated Electronic Warfare System (INEWS) are examples of this class of integrated system. Problems of interface connections and electromagnetic interference (EMI) within the racks must still be solved, however.

### 3-4.4 Distributed Architectures

A distributed architecture resembles an integrated architecture in that elements of the system are co-located in racks or cabinets. It also resembles a federated system in that a considerable amount of digital signal processing is done in the subsystems. In a distributed system, major elements of the digital signals processing are located at the sensor or other subsystem site. For example, a radar system may have substantial signal preprocessing capability, located with the radar receiver, at or near the antenna site. The main signals processing function is performed in the central computer, as in the integrated system, but preprocessing eases the chore. Such systems have the potential for off-loading traffic on aircraft data buses, and reducing the workload of the central processors. In addition, special requirements of a system can be locally accommodated without being imposed on the entire system.

Some advocates assert that the distributed architecture is more robust than the integrated architecture because combat damage to a single site cannot incapacitate all systems. Others believe that proper design, with built-in redundancy, can overcome this problem with integrated architectures. It is not clear that the combat robustness of distributed systems is better than that of integrated systems, and the arguments of the advocates are as yet unconvincing.

A system can still be regarded as a distributed architecture if it uses common RF signal and wave form sources located at a central site. At some level, however, the boundaries between a sophisticated federated architecture, the distributed architecture and the integrated architecture begin to blur. The location of those boundaries, and the degree to which a system is either distributed or integrated, is the proper subject of rigorous systems engineering analysis, and may vary with the overall system design, the inherited legacy of earlier subsystems in the same system, available technology and cost constraints.

### 3-5 Architecture Standardization

The rising complexity and cost of avionics products has increased the need for 1) the systems developer to leverage the work of others where ever possible, and 2) the product vendor to have access to sufficient market share to amortize development and production design cost over an acceptable number of units. Consequently, an important issue that must be considered when designing a new or substantially modified avionics system or subsystem is whether to make it a *closed system architecture* (CSA) or an *open system architecture* (OSA). There is a lot of discussion in both commercial and military sectors over open systems architectures versus closed systems architectures. AART believes that the OSA approach should be considered first for most applications, but not at the automatic exclusion of CSA approaches if the true requirements are supported only in a CSA design. In the sections below these terms are discussed in detail.

#### 3-5.1 Closed Systems Architectures

The term *closed system architecture* (CSA) merely places a label on an older concept that did not need a label until open systems became popular. A CSA implies that the standards and specifications for the system are not open for use by anyone who wants to use them. Closed systems tend to be made to proprietary standards that are not in the public domain, so as a result the government does not obtain the information needed for complete support. For example, a computer backplane bus, and the software operating system used by the computer, may be the private property of the contractor supplying the product. While one can argue that Government contracts impute all rights to the Government, that issue is frequently in contention, and the Government frequently loses the battle. Thus, the CSA limits the ability of the Government to use any source other than the original contractor.

It is possible to optimize a "point solution" closed system design for the intended purpose. While that advantage is sometimes lost in open systems designs. No regard need be paid to industry-wide consensus standards committees for needed changes to a closed system. On the other hand, that freedom is purchased at the cost of effectively locking in the original contractor, and history shows that approach is very costly. Optimized point solutions are usually difficult and expensive to change.

#### 3-5.2 Open Systems Architectures

An *open system* is one that is sufficiently defined to provide for expansion, upgrading or functional reconfiguration through the incorporation of replaceable modular elements. The concept applies to both hardware and software. An example of an open system is the desktop computer in which the hardware and software can be configured as a word processor, a data or signals acquisition system, or a graphics processor depending on the plug-in cards and software programs installed.

An open system must have all aspects of the system interfaces so well defined, preferably in public domain standards, that independent designers of subsystems or modules can do their work without close coordination with each other. Under the ideal situation, a product can be installed in an open system with only minimal integration. The integration normally required when integrating plug-in boards into a desktop computer, for example, typically involves redirecting conflicting communication port assignments, interrupt line assignments, and the memory locations sought by on-board memory. As

long as one does not attempt to install two like-function boards into the same system, the effort is ostensibly trouble free.

The standards and specifications governing the open system must be non proprietary, widely used, and well recognized throughout industry. In most cases, an open system will be compatible with one or more consensus standards promulgated by industry-wide organizations such as the Institute of Electrical and Electronic Engineers (IEEE), the Society of Automotive Engineers (SAE), American National Standards Institute (ANSI), the International Electro-technical Commission (IEC) and the International Standards Organization (ISO). Key elements defining open systems standards include a formal control mechanism involving due deliberation of interested parties for changes, and an effective configuration management system.

Some people maintain that an "open" standard must be in the public domain (i.e., non proprietary) so that it can be used without either permission or payment of license fees. In general, this position seems reasonable because proprietary standards may give the proprietor both unacceptable control over a design, and an unfair competitive business advantage when competing with non proprietors for Navy contracts. But the ISO regards a standard as "open" even if license fees are required, provided that the fees are moderate and licensing is not discriminatory. Thus, degrees of openness may be encountered, especially in regards to international commercial consensus standards.

### **3-5.3 U.S. Government Standards and Openness**

Most U.S. Government standards and specifications are not usually open in the sense of being in widespread use, but that is not universally the case. If a standard is relatively stable, configuration controlled, and has achieved widespread recognition, then it may be construed as being open. For example, MIL-STD-1553 defines a data bus for use between systems or boxes in a military avionics system. It is not open in the normal sense of the word, because its use is relatively narrow (military aircraft, with only a few commercial applications). It becomes quasi-open, even though not an industry consensus standard, because a large number of commercial chip providers make 1553 transceiver devices for incorporation into other products. Thus, any user, military or not, can use the MIL-STD-1553 bus without needing permission from a proprietor.

Although the Joint Integrated Avionics Working Group (JIAWG) standards can evolve into an open, or at least quasi-open standards, at present its general unavailability and lack of widespread use does not suggest openness according to the above definition. It is claimed that JIAWG will evolve to an open standard by 1996.

### **3-5.4 Advantages and Disadvantages of Open Systems Architectures**

The use of an open systems architecture (OSA) makes it possible to make maximum use of multiple commercial sources of both hardware and software at minimal cost. An OSA is a method for making the commercial-off-the-shelf (COTS) concept work. In addition, system design turnaround times can become much shorter when good open systems architectures are specified because of the potential availability of hardware and software products on the commercial market. There is also the fact that key interface problems are solved by the specification of the particular OSA selected. A good OSA permits rapid upgrading and prototyping of system enhancements once the system has been fielded.

Computer based modeling and simulation of systems is widely accepted as a powerful systems engineering tool. Some see these related tools as being absolutely essential in the design of complex systems. They share the attribute of allowing the systems designer to "test drive" different architectures and configurations on a computer prior to committing large amounts of resources and time to a design. Problems can be identified and either solved or evaluated as to seriousness prior to any final commitment. An advantage of the open systems approach is that an OSA is easily modeled, and the model can remain stable as long as the underlying standards that define the OSA are also stable. In addition, there are often multiple models of any given OSA available, so the systems designer can choose from the many. While each model may reflect different assumptions, a general sense of the design integrity can emerge from their use. The use of standardized systems models allows quick, low cost comparisons to be made on "before" and "after" versions of the system whenever changes are proposed. If the system were a closed point solution design, on the other hand, the program manager will be faced with the cost of creating and validating a model for the specific system, and that task may prove prohibitively expensive.

Despite its strengths, the OSA concept is not a panacea to all problems involved in avionics systems development. While the advantages of OSAs are substantial, and an argument can be made that OSA is the way the Navy should build most future systems, there are some disadvantages and problems that need to be addressed.

One major problem is that products are often touted as meeting the specification on which an OSA design is based, only to find out later that certain functions, memory locations or connector pins that are left available for the user's definition in the standard are used differently by different vendors.

Configuration control over modules will usually be lacking, so a user will find that a system element that worked well with the rest of the system becomes incompatible because of unilateral changes made to the user definable portions of the system.

Another disadvantage is that OSA solutions are usually not optimum solutions. All OSAs involve compromises of one sort or another, for that is the nature of consensus standards, and these may work together to make a solution sub optimum. The question for the system designer is whether or not the solutions allowed by the OSA are acceptable - as opposed to optimum - solutions. If they are not, then the systems designer may well want to consider a point design solution. Again, a rigorous systems engineering process is needed to discern truth.

Some opponents of open systems architecture assert that the very nature of OSAs requires the Navy to accept all of the package, or none of it. What this approach means is that bugs, errors, or glitches lurking undetected in the hardware or software - which was supplied as compliant to the standard - can cause unforeseen problems in the field (some of which may take years to surface). Because commercial vendors may not test all possible conditions as rigorously as military systems are tested, problems that are not critical to civilian users may not surface until the system is subjected to military environments. No one warrants a system to be free of such defects, especially when the system is made up of hardware and software products from different sources. An assignment of a key point of responsibility for the system is required, and it's not clear how this factor affects the cost savings of the OSA.

Maintenance problems also must be addressed in an OSA environment. The ideal situation is to be able to replace a module with a like module from another vendor. The office personal computer (PC) is often touted as the ideal implementation of this concept;

a VGA video card from manufacturer "X" can usually be replaced directly with a similar, but different, card from manufacturer "Y." The analogy is appealing, but is not always applicable to the military situation. It is also not true that cross-vendor module substitution is perfected in the PC world, especially in more complex modules; incompatibilities are found frequently. While it can be argued that the Navy should be able to cope with the situation, as commercial organizations do on a daily basis, the entire logistics situation needs to be evaluated further in light of the criticality of the Navy's missions.

A problem has been identified regarding parts substitution of OSA replacement modules from different vendors: the indiscriminate fielding through the logistics chain of replacement modules, from different sources than the original, may result in unsatisfactory performance. This problem arises from not just poor designs by some vendors, but also from different interpretation of ambiguities in standards and specifications, as well as different uses for user definable features of the standard. It is also the case that some vendors claim compatibility with a standard, even though they are not fully compliant. A program for testing of replacement modules, prior to certifying them fit for use in any particular application, is required before the benefits of OSA can be fully exploited.

A final disadvantage is that the Navy does not control future releases of the standards that underlie any particular OSA. While the Navy can, and does, participate in the standards setting process, the most commonly used standards are consensus standards formulated by industry at large. By using an open systems architecture, you potentially face a difficult decision later, i.e., whether to remain consistent with the rest of industry, or go it alone with an obsolete standard. This problem arises from evolving standards which may create products to replace those of older technologies only to discover that they no longer perform properly in older systems. Replacement parts in older systems, as well as newly manufactured systems, will not work properly unless they are upgraded to reflect the newest release of the industry standard.

There is wide agreement that OSA systems can reduce initial outlays in the early development phases of a system. However, we do not know whether the benefits imputed to OSAs in the Engineering & Manufacturing Development (EMD) portion of the system life cycle are maintained all the way through it. The costs of continuous retesting of new replacement modules may be greater than the initial cost savings of the OSA.

Implementing an effective open systems modular avionics architecture requires a simulation, modeling, and testing capability to validate candidate architectural concepts. The capability is needed to validate design logic, to ascertain expected systems performance with respect to standard benchmarks, and to test changes before committing to a specific design. When the OSA approach is fully implemented, the Navy probably will need a facility that is a center of excellence in simulation and modeling.

It is also necessary to test hardware and software against the selected standards to ensure inter operability of products from various vendors when used as an integrated system. Testing of hardware should include evaluation of performance over environmental extremes to guarantee adequate operation in the avionics environment. This environmental testing is especially critical if commercial off-the-shelf (COTS) hardware is selected for use. Such hardware is not usually designed specifically for the military avionics environment. Therefore, a testing capability must be achieved for OSA to be fully beneficial.

When technologically innovative solutions are offered, there is an element of risk present due to unknown performance and reliability issues. Selected flight test demonstrations might be required to gain the confidence in the technology and to reduce risk to the program manager prior to acquisition decisions. When selected flight testing is coupled with ground-based simulation and modeling, it becomes possible to mitigate risk at low cost.

### 3-6 Buses Provide Architecture Communications

The term *bus* is used a lot in the context of computers systems, but the word "bus" is applied to more than one thing. These different uses of the term "bus" all have similar purposes: transferring digital computer data from one place to another. Beyond that similarity, however, the various uses of the term are different. In this section are described some common forms of bus that might be found singly, or in combination, in avionics computer systems.

#### 3-6.1 Box-to-Box Interconnect Buses

One familiar type of bus is the *box-to-box interface buses*, an example of which is MIL-STD-1553. Such a bus defines the inter-box wiring, the connectors, the pin assignments, and the operational protocols governing how several boxes in the same system talk together. The concept of this type of bus is similar to the more familiar Local Area Network (LAN) used in modern offices. The LAN allows computers in different elements of the organization to communicate with each other. Similarly, the box-to-box 1553-style interconnects allow the different elements of the avionics suite to communicate with each other.

#### 3-6.2 Peripherals Interface Buses

This form of bus is used to interconnect peripherals within a computer system. For example, the circuit cards within a computer need to talk to hard disk drives, floppy disk drives, CD-ROM drives and other peripheral devices. To accomplish this job, the designer may opt to use a peripherals bus such as the *Small Computer Systems Interface* (SCSI) bus. A plug-in SCSI bus controller will be connected to the computer motherboard ("backplane bus" — see below), and in turn is connected via a multi-conductor cable to the peripherals. Most personal computers have at least one of these types of bus, as does the Navy standard desktop computer (DTC-2) used in TAMPS.

A related type of bus is intended to bring together several different types of equipment, which may or may not be computers, and cause them to function as a system. While a box-to-box interface bus, or a regular peripherals bus, might simply operate as a communications pathway, buses such as the IEEE-488 *General Purpose Interface Bus* (GPIB) serve a different, if similar, function. The IEEE-488 GPIB is intended for use in automated test equipment, and uses a central computer controller to execute a program that performs tests and measurements. In this type of bus, the "listeners" and "talkers" on the bus act as peripherals of a single controlling computer. For example, radio receiver sensitivity tests might use a central computer driving as peripherals radio test equipment such as DC power supplies, signal generators, distortion meters, voltmeters, and oscilloscopes. Each piece of discrete test equipment is fitted with an IEEE-488 compatible interface connector, and its operation is under the control of the central computer.

### 3-6.3 Backplane Buses

The *backplane bus* defines the internal motherboard of a computer system: connectors, pin assignments, operational protocols, etc. Modules (printed circuit boards) plug into the backplane bus, which serves the interconnection function to allow modules to communicate with each other, as well as draw power from the system. The purpose of the backplane bus is to define and standardize the way plug-in circuit cards talk to each other, similar to the way MIL-STD-1553 defines how boxes talk to each other. The 4 $\pi$  bus used in the A-6E mission computer, and the advanced JIAWG bus are examples of current military computer backplane buses; the VMEbus and Futurebus+ are examples of commercial backplane buses (both of which are now being adopted to military purposes). It is often the case that the backplane bus standard will also define the plug-in printed circuit cards, card racks, and other mechanical aspects of the system. In some cases, for example the VME bus, a subsidiary specification is used to define the mechanical aspects.

### 3-6.4 Sub-Buses Within The Backplane Bus

Backplane bus usually contains several sub buses that are also commonly called "buses." These include such functions as *address bus*, *data bus* (or *data transfer bus*), *priority interrupt bus*, *arbitration bus*, and *utility bus*. While the backplane bus is important to the outside user insofar as selecting the right plug-in cards is concerned, these sub buses are rarely important to the outside user (although often critically so to the system designer and programmers).

A problem that is sometimes seen, however, is that the backplane bus will have either unused or "user configurable" pins or data pathways. While these features were intended to improve flexibility, they have the effect of creating conflicts when different after-market vendors produce cards that use these features in incompatible ways.

It is often the case that a computer will have an internal backplane bus, which is transparent to the outside world, as well as a plug-in card that contains the box-to-box interface bus. The backplane bus controls internal operations, while the box-to-box bus controls inter-box operations.

When the same word — such as "bus" — is used for several different things it makes understanding a more difficult. Understanding these differences is key to understanding the architectures of modern computer based avionics systems.

## 4-0 MAJOR AVIONICS ARCHITECTURES

This section describes some of the current and evolving systems architectures that are considered viable candidates for avionics systems in naval aircraft.

### 4-1 Joint Integrated Avionics Working Group (JIAWG)

The fiscal year 1987 DOD appropriations Act Conference Report (Report No. 99 1005), dated 15 October 1986, directed the Service Representatives to prepare a joint plan for the inclusion of fully integrated, digital avionics, communications, sensors, embedded communications security and other electronics on all aircraft under development. Accordingly the Assistant Secretary of Defense for Command Control, Communications and Intelligence directed the Army, Navy and Air Force prepare a joint plan to meet the intent of the aforementioned congressional direction. To this end, the Joint Integrated Avionics Plan (JIAP) for New Aircraft, dated March 1987 was prepared and forwarded to congress outlining the Tri-Service plan to achieve this goal.

#### 4-1.1 Technology Base/Related Programs

The JIAWG draws heavily upon the bus and processor technologies of the Very High Speed Integrated Circuits (VHSIC) and PAVE PILLAR programs. The next generation of these programs continue to impact JIAWG.

The VHSIC program was begun in 1983 to provide seed money for the development of industrial production capacity for high performance military microcircuit technologies. This was motivated by the increasing use of foreign supplied key components, as well as the divergence of military and commercial computer needs. In order to assure that components from all participating suppliers were of general utility, VHSIC established inter operability standards for numerous communication channels. The principal products remaining from these efforts are the standards for Pi bus (see discussion contained in section 6-2.1) and TM bus (see discussion contained in section 6-2.2).

In 1985, the Air Force Aeronautical Systems Division started the PAVE PILLAR effort as an adjunct to avionics concept exploration for ATF. PAVE PILLAR conducted numerous technology demonstrations based on products from the VHSIC program. These include the first 16 bit Pi bus and VHSIC MIL-STD-1750 processor implementations in the VHSIC Avionics Modular Processor (VAMP).

After completion of VHSIC and PAVE PILLAR, the VHSIC products became the responsibility of JIAWG. JIAWG management funded numerous risk reduction efforts designed to mature and validate the VHSIC inter operability specifications. These efforts included detailed gate level simulation of vendor designs intended to identify implementation and specification problems with JIAWG updates to the VHSIC specifications.

The JIAWG Advanced Avionics Architecture (A<sup>3</sup>) is an integrated avionics architecture, based on open architecture principals. The A<sup>3</sup> level of integration is both physical and functional. Physical Integration is implemented in the avionics suite as a processing network with high levels of connectivity among the processing, storage, and input/output (I/O) elements to support real-time exchange of data, dynamic task

allocation, hierarchical built-in diagnostics, and other system capabilities. Functional Integration is a coordinated functioning of the elements of the avionics suite as a single information processing entity to implement fusion of data from multiple sources, synthesis of cockpit display contents, and other capabilities based on shared data and processing. (Figure 4-1.1) The Open architecture tenant is based on; explicit provision for expansion or upgrading through the incorporation of additional or higher performance elements, definition of interfaces, including physical, functional to facilitate integration of new or additional system capabilities and elimination of proprietary features. It has not been determined if any JIAWG specifications will have proprietary features.

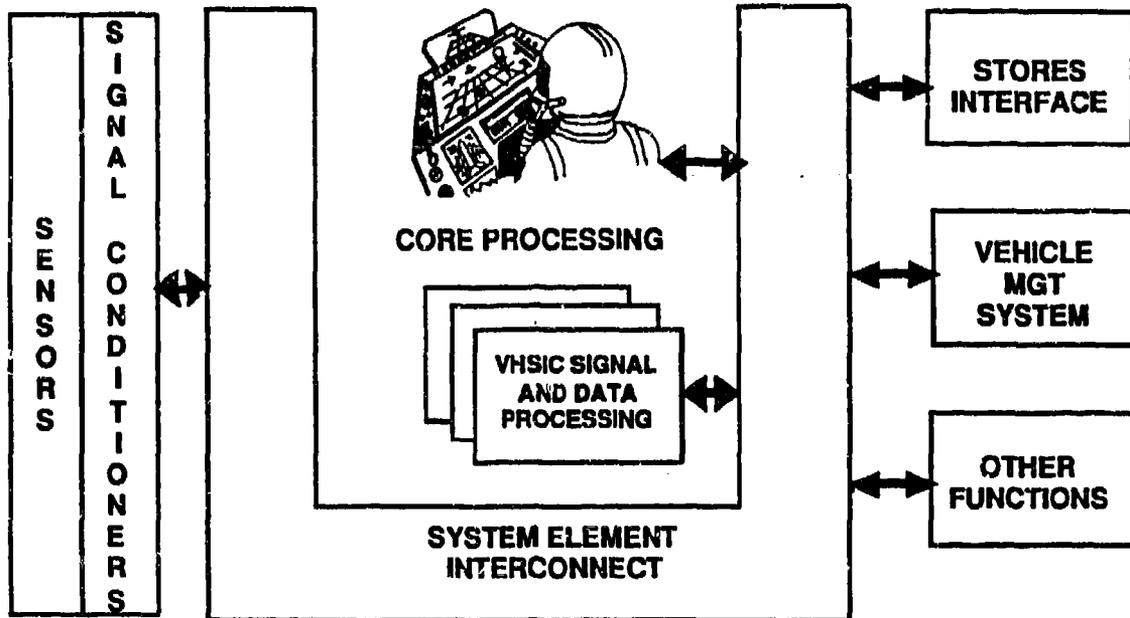


FIGURE 4-1.1 - JIAWG Advanced Avionics Architecture

The JIAWG Standards and Specifications are defined by the Common Avionics Baseline (CAB) consists of Specifications, Standards, and other appropriate documents defining an inventory of modules, interconnects and design tools which can be used to assemble avionics systems for multiple aircraft while maintaining a high level of commonality. The documents define items such as hardware components, software modules, and system integration tools. The JIAWG standards and specifications are released periodically as the CAB, based on the major milestones of the three primary JIAWG programs. The current release is CAB III/Rev 3. Table 4-1, CAB Release Dates, shows that the release dates are driven by major milestones of the three primary tactical aircraft programs.

Table 4-1. CAB Release Dates

<u>CAB</u>	<u>DATE</u>	<u>PROGRAM MILESTONE</u>
I	January 88	All three programs
II	December 88	All three programs
III	April 89	All three programs
III, Revision 2	April 90	LH Milestone II
III, Revision 3	November 90	ATF Milestone II
III, Revision 4	TBD	AFX Milestone I
IV	October 94	ATF, LH Milestone III, AFX Milestone II
V	October 95	AFX Milestone III

Four areas of standardization are being addressed by the JIAWG, are listed below:

#### 4-1.2 Core Avionics

Core avionics are the modular computer resource assets that can be integrated into the platform to implement data, signal, and other central processing functions required in an integrated avionics suite. The Core Avionics include:

- Architecture
- Backplane
- Data Buses
- Packaging
- Power supplies
- Processors

#### 4-1.3 Mission Avionics

Mission avionics includes sensors, Communications, Navigation, Identification/Communications Security (CNI/COMSEC), inertial navigation, and other aspects of the integrated avionics suite which furnish offensive, defensive, and mission management functions.

#### 4-1.4 Supportability

- Configuration/Data Management
- ILS/Training
- R&M/Ground Support

#### 4-1.5 Software

- Ada Language/Real Time Support
- Software Engineering Environment
- Software Standards & Practices
- Software Reusability

#### 4-1.6 JIAWG Standards & Specification Descriptions for Core Avionics

The JIAWG standards and specifications are described in the following sections. The section on Core Avionics and in particular, the Architecture, Processor, Data Bus and Backplane standards and specifications are done in more detail than the other sections because of its applicability to the present study. The JIAWG core avionics are described in the section below.

##### 4-1.6.1 Architecture

The JIAWG architecture standard (J87-01) describes an open system architecture which provides provisions for expansion and upgradability through the explicit definition of all avionics system interfaces. The JIAWG architecture provides the overall description of the structure and function of an avionics system, including the top-level functional partitioning, network topology, signal and data communications protocols, hardware and software interfaces, and procedures for system control and resource management. Modules form the basic component or "building block" of the JIAWG architecture. Modules may be grouped in clusters to provide avionics functions. Clusters may be grouped to create system elements which provide sub-system capabilities. Figure 4-1.6.1 depicts the A<sup>3</sup> hierarchical structure.

This architecture approach provides an avionics system which has flexibility, is fault tolerant, can be reconfigured, and has the ability to share resources.

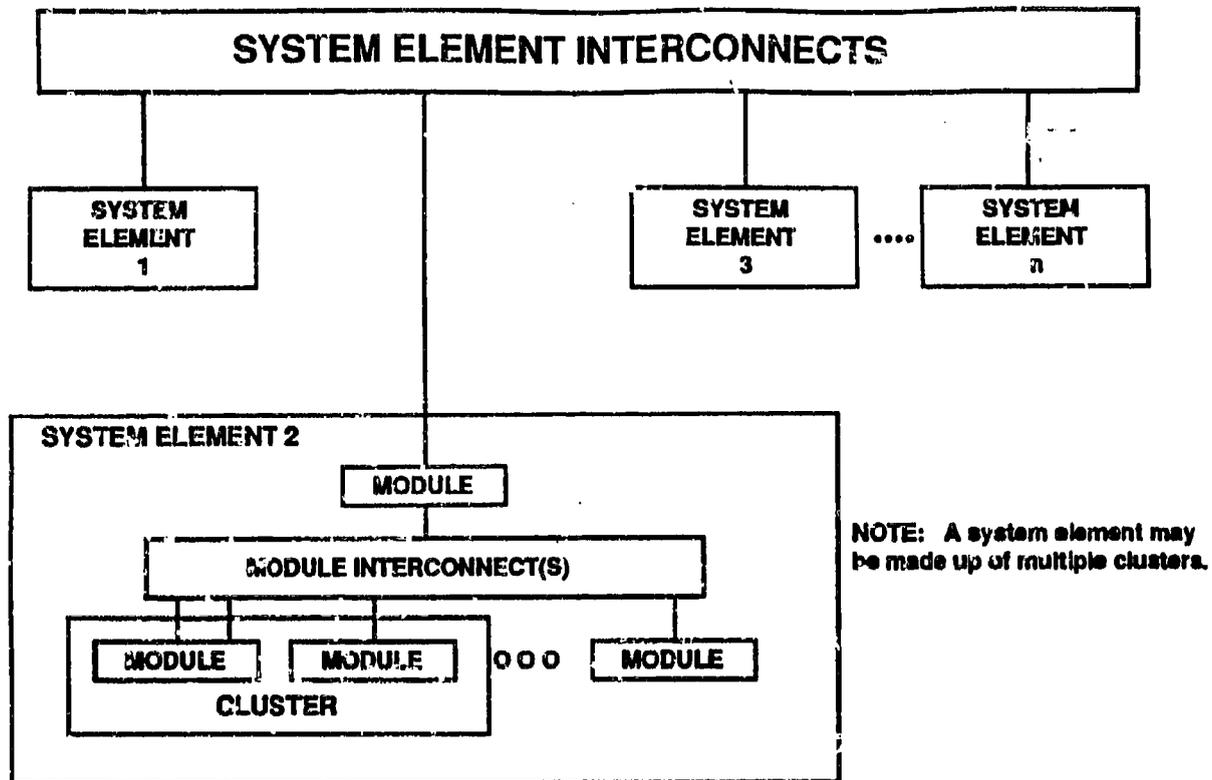


Figure 4-1.6.1 - JIAWG A<sup>3</sup> hierarchical structure.

#### 4-1.6.2 Processors

**Data Processing Modules.** The following modules are used as applicable within a data processing system element in which the primary data path among modules is the JPI-bus.

**16 Bit Processor** provides the necessary computational, control, data storage, interface and BIT capability to configure a single or multiple processor system that will perform the data and/or control processing for an advanced integrated avionics architecture. This specification is based on MIL-STD-1750.

**32 Bit Processor** provides the necessary computational, control, data storage, interface and BIT capability to configure a single or multiple processor system that will perform the data and/or control processing for an advanced integrated avionics architecture. This processor conforms to either of the 32 bit Computer Instruction Set Architecture Specifications, as described by JIAWG CAB document J89-M2D1.

**Signal Processor** is a general purpose processing element employing either of the 32 bit Computer Instruction Set Architecture Specifications, as described by JIAWG CAB document J89-M2D1.

#### 4-1.6.3 Data Buses

The following data transmission types are used for connecting system elements.

High Speed Data Bus (HSDB). The HSDB Protocol shall be as defined in the Linear Token Passing Multiplex Data Bus Protocol, as described by JIAWG CAB document J88-N2. The HSDB is a fiber optic, linear, token passing data bus capable of 50 MBPS data rate.

Signal Data Distribution Network (SDDN). High speed, low latency data transmissions will use the JIAWG SDDN as defined by the Signal Data Distribution Network Specification, as described by JIAWG CAB document J89-N5. This network is a point-to-point, fiber optic data link operating at rates on the order of hundreds of megabits per second and interconnecting apertures/signal conditioners with core processing

Video Data Distribution Network (VDDN). High speed, low latency digital video data transmissions will use the JIAWG VDDN as defined in the Video Data Distribution Network Specification, as described by JIAWG CAB document J89-N6. This network is a point-to-point fiber optic link operating on the order of hundreds of megabits per second and interconnecting crew station video displays with core processing.

High Bandwidth Interface (HBI). The HBI will provide the primary path for transmission of high bandwidth data between core processing system elements. (J89-SP-01).

MIL-STD-1553 Data Bus. The 1553 is a 1 MBPS, serial, multiplex data bus, employing wire medium, with a command response protocol.

MIL-STD-1773 Data Bus. The 1773 is a 1 MBPS, serial, multiplex data bus, employing a fiber optic media, with a command response protocol identical to MIL-STD-1553.

MIL-STD-1760 Stores Interface. Stores control and data interconnects shall use the MIL-STD-1760 interface standard.

Discretes Selection Criteria. The use of discretes between system elements in architectures compliant with this standard is discouraged, and shall be limited to interfaces which are judged inappropriate using MIL-STD-1553 or HSDB implementations. Discrete interfaces will be selected to provide maximum reliability and maintainability.

#### 4-1.6.4 Backplane

All module interconnects will meet the requirements of the Module Interconnect Document, as described by JIAWG CAB document J87-N1.

JIAWG Parallel Inter Module Bus (JPI-Bus). The JPI-bus is defined by the JIAWG PI-bus Specification, as described by JIAWG CAB document J89-N1A.

JIAWG Test/Maintenance Bus (JTM-bus). The JTM-bus is defined by the JIAWG TM-bus Specification, as described by JIAWG CAB document J89-N1B. The JTM-bus is a serial path for test and maintenance control and data communications within

a system element. The bus is a linear, master-slave protocol, multi-drop medium operating at a clock rate of 6.25 MHz.

Utilities. Module interconnect discrettes, power distribution, and grounds will be those identified in the Utility Signals Specification, as described by JIAWG CAB document J89-N1C. There shall be no other discrettes included in the module interconnection.

Local Memory Bus (LM-bus). The LM-bus shall be used to interconnect the EM-32 module with its host DF-CAP-32 processor module as defined in the Local Memory Bus Protocol Specification, as described by JIAWG CAB document J89-N1D. The LM-bus is a single-master, parallel, high-speed, very low-latency direct memory access (DMA) bus. This bus shall be used within a cluster between a processing module and an extended memory module when the on-board processor memory is not large enough to support application program needs.

Signal Processing Network (SPN). The SPN will be defined in the Signal Processing Network Specification, as described by JIAWG CAB document J89-N1E (to be developed, see J90-SP-01). The SPN provides high-speed, data transmission paths between module performing a common function. The SPN will be a network switching concept which supports simultaneous interconnections of any independent pair of sources and destinations of data connected to a network.

User Console Interface (UCIF). The UCIF is defined in the User Console Interface Specification, as described by JIAWG CAB document J89-N1F or the User Console Interface Specification for 32 Bit Modules, J89-N1H. The UCIF defines all software development interface functional capabilities that shall be supported by JIAWG modules. This is done for both system level testing of hardware and software systems, and for specific testing of the user software.

Signal Processing Bus (SP-Bus). The SP-bus will be defined in the Signal Processing Bus Specification, as described by JIAWG CAB document J90-N1J. The SP-bus provides labeled message communications, control, and event signaling between clusters or functional elements (see J89-SP-01).

#### 4-1.6.5 Packaging

JIAWG architectures are implemented using a modular electronics approach. This approach utilizes standard electronics modules integrated in modular racks and interconnected via an electrical backplane. Modules in the integrated avionics suite now must conform to the Standard Hardware Acquisition & Reliability Program (SHARP) Standard Electronic Module Format-E (SEM-E) as specified in MIL-STD-1389. This standard defines the physical, electrical, and environmental characteristics for standard avionics modules. JIAWG modules rely on various cooling systems concepts for normal operation and to achieve reliability requirements. Current generation cooling systems include; conduction, air, and liquid flow through.

Standard Module. The JIAWG Standard Module specification (J88-G2B) is being developed that addresses the physical and electrical characteristics for developing avionics modules which meet the Form, Fit, Function and Interface (F<sup>3</sup>I) requirements necessary to implement JIAWG architectures.

**Standard Connector.** The JIAWG Standard Connector specification (J87-G2A) defines the physical (form, fit) and electrical (interface) characteristics of the JIAWG module connector.

#### **4-1.6.6 Power supplies**

The following modules will be used as applicable for power conditioning within integrated rack/enclosures.

**Airborne Standard Power Supply Specification, J88-M7A.** This module is a 270 VDC input, 5 volt, 50 ampere output, airborne power supply.

**Airborne Standard Power Supply Specification, J88-M7B.** This module is a 270 VDC input, multiple output power supply with outputs of +15V at 2 ampere, -15V at 2 ampere and 5V at 30 ampere.

**Airborne Standard Power Supply Specification, J88-M7C.** This module is a 28 VDC input, 5 volt, 32 ampere output, airborne power supply.

#### **4-1.7 JIAWG Standards & Specification Descriptions for Mission Avionics**

**ICNIA** The integrated CNI system as defined by this standard will be capable of performing any subset of the following:

- Communications functions:
- Radio navigation functions.
- Identification functions.

#### **4-1.8 JIAWG Standards & Specification Descriptions for Supportability**

Module designs will conform to JIAWG specifications for Reliability and Maintainability (J88-G3), Configuration Management (J88-G4), and Integrated Logistics Support (J88-G6) supportability requirements.

Module design will support system-level diagnostics and testability. The design and implementation of all electronic equipment shall provide fail-safe features to ensure personnel safety during all phases of operation and maintenance. Additional supportability requirements are contained in the individual module specifications. The anticipated operating and maintenance environment is specified in the Standard Module Specification, J88-G2B, Environmental Requirements.

**ILS/Training** JIAWG ILS elements are defined in DoD Instruction 5000.2. ILS responsibilities will flow down from contractor to subcontractor through the Logistics Support Analysis/Logistics Support Analysis Record (LSA/LSAR) process as described in MIL-STD-1388/1A and MIL-STD-1388/2A. The LSA program will be a single, analytic effort interfaced with the design process and systems engineering process. Detailed requirements are contained in the Integrated Logistics Support Standard, J88-G6.

R&M/Ground Support Common module designs will be compatible with a two-level maintenance concept and shall minimize dependence on depot special test equipment. Detailed requirements are contained in the Reliability and Maintainability Specification, J88-G3, and in individual module specifications.

Configuration and Data management System module designs will be controlled in accordance with the JIAWG Configuration Management Plan, J88-G4.

#### 4-1.9 JIAWG Standards & Specification Descriptions for Software

Ada Language/Real Time Support. The JIAWG software task group is currently participating in the development of the following concept papers which have application to this interface:

- Common Ada Run-time (CART), and
- Catalog of Interface and Function Options (CIFO).

These documents represent the current state of progress in the area of a common operational run-time environments. When a sufficient number of the associated issues are resolved and appropriate detail is available, those documents will be proposed as specifications or standards in a future CAB release.

Software Engineering Environment. The Software Engineering Environment (SEE) will support the development of an A3 OFP in accordance with the JIAWG Software Engineering Environment Specification, as described by JIAWG CAB document J89-S3.

Software Standards & Practices for Software Reusability The OFP software architecture should support reuse, system modularity and reconfiguration. The current CAB document set contains the following concept papers which discusses the issues associated with reusability within the OFPs of the JIAWG weapon systems:

- J89-S7 Software Reuse Concept Paper, and
- J89-S9 Software Reuse Domain Analysis Concept Paper.

These documents represent the current state of progress in the area of software reuse. When a sufficient number of the associated issues are resolved and appropriate detail is available, those concept papers will be proposed as specifications or standards in a future CAB release.

Software Development Standard. The OFPs for JIAWG programs shall be developed in accordance with DOD-STD-2167A. The JIAWG Tailored DOD-STD-2167A, J89-S6, shall be utilized as the standard approach to tailoring DOD-STD-2167A and its applicable Data Item Descriptions (DIDs).

## 4-2 Next Generation Computer Resources (NGCR)

The NGCR program provides an open system architectural framework for computer resource hardware and software standards capable of meeting the Navy's mission critical computer resource (MCCR) requirements. It encompasses all tactical embedded computer resources aboard the full range of naval systems and platforms, including aircraft, surface ships, submarines and shore locations. These standards provide a mix of low, medium and high performance systems to meet the diverse computing needs of future naval systems. They permit program managers and system developers to design and build MCCR systems with enhanced levels of product commonality and inter operability. Products conforming to NGCR standards will be implemented in a wide range of naval systems and will provide for a full spectrum of functions from data manipulation and communications routing to signal and symbolic processing.

NGCR interface, protocol and service standards are intended to become the primary mechanism by which the Navy alters its traditional development and acquisition strategies for standard embedded computer hardware and software resources. The purpose of this transition is to maximize the potential benefits inherent in applying an open systems architecture (OSA) approach to the development, procurement and upgrade of future tactical airborne weapons systems.

The Navy, through the NGCR program, has selected an industry-based open systems architecture and corresponding acquisition approach which stresses: reliance on industry trends and investments in hardware and software technologies, increased market competition, inter operability, modularity, and the ability to field cost-effective technology improvements in a timely manner.

Three general areas of standardization are being addressed by the program: multiprocessor interconnects, multi system interconnects, and software interfaces. Nine standards are being defined or selected in the standardization areas:

### A. Multiprocessor Interconnects

- Baseline Backplane and Modular Open Architecture
- High Speed Data Transfer Network (HSDTN)
- High Performance Backplane

### B. Multi system Interconnects

- SAFENET
- High Performance Local Area Network (HPLAN)

### C. Software Interfaces

- Operating System Interface
- Project Support Environment Interface
- Graphics Interface
- Data base Management System Interface

#### 4-2.1 NGCR Standards Descriptions

The NGCR standards are described in the following sections. The section on Multiprocessor Interconnects, and in particular, the Baseline Backplane and Modular Open Architecture standard is done in more detail than the other sections because of its applicability to the present study.

#### 4-2.2 Multiprocessor Interconnects

The three NGCR multiprocessor interconnects are described in the section below.

#### 4-2.3 The Baseline Backplane and the NGCR Modular Open Architecture

The baseline backplane and the NGCR modular open architecture is discussed in MIL-STD-2205 *Interface Standard for a Modular Open Architecture (Draft)*. The NGCR modular open architecture breaks computer and electronic systems into board (module) level components with standard interfaces between the boards. Figure 4-2.3.1 shows the three NGCR standard board form factors and some intended applications for each. The IEEE 896.5 standard, *Futurebus+ Military Profile Specifications*, and its referenced documents, provide the details for each form factor. The three form factors are the MIL-SEM-E intended for integrated rack use, MIL-10SU intended for ATR box and 6U VME replacement use, and MIL-12SU intended for milder environment cabin avionics and shipboard use.

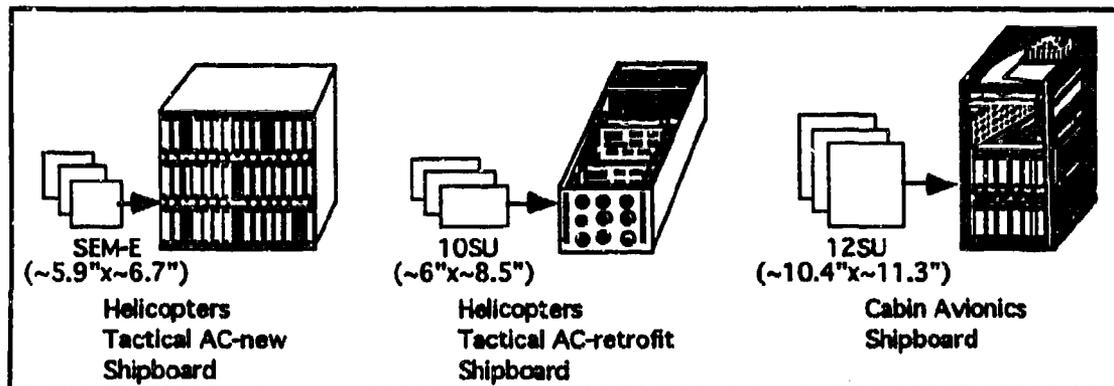


Figure 4-2.3.1 - NGCR Board Form Factors and Applications

Figure 4-2.3.2 shows the standard board level interconnects for the NGCR program. These are the IEEE 896 Futurebus+, the IEEE 1394 High Speed Serial Bus, and the High Speed Data Transfer Network (HSDTN). Detailed pinout and other specifications for Futurebus+ and Serial Bus are contained in IEEE 896.5. While the Futurebus+ is the primary backplane interconnect, the auxiliary Serial Bus is used for test and maintenance, software debug, low bandwidth data transfer and other purposes. Customized local interconnects among a few boards, input/output interconnects, backplane discretes, control and status registers, power, grounding, mechanical interfaces, and thermal interfaces are also defined in IEEE 896.5.

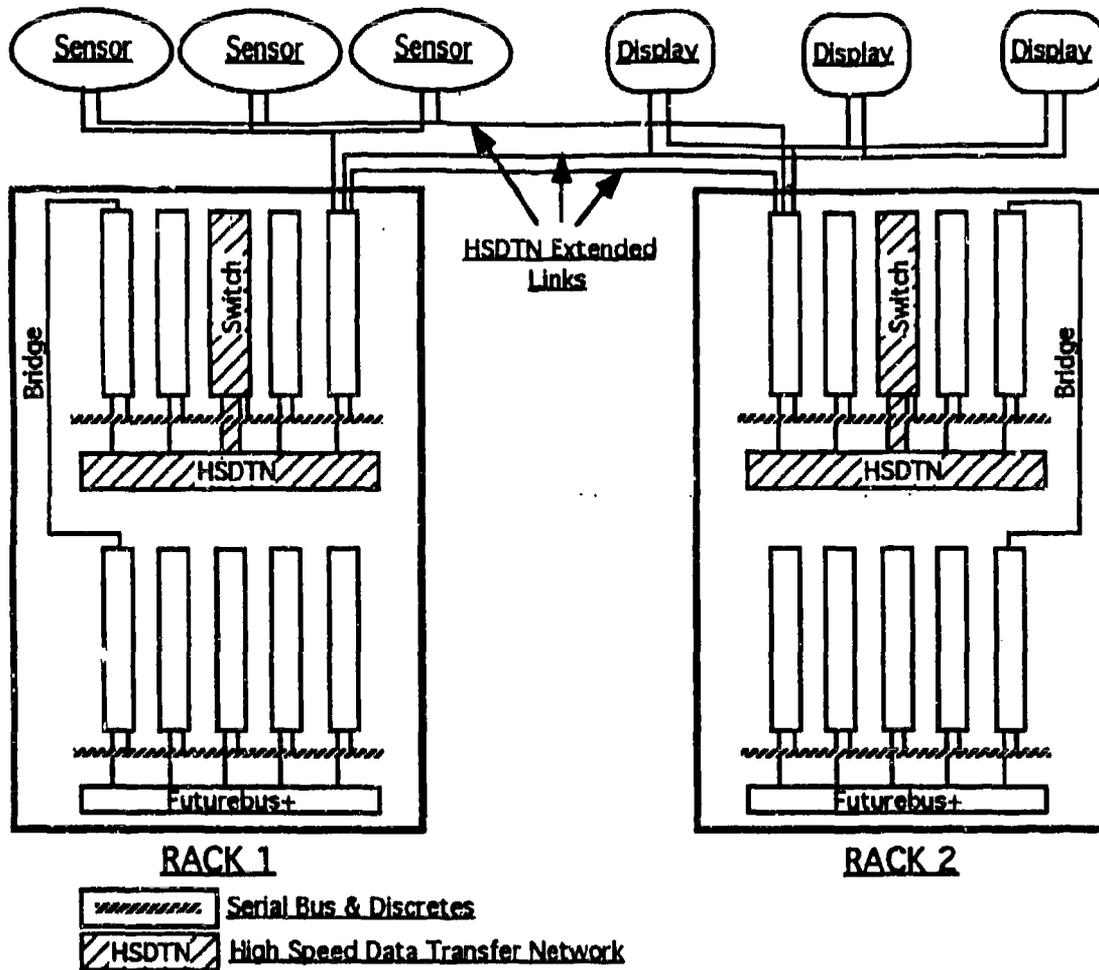


Figure 4-2.3.2 NGCR Modular Open Architecture; Board Level, Sensor / Display, and Inter-rack Interconnects

#### 4-2.4 The High Speed Data Transfer Network (HSDTN)

The HSDTN augments the baseline backplane (Futurebus+ / Serial Bus linear buses) with a backplane switched network and a backplane ring network. Figure 4-2.3.2 shows HSDTN used as a switched network. This figure also shows the HSDTN extended links connecting to sensors and displays as well as interconnecting separate racks. This sensor/video extended interconnect network contains 1+ GBPS point to point links for bringing very high speed sensor information to processing racks, for driving displays with refresh memory remote from the display, and for interconnecting racks at backplane speeds. The HSDTN allows multiple simultaneous conversations among boards and among Futurebus+ clusters. It also allows simultaneous distribution of sensor data to multiple racks, and distribution of video data from multiple racks, for fault tolerance or other purposes. The IEEE 1596 Scalable Coherent Interface (SCI), Asynchronous Transfer Mode (ATM), and Fibre Channel are the three leading candidates for selection as the base standard for HSDTN.

**4-2.5 High Performance Backplane (HPB) Standard**

The third multiprocessor interconnect standard, the High Performance Backplane (HPB), has not yet been initiated, but the effort will start in September 1997. Candidate technologies for the HPB mostly revolve around the use of optics, although further improvements to electrical protocols have not been ruled out. Optical technologies of interest include fiber optics, free space optics, optical switches, etc. Electrical technologies of interest include lower power, lower latency protocols, and higher performance electronics.

**4-2.6 NGCR Modular Open Architecture Features**

Table 4-2.6.1 contains a "Requirements versus Supporting Features" itemization for the NGCR modular open architecture. It shows the specific features developed for satisfying the various system requirements.

**Table 4-2.6.1. NGCR Modular Open Architecture Features**

System Requirement	NGCR Modular Open Architecture Supporting Feature
<p><u>High Performance</u></p> <ul style="list-style-type: none"> <li>• Emerging multi-hundred megahertz computer chip support.</li> </ul>	<ul style="list-style-type: none"> <li>• Futurebus+ at up to 6.4 GBPS is 10 times faster than previous buses</li> <li>• HSDTN at up to 8 GBPS on each leg of switched network</li> <li>• Cache coherent shared memory</li> </ul>
<p><u>Low Weight, Volume, Power</u></p> <ul style="list-style-type: none"> <li>• Emerging ultra high density electronics support.</li> </ul>	<ul style="list-style-type: none"> <li>• Low heat 3.3 volt power</li> <li>• High performance flow through cooling (under development)</li> </ul>
<p><u>Economical systems</u></p>	<ul style="list-style-type: none"> <li>• Based on widely used commercial standards to allow use of economical commercial electronics technology</li> <li>• Defines interfaces so components from different vendors can work together and be used in multiple systems providing economies of scale</li> <li>• At the board level so components are "small" enough that they can be developed on vendor's own money thereby shifting development costs away from government</li> <li>• Supports ruggedized and mil-spec versions of standard commercial boards</li> <li>• Supports full ATR size boards on 0.8" pitch to accommodate cheaper high profile devices and heat sinks up to 0.2" for economical but high capacity air flow through cooling</li> </ul>

Table 4-2.6.1. (continued) NGCR Modular Open Architecture Features

<u>Reduced development &amp; upgrade times</u>	Open architecture allows system development from previously developed components as well as piecemeal upgrades with newer components
<u>Both new platform and retrofit support</u>	Three board sizes to fit efficiently into the available space in different platforms: <ul style="list-style-type: none"> <li>• MIL-12SU (~10.4"x~11.3")</li> <li>• MIL-10SU (~6"x~8.5")</li> <li>• MIL-SEM-E (~5.88"x~6.68")</li> </ul>
<u>Varied platform environmental requirements support</u>	Four environmental levels supported: <ul style="list-style-type: none"> <li>• Commercial of the shelf</li> <li>• Ruggedized</li> <li>• Full mil-spec (shipboard)</li> <li>• Full mil-spec (airborne)</li> </ul>
<u>Integrated architecture support</u>	<ul style="list-style-type: none"> <li>• Very high performance backplane and switched network (HSDTN) allow combining of data from different sensors</li> <li>• Cache coherent shared memory (as well as message passing) for efficient tightly coupled processing</li> </ul>
<u>Fault tolerance</u>	<ul style="list-style-type: none"> <li>• Optional dual backplane buses</li> <li>• HSDTN provides rack to rack interconnects at backplane speed for N+1 redundancy across racks</li> </ul>
<u>Maintainability and reliability support</u>	<ul style="list-style-type: none"> <li>• Serial bus for test and maintenance, software debug, and miscellaneous functions</li> <li>• On board stress and error history log</li> <li>• On board revision history log</li> <li>• Live insertion (some form factors)</li> </ul>
<u>Software development support</u>	<ul style="list-style-type: none"> <li>• Standard unit for real time non intrusive debug of integrated systems</li> <li>• Both cache coherent shared memory and message passing for either tightly or loosely coupled systems</li> </ul>
<u>Massively parallel processor support</u>	HSDTN provides standard mesh (or other) interconnect for up to 65000 processors.

Table 4-2.6.1. (continued) NGCR Modular Open Architecture Features

<u>Fiber Optic Interconnects</u>	<ul style="list-style-type: none"> <li>• Optional fiber optic contacts in backplane connector</li> <li>• HSDTN may optionally have fiber optic physical layer</li> </ul>
<u>Input/output from external to the rack</u>	Standard pin assignments for commonly used I/O such as Mil-Std 1553, Mil-Std 1397 and others.
<u>Multi-board functional elements</u>	Connector space allocated for custom interconnects among a few boards making up a multi-board functional element.
<u>Hierarchical bus structures</u>	Modules with dual buses may use one bus as a secondary bus to form clusters or other bus structures.
<u>Sensor/video network</u>	HSDTN provides 1+ GBPS serial link.
<u>Mixed digital and analog/RF systems</u>	<ul style="list-style-type: none"> <li>• +/-15 volt power</li> <li>• Separate analog ground</li> <li>• Clock signal</li> <li>• Serial bus to be used as RF control bus</li> <li>• Optional coax contacts in backplane connector</li> <li>• Optional board covers</li> </ul>
<u>Real time usage</u>	<ul style="list-style-type: none"> <li>• Clock coordination accuracy to 50 ns or better</li> <li>• Up to 256 priority levels for deterministic scheduling</li> </ul>
<u>Secure system usage</u>	Classified memory erase message.
<u>Nuclear event survival</u>	Nuclear event shutdown discrete.
<u>Unstable power system ride through</u>	<ul style="list-style-type: none"> <li>• Power fail imminent warning message and discrete allows orderly shutdown</li> <li>• Battery backup power</li> </ul>
<u>Signal processor support</u>	<ul style="list-style-type: none"> <li>• Futurebus+ provides the speed to handle many signal processing applications on single linear bus</li> <li>• HSDTN provides a switched network for signal processing applications requiring multiple simultaneous data transfers</li> </ul>

#### 4-2.7 Standards

The local area network standard has been in preparation since January 1987, and was scheduled for final publication in September 1992. The network is a dual-ring, token passing LAN-based standard for inter computer and computer-to-peripheral data transfer. It is based on the ANSI X3T9.5/84-89 Fiber Distributed Data Interface (FDDI) standard.

##### 4-2.7.1 High Performance Network Standard

The HPNET working group will be started in September 1992, with publication of the standard targeted for March 1997.

##### 4-2.7.2 Software Interfaces

Four different software interface areas are being standardized, and these are described in the sections below.

##### 4-2.7.3 Operating System Interface (OSIF) Standard

The OSIF effort commenced in March 1989 to prepare a commercially-based family of operating system (OS) interfaces. Following the requirements definition and a survey of available technologies and standards activities, POSIX (IEEE 1003) was selected as the baseline standard for the NCR OSIF standards. The Operating System Working Group is now actively involved in the IEEE 1003 project to participate in the POSIX standards definition. This set of standards will address systems which are Ada-oriented, real-time, distributed/networked, multi-level secure, reliable and realizable on heterogeneous processors. The initial standards will be published in October 1995.

##### 4-2.7.4 Project Support Environment (PSE) Standard

The PSE standardization effort was initiated in April 1991, and will define commercially-based PSE framework interfaces and protocols. The standards will not define standard tools or tool sets to be mandated for use in Navy systems, but rather will focus on tool integration mechanisms, data interchange mechanisms, and the logical contents of project data repositories. The adoption of a harmonized set of standards for PSE interfaces, services, and protocols will provide a means for better integration within a PSE and better interaction between and among different PSE implementations. These standards will be used in the development and deployment of Navy hardware and software applications in the mid- and late-1990s. This work is being carried out in concert with industry, DoD, government, and national and international standards organizations. The Software Technology for Adaptable Reliable Systems (STARS) program and the National Institute of Standards and Technology (NIST) are prominent among the groups with which the group's work is being coordinated. The draft PSE standards will be available starting in late 1995, with the final set targeted for publication in September 1998. The primary objectives of the standard are to: allow the "mixing and matching" of various PSE tools from different vendors, minimize user training, provide hardware and language independence, achieve host interchangeability, respond to the requirement for use of Ada, achieve compatibility with the other NCR standards, and span a wide range of applications.

4-2.7.5 Data Base Management System (DBMS) Interface Standard

The DBMS standard effort was initiated in May 1991 with the preparation of a white paper and organization planning for the working group, which was to be formed in September 1992. This standard will define DBMS interfaces for naval systems which are typically real-time or critical-time, heterogeneous, distributed, language and operating system independent, network independent, secure, and fault tolerant. Current trends in commercial and military DBMS technology as applied to C3, sensor, intelligence and weapon systems will be assessed to define standard interfaces for a broad range of platform applications. Publication of this standard is scheduled for September 1998.

4-2.7.6 Graphics Language Interface Standard

The Graphics Interface Standard working group was also scheduled to commence in September 1992, and has a scheduled standard publication date of September 1998.

4-2.7.7 NGCR Standards Applied to an Avionics System

Figure 4-2.7.7.1 illustrates the application of the various NGCR interface standards to an integrated avionics system. These hardware and software standards are designed to allow the mixing and matching of components from different vendors in an aircraft.

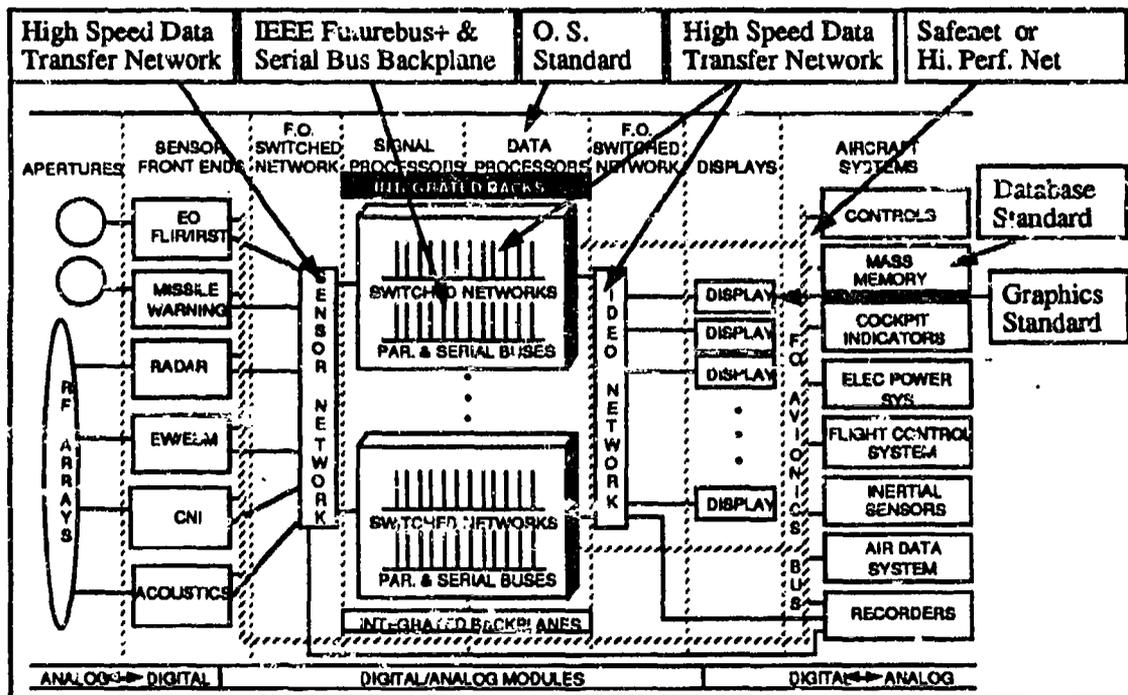


Figure 4-2.7.7.1 - NGCR Applied to Avionics

#### 4-2.8 Multi System Interconnects

These interconnects provide rack-to-rack or box-to-box communication. They are, in fact, local area networks (LANs). Two different complementary LANs, one medium speed and one high speed, are being standardized: SAFENET and a yet to be selected High Performance Network (HPN).

#### 4-2.9 Common Operating Environment (COE)

Commercial-off-the-shelf (COTS) computer resources are being examined by the U.S. Navy, and in some cases adopted for service use. As part of this effort, the services are defining a *common operating environment* (COE) for shipboard and ashore tactical computers. The use of the COE for airborne applications is not well defined at this point, but should be considered where applicable. The COE is a collection of specifications and designated standardized commercial hardware and software products. The U.S. Navy, the U.S. Air Force and the U.S. Army are in close, if not complete, agreement concerning the COE. Specified are the Sun workstation, UNIX operating system, POSIX, X-windows, MOTIF and certain other software packages. The Navy DTC-2 and TAC-3 computers are compliant with the COE.

#### 4-3 VMEbus

One of the principal computer backplane architectures used thus far in the COE is the industrial VMEbus. The VMEbus is one of several competing commercial standard microcomputer buses. As recently as 1991 it was considered state-of-the-art for applications requiring more computing power than the MS-DOS or Macintosh class of desktop computers. The VMEbus was designed to accommodate a wide range of processors, including most of the newer Complex Instruction Set Computer (CISC) and Reduced Instruction Set Computer (RISC) designs. Its heritage processor is the Motorola MC68xxxx family. Products based on VMEbus are widespread, and include such things as process control, analog and digital data acquisition, digital graphics and other applications where high speed processing is needed.

The origins of the VMEbus are found in the late 1970s when microprocessor integrated circuits first gained widespread acceptance. Motorola Semiconductor, Inc. introduced the VERSAbus architecture that accommodated their VERSAmodules, which were in turn centered around the company's MC68000-series 16-bit and 32-bit microprocessors.

VMEbus evolved as a European adaptation of VERSAbus. The original concept was developed at Motorola's European Microsystems Group in Munich, Germany, and was based on the standard Eurocard printed wiring boards. In October 1981 Motorola (USA), Mostek (USA), Signetics/Phillips (USA and Netherlands), and Thompson/CSF (France) placed their version of the VMEbus specification in the public domain in order to allow others to develop compatible products. The IEEE and ANSI began working on the specification in 1983, and by June 1987 the VMEbus specification was finalized.

The VMEbus standard, ANSI/IEEE 1014-1987, defines the digital, electrical and mechanical attributes of an interface backplane ("motherboard" in PC terminology) that allows assorted processor, functional, interface and interface control printed circuit cards to be plugged in and used together. A companion specification, issued by the International Electro-technical Commission (IEC standard 297), defines the mechanical attributes of the motherboard, backplanes, racks and card enclosures of the VMEbus system.

## 4-4 Commercially Supported Avionics Architecture Efforts

### 4-4.1 AEEC and ARINC

Since the mid-1980s, the Airlines Electronic Engineering Committee (AEEC), sponsored by Aeronautical Radio, Inc. (ARINC), has been investigating the application of new technologies to advanced avionics systems for commercial aircraft designs of the 1990s and beyond. This initiative involved meeting and consulting with representatives from academia, government and industry, to discuss how new technologies in the areas of microelectronics, fiber optics, open systems interconnection (OSI), communications, fault tolerance and software, could benefit future avionics generations. From this work, was born the concept of the Integrated Modular Avionics (IMA) system, which relies on a set of standardized hardware building blocks to perform specific avionics generic functions. Although the IMA architecture is still in development, AEEC/ARINC has published a top-level design guide as documented in *ARINC Report 651; Design Guidance For Integrated Modular Avionics*, dated 9 November 1991.

The IMA concept is centered around a powerful computing capability with an operating system that allows independent application software processing within a framework of strongly partitioned software modules. The computing capability is housed in a series of cabinets containing common, interchangeable hardware modules types that are interconnected by the ARINC 659 backplane bus. The cabinets and remaining avionics (sensors, displays, controls, actuators, etc.) are interconnected via the ARINC 629 global bus which forms the system backbone. Signal formats incompatible with ARINC 629 may interface through remote data concentrators, RF conditioners or cabinet hardware modules. The system standard High Order Language is Ada, although other languages can be integrated. The hardware provides a flexible electrical interface through the use of shared standard resources, while avionics/aircraft functions are nearly entirely implemented via the application programs (software modules). Figures 4-4.1a and 4-4.1b illustrate the typical IMA physical and functional system architectures.

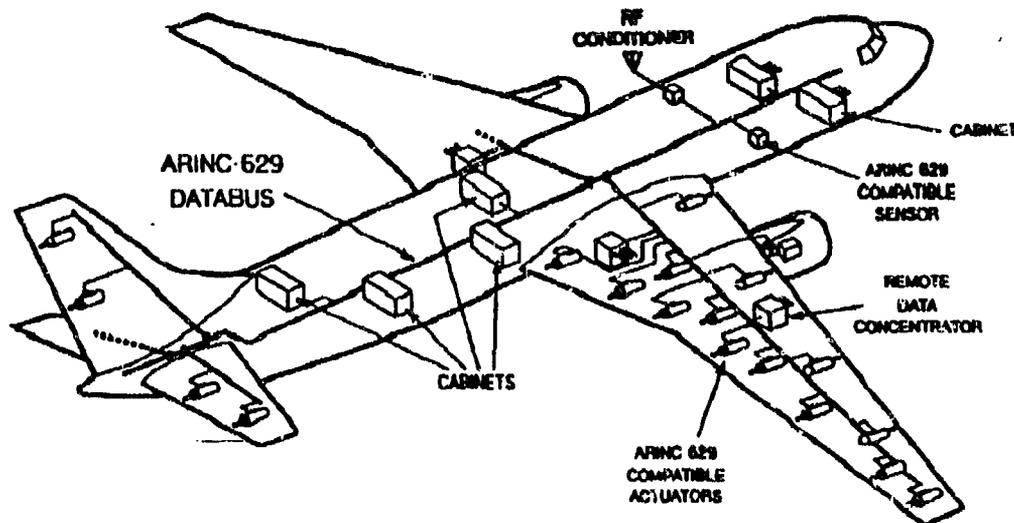


Figure 4-4.1a - Physical Architecture

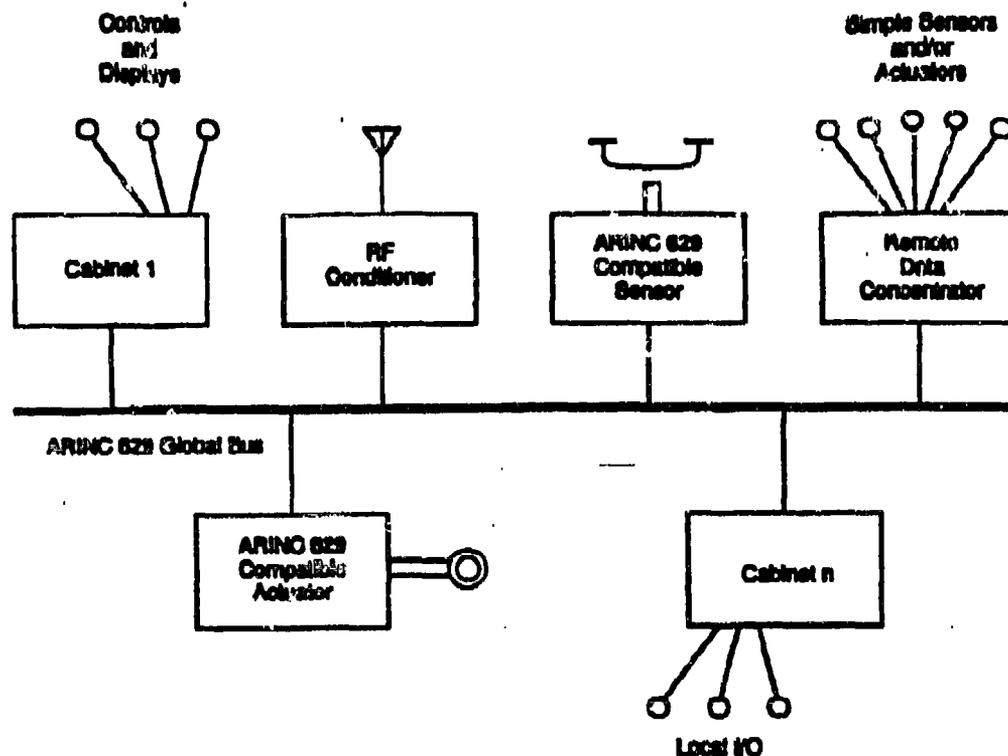


Figure 4-4.1b - Functional Architecture

Specific characteristics of the ARINC 629 data bus are defined in *ARINC Specification 629-2*, published 16 October 1991. Key parameters and characteristics include serial bi-directional data transfer at 2 MBPS, basic protocol operation in both periodic and aperiodic modes, autonomous multiple-access protocol, directed message handshaking or broadcast mode, 120 terminal capacity via non intrusive inductive coupling, and a twisted wire medium with provisions for fiber optics. Terminals (i.e., cabinets or other avionics systems and components) can be interconnected as desired throughout the aircraft, thereby eliminating the need for traditional electrical and electronics bays, and affording maximum flexibility with respect to considerations for functional and operational performance, environmental concerns, maintenance philosophy, other systems integration and growth potential.

Primary elements of the cabinet include the cabinet frame, ARINC 659 backplane bus assembly and hardware modules. The cabinet frame provides the mechanical and electrical environment to house the hardware modules, ARINC 659 bus assembly and additional aircraft interfaces. The ARINC 659 backplane bus assembly provides the intra cabinet and external interfaces for aircraft wiring, inter module traffic and power distribution. Figures 4-4.1c and 4-4.1d illustrate the generic cabinet physical and functional architectures. Key parameters and characteristics of the ARINC 659 bus include serial data transfer, 80MBPS throughput capacity, 32 bit data words, 2 bit message gaps, table driven protocol and fault tolerant processing. Candidate hardware module types include the core processor, standard I/O, special I/O, power supply, bus bridge, gateway and mass memory. To the greatest extent possible, IMA will seek to standardize on a set of common, interchangeable hardware module types that can be

configured within the cabinets such that the position, mix and number of modules can be tailored to suite specific cabinet functions. Physical and functional hardware module characteristics will be defined under separate ARINC specifications.

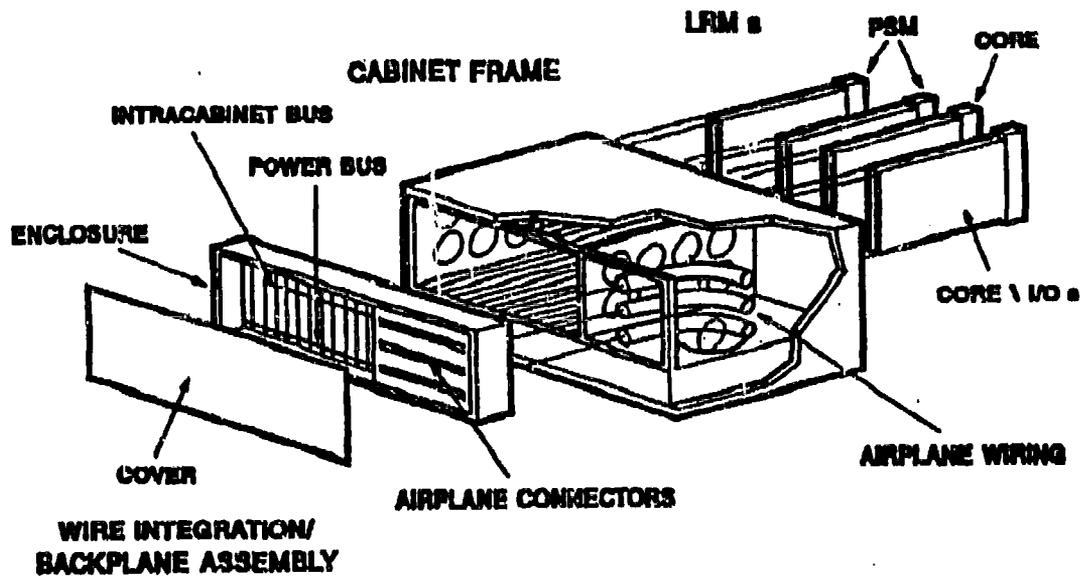
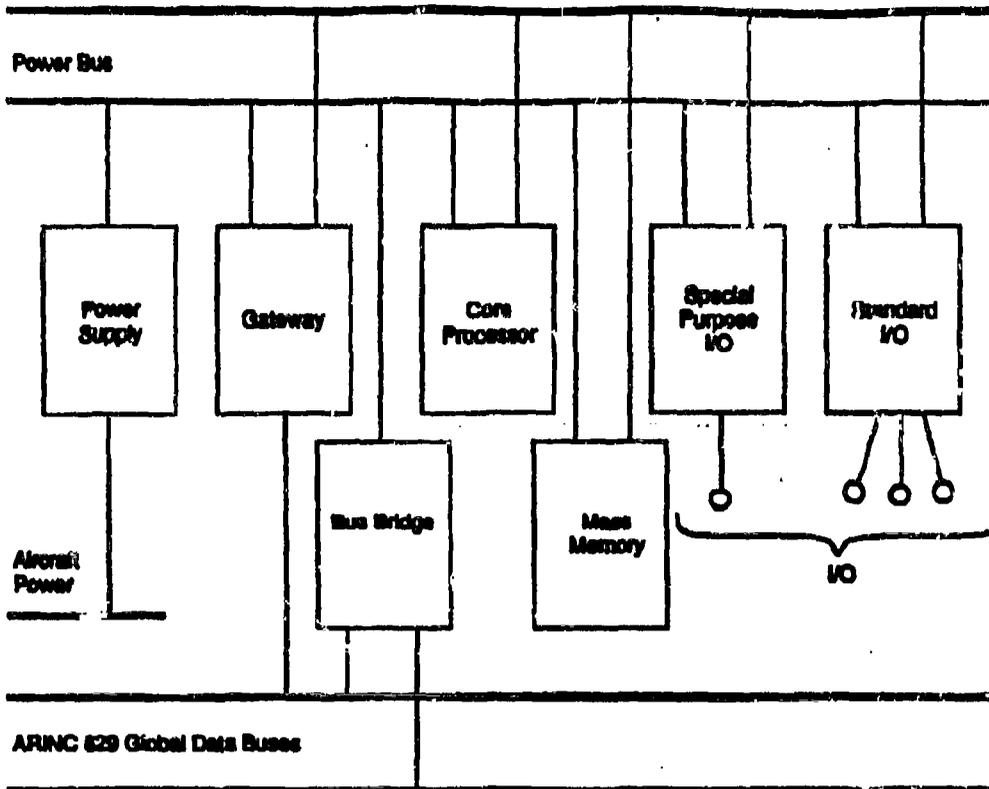


Figure 4-4.1c - Generic Cabinet Physical Architecture

ARINC 659 Backplane Bus



**NOTE:** The number and distribution of units in any cabinet depends upon the location, the number of functions, their type, and the level of redundancy/fault tolerance required.

Figure 4-4.1d - Generic Cabinet Functional Architecture

## **5-0 AVIONICS TECHNOLOGY DEVELOPMENT & REVIEW EFFORTS**

### **5-1 U.S. Air Force PAVE PILLAR and PAVE PACE Programs**

#### **5-1.1 PAVE PILLAR.**

The Pave Pillar Program was an Air Force funded development program to develop a standard modular avionics architecture. Various prime contractor and avionics systems contractors were teamed for this development. The major elements of the architecture were: a standard instruction set architecture using 16 bit VHSIC processors (MIL-STD-1750A), a common signal processor, a standard parallel backplane bus (PI Bus) and a test and maintenance (TM) bus, a global memory and a high speed fiber optic data bus using a linear topology. Other elements of the Pave Pillar architecture included a sensor data distribution network, a MIL-STD-1760 based armament interface and a MIL-STD-1553 based vehicle management system.

The Pave Pillar architecture was used as the baseline for the ATA/ATF/LH programs in their Demonstration and Validation (DEM-VAL) phase in an attempt to achieve common avionics as prescribed by Congress. The Navy A-12 program used this architecture in the EMD phase and modular hardware was developed for this architecture before the program was terminated. Likewise the Air Force F-22 and Army Comanche programs used this architecture. Unfortunately, the procurement process did not provide for the required configuration control between the platforms during development so that variants of the architectural standard produced hardware that is not interoperable. The Joint Integrated Avionics Working Group (JIAWG) has been attempting to force these developments to common interoperable hardware and a true open systems architecture.

#### **5-1.2 PAVE PACE**

The Pave Pace program is an Air Force funded development program supported to extend the modular integrated avionics architecture for application to advanced versions of the F-22 and the multi-role fighter (MRF). The program is looking at advanced data and signal processors, very high speed optical networks, advanced packaging, and modular sensor configurations. Specific efforts in integrated modular RF and EO subsystems are being explored.

### **5-2 Integrated Communications, Navigation, Identification, Avionics (ICNIA)**

The ICNIA advanced development program was chartered to improve the Navy's war fighting capability by enhancement of aircraft communications, navigation, identification (CNI) avionics. The program was designed to improve the performance and supportability of Navy tactical and tactical support aircraft by providing them with an Integrated Modular Avionics (IMA) architecture. This would be done by replacing the hundreds of different module types associated with current black box implementation of CNI functions with the greatly reduced number of unique module types (approximately 30) required for the ICNIA implementation. An important feature of the ICNIA architecture is that the total number of modules required is based on the maximum number of functions which must operate simultaneously in the mission timeline, rather

than the total number of functions which must operate over the period of entire mission. This approach allows the same individual modules to be used to implement different functions at different times in the mission. An important performance advantage of this approach is that if there is a failure in an individual module, then modules which are still functioning can be reallocated to that function if it is more mission critical than the functions that they are currently assigned to.

ICNIA was a joint Navy/Army/Air Force non ACAT program under which the Air Force was designated the lead service for tri-service common requirements, and the Navy and Army provided funding for Navy and Army unique requirements. The Navy sponsored the development of three Navy unique functions into the ICNIA architecture: Link 11, Link 4/ACLS, and FLTSATCOM. The tri-service effort supported the development of thirteen other functions: ARC-199 (HF), ARC-186 (VHF), Singgars (VHF ECCM), ARC-164 (UHF), Havequick (UHF ECCM), JTIDS, GPS, TACAN, MLS, VOR/ILS, Mode S, TCAS, and Mark XII IFF. The ICNIA Advanced Development Models (ADMs) incorporating all of the above functions were accepted in May 1991. ADMs three and four were integrated into the Integrated Electromagnetic System Simulation (IESS) facility at Wright Laboratories. The Demonstration and Validation of the ICNIA architecture was completed in FY-1992. Basic features such as the ability to operate multiple functions simultaneously, conduct built in test to identify defective modules, and reassign functions from defective modules to other modules in real time were all demonstrated. The flexibility of the ICNIA architecture to implement all of thirteen above-listed CNI functions was demonstrated as well as the ability to later implement an additional function (the Constant Source function). The ICNIA ADM development led to the development of a similar architecture for implementing a whole family of COMSEC units in a size and configuration commensurate with ICNIA modules.

ICNIA technology has transitioned to the Air Force Advanced Tactical Fighter (ATF) for full scale engineering development. This technology is available for transition to other new platforms or retrofit to existing platforms which are undergoing a major avionics upgrade. This technology does not lend itself to replacement of functions on an individual basis, but rather achieves its benefits when it is applied to a group of functions. This is due to the fact that the individual modules of the architecture are over designed for any one function since they must work generically for a number of functions. Thus, for a single function implementation, modules which are optimized for that function will always be smaller, lighter and cheaper. The payback (in terms of size, weight and acquisition cost) for the overhead involved in providing generic modules, which require a limited number of module types, occurs because of the ability to use the same module to implement different functions which are performed at different times in the mission, thus reducing the total number of modules which are required to equip a platform. Additional benefits of the ICNIA architecture occur in the life cycle support of the avionics system since the greatly reduced number of module types greatly reduces sparing requirements and function specific diagnostic requirements.

### 5-3 Integrated Electronic Warfare System (INEWS)

The (INEWS) program is the result of the merger of the Air Force's New Threat Warning Receiver and the Navy's Integrated Electronic Warfare System (IEWS) technology development programs. The joint Air Force/Navy INEWS technology development program started in 1983 to develop a modular integrated Electronic Warfare suite applicable to the advanced technology aircraft. The INEWS architecture baseline was derived from the Air Force's PAVE PILLAR architecture.

During the Concept Exploration and Definition phase and Demonstration and Validation phase 1A, five contractor teams were active players in the technology development effort. These teams included Sanders/General Electric; TRW/Westinghouse; Loral/Hughes; ITT/Litton Amecom; Raytheon/Northrop. The INEWS phase 1A program was an continuation of the Concept Exploration and Definition phase where the contractor teams refined their concepts and detailed the areas of major risks. Down selection to two contractor teams was accomplished in early calendar 1986. The winning INEWS teams were Sanders/General Electric and TRW/Westinghouse.

At the beginning of the Demonstration and Validation phase 1B program recipient platforms for the INEWS technology had not stepped forward and identified themselves, thus the missions, observability levels, etc. were generic and included both high and low altitude flight profiles. Each of the INEWS concepts were driven by aircraft survivability numbers and thus included more functionality than was normally considered for a tactical aircraft. Weight was a driving concern, however and 1200 to 1800 pound budget was allocated for each of the individual INEWS concepts. Volume was specified but it was not a driver. A cost goal was imposed which was in then year dollars equivalent to a ship set cost of \$3.2 M. During the later portion of the INEWS phase 1B Demonstration and Validation (DEM-VAL) program, the ATF program office embraced the INEWS concept of modular integrated EW avionics and took control of the two INEWS DEM-VAL contracts and subsequent activities. The INEWS program schedule was revised to more closely match the ATF DEM-VAL schedule.

The INEWS statement of work was modified to provide for the fabrication of INEWS Advanced Development (ADM) hardware and software. The ADMs were not required to be packaged in SEM-E modules, even though some functions were in SEM-E modules and the ADMs were not required to be entire systems.(i.e., complete RWR system). The ADMs could be a thread within a subsystem but was required to be sufficiently complete to prove the function could be performed. An example is the RWR function where all of the necessary sub functions were included but not all of the receivers (where multiple receivers were required) were not fabricated. Also, only one of each antenna type was fabricated.

The ATF SPO after taking over management of the INEWS activities directed its aircraft prime contractors to team with the INEWS teams to develop the Electronic Combat (EC) suite for the ATF. As a result of this direction, the YF-23 prime teamed with TRW/Westinghouse and the YF-22 prime teamed with Sanders/General Electric. After the teaming arrangements were completed, the INEWS program per se became nonexistent and it became an EC development effort for the ATF program. Much of the functionality originally included in the INEWS concepts disappeared due to weight, cost or other driving constraints on the ATF concepts. In early 1990, the Air Force down selected ATF contractors to the Lockheed F-22 team which in effect relegated all INEWS

activities to the Sanders/GE team. TRW/WEC are involved with the RAH-66 EC suite but this is an insignificant effort compared to their original INEWS concept.

A follow on program, SEEK SPARTAN, was started by the Air Force which investigated retrofit applications for INEWS technologies on existing Air Force tactical aircraft such as the F-15, F-16 and F/B-111. SEEK SPARTAN was terminated after approximately two years of operation. The Navy is continuing to evolve INEWS functionality and technologies not embraced by the F-22. The Navy has placed particular emphasis on evolving Missile Approach/Launch Warning Systems targeted for retrofit on current or emerging Naval Tactical aircraft.

#### **5-4 Advanced Avionics Subsystems and Technology (AAS&T)**

The AAS&T is a multi-faceted Navy program for maturing the advanced Integrated Modular Avionics (IMA) concepts developed under both this program and the Air Force's PAVE PILLAR and PAVE PACE programs, and which have been adopted by the Joint Integrated Avionics Working Group (JIAWG). AAS&T focuses on the common Advanced Avionics Architecture (AAA) directed by Congress for all "advanced aircraft." The program thrust is Navy peculiar applications of advanced IMA for current and future Naval aircraft. Current task areas of the project are: shared aperture systems, digital technologies, avionics photonics, avionics packaging, and situation assessment and awareness.

The primary payoffs sought are: (1) reduced risk for Navy DEM-VAL and EMD programs; (2) increased readiness and improved mission and weapon systems effectiveness; (3) increased aircraft survivability; (4) enhanced technological readiness; and (5) leveraging of DoD and industry initiatives and developments to fulfill Navy needs at the lowest possible cost.

The nature of this program is such that medium to high risk avionics developments are demonstrated which have a potential for providing high payoffs. The risk to future platform developments are minimized by maturing and proving the technology prior to avionics integration.

##### **5-4.1 Documented OPNAV Requirements**

NAPDD No. 083-50 documents requirements for: (1) Reducing the risk for DEM-VAL and EMD aircraft and advanced avionics programs; (2) Supporting Advanced Avionics Architecture (AAA) in Navy aircraft; (3) Reducing avionics systems cost; (4) Increasing aircraft survivability and lethality; and (5) Leveraging DoD initiatives plus service and industry investments in AAA.

##### **5-4.2 Description of Project**

Current principal goals of the AAS&T project are: (1) Demonstration of risk reduction for next-generation Navy platforms by demonstration and evaluation of critical advanced avionics subsystems and architecture concepts and prototypes; (2) Demonstration of enabling avionics technology for multi-mission aircraft; (3) Application of open architecture and leveraging of commercial software and technology; (4) Leveraging and inserting maturing technologies into Navy avionics; (5) Improved surveillance, detection and classification of targets and threats; (6) Increasing avionics commonalty across tri-service and Navy platforms; (7) Real-time scene generation.

situation assessment and mission analysis for manned and unmanned vehicles; and (8) Rapid technology transfer to Unmanned Air Vehicles (UAVs) via the Joint Defense Force (JDF).

#### **5-4.3 Rationale for The Project**

The rationale for this project is best outlined in the following five statements:

1. For some forty years, the threat faced by each successive generation of Naval aircraft has grown increasingly more lethal.
2. Our response to this steady trend has been to make the avionics of each successive generation of aircraft successively more complex. Measured in terms of circuit element count, the avionics of an aircraft such as the AX will be some eight orders of magnitude more complex than a 1950s vintage F-9F.
3. With this growth in capability have come penalties: the cost of avionics has grown from less than 1% to be some 40% of the overall cost of the aircraft; the reliability of avionics has come to be a prime factor both in availability and support costs. We are rapidly approaching the point where we cannot afford needed capability.
4. In response to this trend, the Air Force and the Navy have developed the concept of integrated modular avionics architectures; architectures, which in contrast to previous black box concepts, are built up from standard functional modules, heavily based on the use of VHSIC and MIMIC technology, programmed in Ada for reusability and supportability, extensively instrumented for built-in test and diagnosis, interconnected by the high bandwidth fiber optic data buses necessary to implement to complex functions such as sensor fusion, and arranged in a pooled spare concept for fault tolerance.
5. But no platform can risk commitment to such a complex concept with so many technological factors not yet proven. The thrust of this effort is the development and demonstration of the concepts and technology necessary for low-risk insertion of these advanced architectures in next-generation Navy platforms.

This effort is funded in category 6.3A in that it evaluates and demonstrates emerging technologies (VHSIC, MIMIC, fiber optics, integrated architectural concepts) for the purpose of DEM-VAL and EMD program risk reduction.

#### **5-4.4 Tactical Utility**

The potential operational employment of products from this project span the entire spectrum of core and mission avionics. The IMA and AAA efforts strengthen reliability and support ability factors across the board. Examples of war fighting high payoff task areas are: (1) Situation Assessment & Awareness, to provide improved air crew and system cognitive performance in high stress, combat environments, and (2) Shared Aperture Systems, to enhance aircraft stealth and upgrade avionics survivability and multi-mission capability.

#### **5-4.5 Resource Savings - Manpower, Logistics Support, Etc.**

The IMA concept allows module interchangeability such that a few unique modules can be used for the whole of the avionics computer suite (processor, I/O and memory modules). Module spares would not be required in large numbers, and lower maintenance staffing and training would be required to repair or replace the modules. The AAA concept can improve fault tolerance, accommodate common subsystems and buses, permit reduced sparing and a two-level maintenance, and simplify modifications and mission reconfigurations.

#### **5-4.6 Countermeasure Resistance**

The project has efforts in photonics components that are resistant to enemy ELINT and to failures due to EMI or EMP. Advanced algorithms under development and planned will enable radar clutter discrimination between rain and enemy chaff and increase radar detection and track initiation ranges. One of the task areas, Shared Aperture Systems, is involved in demonstrating electronic beam steering of the antenna systems. An intended function of beam steering is to be able to reshape the antenna's radiation pattern such that very low gain nulls are directed towards a jammer significantly reducing the effects of the jammer through its penetration via the antenna system. The Low Probability of Intercept (LPI) Communications task in the Digital Technologies task area has an indirect bearing on jamming by developing a communications wave form that has very low detectability. This has the effect of not alerting a jamming source to the need to initiate jamming.

#### **5-4.7 Major Tasks and Subsystems**

##### **5-4.7.1 Shared Aperture and Multi Function Systems.**

This category includes the Special Airborne Antenna System (SAAS), which is jointly sponsored by the Air Force and Navy INEWS program, and the Airborne Shared Aperture Program (ASAP) which includes joint participation by the Air Force and the Naval Sea Systems Command.

SAAS is planned to cover all CNI applications operating from 2 MHz to 6 GHz. It is funded by this project, the Navy INEWS program and the U.S. Air Force. SAAS is being considered for application on the Navy and Air Force AX aircraft, the F/A-18E&F, and as an upgrade to the F-22. SAAS will operate with today's federated CNI systems as well as the integrated systems (ICNIA, IMA, INEWS, etc.) in the future.

ASAP is designed to serve as the aperture for multiple radio frequency subsystems operating in the C, X, and Ku bands, including radar, ESM, ECM, and communications. ASAP will utilize technology currently under development in the DoD/tri-service MIMIC Program and is leveraging chip and brassboard demonstrations conducted under the MIMIC contracts. Both the Air Force and NAVSEA have expressed interest in the ASAP concept and may jointly sponsor applications to their own platforms in the future. ASAP is designed to operate interactively with the subsystems served by SAAS such as ICNIA and INEWS. ASAP includes the design and demonstration of a system resource manager which will automatically select the proper subsystems to successfully complete a mission segment.

The data from the ASAP and SAAS apertures will be directed to the signal processors via the Sensor Data Distribution Network (SDDN), which is also under development in this project, where it will be processed in signal processors using many of the tools developed under this project as well.

Receivers and other RF modules behind the shared apertures will eventually benefit from a planned Air Force advanced technology transition demonstration program entitled Integrated Sensor Systems which is to demonstrate a family of common RF modules can be used to replace most of the custom modules found in today's avionics.

The Navy and Air Force plan to work together on both the shared aperture and ISS efforts. The processors which are used in conjunction with ASAP will make use of many of the advanced algorithms (e.g., distinguishing rain from chaff, super range detection and track, etc.).

Finally, both the ASAP and SAAS apertures will serve as data sources for the Advance Airborne Situation Assessment System (AASAS) which will use a combination of sensor information and stored data to render real time perspective scenes with threat overlays to assist in situations requiring closed cockpit flight, covert penetration, precision strike, battle damage assessment, etc. AASAS is also in development under this project.

#### 5-4.7.2 Digital Technologies.

This class includes the advanced algorithms work discussed above as well as the evaluation of the special processor requirements which may be unique to shared aperture systems such as ASAP. This task also includes the development, evaluation and demonstration of fault tolerance technology for integrated modular avionics systems. Efforts under this task serve in many ways to connect the needs of naval aviation to the on-going developments of the Air Force F-22 and Army LH programs through interaction in the JIAWG process. Where necessary, efforts go beyond the current JIAWG approach where new or special requirements are foreseen such as in ASAP or SAAS applications to the AX. Since it is currently a premise of this project that the Navy will not develop new processors but will adapt those developed by industry or other programs, this task is by necessity highly connected to most digital processing efforts including the F-22 CIP, NGCR, AN/AYK-14, AN/UYS-2, and others.

#### 5-4.7.3 Avionics and Photonics

This class includes the development and demonstration of fiber optic components and technology for applications including sensor data distribution, SDDN as discussed above, MIL-STD-1553 and MIL-STD-1773 Multi-speed data bus for retrofit applications, high speed linear and ring buses, and optical back plane buses for communications within an IMA rack. By the very nature of these digital data communications efforts, they are connected to the heart of virtually all future avionics systems. The work is coordinated both within the government working groups (e.g., JIAWG) as well as national standards groups (e.g., SAE, IEEE). Efforts and products are connected to the F-22, AX, F/A-18E/F, UAV, NGCR and other programs.

#### 5-4.7.4 Avionics Packaging

Like photonics, packaging is central to the very concept of IMA. Efforts are tightly coupled to the Navy Standard Hardware And Reliability Program (SHARP) and the tri-service JIAWG coordinated IMA efforts. The Navy is viewed as the leader in demonstration of the advanced packaging technology such as Standard Electronic Modules (SEMs), integrated racks, advanced cooling, etc. and often leads efforts jointly funded by the other services. The efforts in this task are connected to the F-22, LH and AX avionics efforts and virtually any future IMA system. Efforts are connected to other tasks within the project such as fault tolerance, photonics, ASAP, and others which will be packaged utilizing the packaging concepts of this project or which provide components for this task (e.g., optical I/O components for an optical interconnect system, bus protocol circuitry, etc.).

#### 5-4.7.5 Situation Assessment and Awareness

This effort will eventually connect with other efforts in this project such as discussed above when it is considered feasible to include the subsystems required for enroute mission planning, mission rehearsal, real time perspective scene with overlays, advanced covert penetration, and post-mission analysis in the IMA system. Present efforts are concentrated on demonstrating the algorithms and the visualization technology required for such a capability in a manner which leverages the revolution in commercial processors and software. It is an open architecture approach which will connect to efforts sponsored by the training systems program, mission planning program, UAV program, and human factors program which are currently sponsoring interrelated tasks. These connections will be increased in the future and will be expanded to incorporate efforts sponsored by SDIO and the USAF (e.g., TENCAP program). A major objective is to demonstrate a single thread approach to hardware and software for mission planning, training, mission rehearsal, in-flight assessment and awareness, and post-flight assessment and planning. Future hardware from this task will utilize the hardware demonstrated elsewhere in this project (e.g., packaging, data bus, ASAP, etc.).

#### **5-4.8 Key Technologies**

Key technologies for future avionics systems include photonics, MIMIC, VHSIC, advanced composite materials, two-phase immersion cooling, VHSIC Hardware Descriptive Language (VHDL), behavioral simulation and modeling, signal and image processing, data fusion, radar, ESM, ECM, antennas, control electronics, LPI, software engineering, and high performance computing.

#### **5-5 Integrated Avionics Development Program (IADP)**

The IADP Avionics program was established during FY 1992 to establish flying test bed aircraft, including an F-18, A-6, and P-3C, which can be configured to evaluate advanced avionics developed for new aircraft (e.g., F-22, LH, A-12, B-2, etc.) for retrofit into Navy platforms. The program is centered at the Naval Air Warfare Center — Aircraft Division (Patuxant River, MD), and is managed by the Navy INEWS Program Officer within PMA-253. Programs will have easy access to the results of the evaluation of avionics hardware and software developed for other programs, as non-development items, or under other means.

## 5-6 Ultra Reliable Digital Avionics (URDA)

Although the name of this USAF/Army funded effort suggests that it seeks to design ultra reliability into digital avionics, the primary task is to develop an extremely high performance digital processing module that incorporates new interconnect, packaging and stress management technologies. The deliverables are targeted for the F-22, RAH-66 P<sup>3</sup>I and AX. Because the packaging is liquid flow through SEM-E modules with F-22 connectors, little truly new module packaging technology is being developed or tested. Also, polymer board-to-board interconnects and die stacking will be used, which again are not new technologies.

There are two contractors on the URDA program: Texas Instruments and AT&T. Both will be providing the same deliverables, including a demonstration computer constructed of the new processing modules, and the software development environment. TI is repackaging its Aladdin computer onto an F-22 SEM-E module. Multiple Aladdin processors will be included on each SEM-E card, along with PI-bus and TM-bus interfaces. The processors will be the R4000 and TMS320C309 devices. The AT&T processor design will use MIPS R4000 and TMS320C40 processor chips interconnected in a hypercube topology, packaged on F-22 SEM-E modules with PI-bus and TM-bus interfaces. Both contractors will develop work station environments that stress supportability.

## 6-0 SYSTEM BUSES, NETWORKS, AND INTERCONNECTIONS

Traditional computer architectures in commercial and military systems are being supplanted with a variety of new architectures that support two key aspects of modern processor development: very high memory access speeds and processor parallelisms. These developments have impacted architecture primarily through the extensive development of distributed memory systems. The result of these changes drives the connection between processors on a board (for example, the interconnection between multi-chip modules (MCMs) on a board), the connection between boards (backplane interconnects, for example, PI Bus and Futurebus+), and the connection between workstations (commercial) and equipment racks (military).

Much of recent improvement in single processor performance (i.e., Reduced Instruction Set Computers - RISC) has come from super scalar techniques within processor design. This represents parallelism that may be achieved without cooperation of the application instruction execution. Workstation performance through the end of the decade is predicted to increase 60% to 100% per year (doubling every 20 months or so). Workstation vendors expect to maintain this rate of growth by extending the degree of parallel execution into the application software domain - that is, multiple processors (perhaps 26 to 28 by the end of the decade) on a workstation-sized board. This approach to complete processor parallelism calls for application of new technology in the above mentioned three categories of interconnection.

### 6-1 Interconnect Buses

#### 6-1.1 High Speed Data Bus (HSDB)

The HSDB standard was developed by SAE AS-2, Interconnect Networks Committee -- Avionics Systems Division, with DoD input and direction. The HSDB has separate token passing protocols for both linear and ring topologies. Both buses are designed for either fiber optics or electrical cable implementation, but only fiber implementation is being seriously considered. The SAE Linear HSDB served as the basis for the JIAWG HSDB standard, which is very similar to the commercial standard, but is customized for needs of JIAWG class aircraft. Linear HSDB is functionally part of the PAVE PILLAR architecture.

The SAE has developed a series of standards for high performance, fiber optic serial data buses. These buses are based on token passing protocols for the arbitration sequence. Such communications protocols are intended to replace buses such as MIL-STD-1553 in the near future.

There are two standards for High Speed Data Bus (HSDB) based on either a ring or a linear topology. The linear topology has been adopted as a JIAWG standard. This bus was simulated using gate level models of bus interface units from IBM, Texas Instruments, and Unisys, in order to isolate specification ambiguities and inter operability problems. The initial hardware development occurred for the A-12 Demonstration and Validation (DEM-VAL) program. F-22 and RAH-66A are also using this bus. The primary suppliers of bus interface units for linear HSDB include Harris and IBM.

## 6-1.2 MIL-STD-1553 and MIL-STD-1773 Serial Linear Control Buses

MIL-STD-1553 (with its MIL-STD-1773 offspring) is the only approved, tested and verified data bus open standard currently available for military avionics systems integration applications. This standard defines a "Command/Response" protocol and a "linear bus" topology that supports a Bus Controller and up to 31 slave Remote Terminals (each with up to 30 "sub-address" functions). This protocol allows data exchange (both transmit and receive) only when and as ordered by the Bus Controller. Broadcast data transmission is accommodated by the standard, but is precluded from Air Force applications by Notice 2 to 1553B.

System survivability and message integrity are provided for by the use of backup Bus Controllers, Dual or Triple Redundant bus implementations and verification of message validity by the recipient's Status Word response at each message event. Message validity is based on the correct number of words received (each word with proper Manchester Sync bit and data bit coding format, correct bit count and valid "parity") with no gaps in the data word sequence.

These two standards utilize self clocking Manchester II Biphase Level data coding, at a 2 MegaBaud modulation rate, to transmit data at a 1 Megabit gross bit rate. The media is defined as shielded twisted pair wire in 1553 and as optical fiber waveguides in 1773, with wave form degradation and response time limitations appropriate for the limited extent of an airborne platform.

The protocol was designed for and is ideally suited to "Command and Control" functions, but the protocol and the low modulation rate limit its data transport capability. Word format overhead limits the peak data throughput to 800 Kilobits per Second. Message formats consist of a Command or Status word accompanied by 0 to 32 data words. The message length, Command/Response overhead words and response times reduce the bus average data throughput to two or three hundred Kilobits per Second.

The need for greater data throughput and more than 32 terminals in a system has resulted in evolutionary system growth. Platforms with 5 or more busses are not uncommon. These needs have led to the development of features like dynamic redefinition of sub-addresses and hierarchical bus networks. The hierarchical network includes a "global" bus connected to one or more lower level busses by a "Gateway" terminal that is an RT on the global bus and the BC on the lower level bus. Each lower level bus may in turn be gatewayed to its own set of lower level busses. Thus a multi-level structure is produced that can expand the number of communicating terminals/function, almost without limit.

The price is additional message overhead and transit delays, that progressively reduce the already limited data throughput capability of 1553, for cross-level message traffic.

MIL-STD-1773, as presently embodied, extends all protocol, topology and message format features of MIL-STD-1553. It only addresses those issues resulting from the realization of the bus media in optical fiber waveguides, rather than wire.

Further more, 1773 does not include the detail specification aspects of 1553. A supporting General Specification and a type or application Detail Specification are required for generation of interoperable 1773 bus hardware. The SAE AS-3 Avionics Systems Group, which prepared the original 1773 draft standard, now has prepared a

draft general development specification in support of MIL-STD-1773, which is to be circulated for approval as a Mil Spec.

The only current advantages of 1773, over 1553, relate to the non-conductive nature of the media and the elimination of the need for electrical shielding and its attendant weight, to achieve EMI/RFI immunity. No present use is made of the high frequency/Baud rate capabilities inherent to optical transmission. AS-3A has currently begun work on a proposed revision A to MIL-STD-1773 that will exploit this capability to provide a further growth path for 1553 based systems.

The purpose of MIL-STD-1773A will be two fold. First to define an optional alternative or replacement modulation format to the existing unipolar optical Manchester that retains the implicit "Message Envelope" information that is available with the Bipolar electrical modulation of 1553, but is lost with the Unipolar optical modulation of 1773. The elimination of the two microsecond delay required to detect the end of a message by a 1773 to 1553 converter will enable two 1553 terminals to communicate through external converters over 1773 bus physical media, within the required 14 microsecond minimum response time-out period, assuming that the terminals meet the 12 microsecond maximum response time limitation.

The second purpose is to incorporate "Multi Speed Data Rate Transmission" (MSDRT) into the standard. One proposed implementation provides enhanced throughput by transmitting 1 to 8 blocks, of 8 data words each (1 to 256 total words per message), with eight Megabit per second Manchester coding, between enhanced terminals. Command/Status words between all terminals and data between unenhanced terminals continue to be transmitted in the existing one Megabit 1553/1773 format, for complete backwards compatibility with unenhanced 1773 terminals on the same bus. The enhancement provides peak data transfer of 6.4 Megabits per second and average throughput of several Megabits per second, capacity adequate to replace six or more existing 1553 busses. The proposed general specification for this Dual-Speed enhancement to MIL-STD-1773 is available in the "MIL-STD-1773 Users Handbook", available from the SAE as document number AIR4508.

### **6-1.3 Fiber Distributed Data Interface (FDDI) / SAFENET**

FDDI is an emerging standard that offers a network with a highly reliable data transfer, active link monitoring, station management, large bandwidth capabilities (100 MBPS transmission rate), survivability features, and the advantage of fiber optics. An FDDI network will prove beneficial in a variety of application areas. FDDI implements a dual counter rotating ring topology and uses a token access method for packet transmission.

#### **6-1.3.1 FDDI in Avionics**

SAE's AS-3, Avionics Systems Group, is tasked to develop and review fiber optic and photonics issues. Within AS-3 there is a new working group formed to develop Military Fiber-Optic Transmission System (MFOTS), and it is tasked with providing recommendations dealing with high-speed fiber-optic networks for the next generation avionics systems. This effort includes developing a system specification for an FDDI-based communications network military aircraft.

SAFENET (Survivable Adaptable Fiber-Optic Embedded Network) is an FDDI network, defining the lower-layer protocols, and a set of middle-layer protocols (transport

and network layers of the ISO/OSI protocol stack) termed the Light Weight Protocol (LWP). This LWP resulted from the realization that expedient, real-time data delivery would be necessary in a tactical situation. Low latency and high throughput must be achieved between equipment, especially when relating to weapons system. Devices capable of this high-speed data delivery would be able to use FDDI's 100 MBPS data rate effectively. SAFENET was developed to meet the shipboard environment.

#### **6-1.4 Asynchronous Transfer Mode (ATM) Local Area Network (LAN)**

ATM was originally developed by the telecommunications community as a logical layer protocol based on bandwidth partitioning for the transmission of large amounts of data including real-time audio, computer data, images, and video on shared media point-to-point switched networks. ATM transfers digital information in the form of consecutive cells of constant length, consisting of a five byte header followed by 48 bytes of information. The header defines a virtual path and a virtual channel as well as other network management functions. The ATM protocols allow a node to establish static or dynamic connections with many other nodes. Although ATM is optimized for virtual connection oriented services, it can be used for connectionless services as well. ATM can be mapped on top of various physical layers such as Sonet, Fibre Channel's 8B/10B physical layer, or FDDI's 4B/5B physical layer. These physical layers support serial communications over wire as well as fiber optics cables at bit rates from 155 MBPS through 2488 MBPS.

ATM services best suit longer distance communications environments, such as local area networks (LANs), metropolitan area networks (MANs), and wide area networks (WANs). In addition, ATM's isochronous services support time critical data transfers including audio, video, and multimedia. Although not optimal, ATM can also be used for input and output to computers.

The primary champions for ATM are a group of companies who have banded together in an organization called the ATM Forum. Their purpose is to further develop ATM as the primary telecommunications protocol to meet all computer related telecommunications needs. The ATM Forum numbers about 200 companies including the big names in both computers and telecommunications. No national standards body, such as IEEE or ANSI has as yet become involved with ATM. However, at this time it seems likely that ATM will become the predominate LAN and long distance telecommunications protocol.

#### **6-1.5 Fibre Channel**

Fibre Channel was developed by ANSI as a transport protocol for the predictable transfer of large blocks of data such as those used in file transfers, disk and tape storage systems, communications, and imaging devices. Fibre Channel transfers asynchronous information in the form of variable length frames, consisting of a 24 byte header followed by up to 2112 bytes of information. Fibre Channel provides bi-directional connections and support for packet switching, connected operations, and connectionless operations. Transmission is isolated from the logical protocols so that a variety of implementations are possible. Currently, point-to-point links and switched topologies are defined. The Fibre Channel physical layers can support serial communications over copper and fiber optics cables at bit rates from 132.8 Mbaud to 1062.5 Mbaud.

Fibre Channel is optimized for input and output as well as communications between nodes. In addition, Fibre Channel is also suited for data flow between

processors since it can provide connection services and can guarantee bandwidth between two nodes. Although not optimized because of a large header, Fibre Channel can also provide processor to memory services. Presently Fibre Channel is competing with ATM for connector space on the backs of work stations. While Fibre Channel is a more mature LAN standard, its protocol has not been extended into the telecommunications arena as has the ATM protocol. For this reason its success may be limited to being a fast flexible I/O channel to super computer tapes and disks.

#### **6-1.6 Other Network Schemes**

ARINC provides standards for the Airline Industry Association, including several data bus standards. ARINC-429 is a serial, point-to-point data link which is clocked at either 100K bits per second or 18.6K bits per second. This bus is used in the airline industry for interconnecting radio navigation and flight management systems. A digital interface format has been published which provides a standard communication protocol for such applications. Many military transport aircraft use this data bus. In addition, Navy aircraft which use over-water navigation aids such as Omega often use this data bus. ARINC-629 is a very similar data bus which is now emerging in new commercial transport designs such as the Boeing-777. This bus is intended to provide more performance and functionality than ARINC-429. As such, it will probably appear in future military transport and special mission aircraft.

#### **6-1.7 Electronics Industries Association (EIA)**

EIA publishes many data bus specifications that are used throughout the computer industry, such as RS-170, RS-232, and RS-422. Such buses are also found on commercial transport aircraft and military aircraft. RS-232 is typical of such bus types. RS-232 is a serial, point-to-point link which can be clocked at up to 20,000 bits per second. Normal transmission rates are the familiar modem speeds, such as 4800, 9600, and 19200 baud. The bus protocol defines a primary bus and secondary bus, as well as control signals. Various modes of operation are defined, such as full duplex, in which a signal is transmitted and then echoed back to the originator, or half duplex with no echo. It is also possible to transmit on the primary bus and receive on the secondary bus, or vice-versa. RS-232 is used in commercial computer applications to connect terminals to computers. Aircraft systems normally use the RS-232 interface as an interconnect to external equipment, such as the console.

RS-422 is an update to RS-232. RS-170 is a serial channel used to control video devices and is found on many aircraft.

### **6-2 Backplane Buses and Networks**

A consequence of modularity in a digital computer is the use of a backplane bus into which modules can be plugged. It is common practice to associate the name of the backplane bus with the entire system, so a "VMEbus" computer is one in which the VMEbus backplane is used, and which will accept plug-in modules that are compliant with the VMEbus standard. There are several backplane bus architectures that have potential for naval avionics use.

### 6-2.1 Pi Bus

Pi bus was originally developed by the VHSIC program as a standard computer system interface. After completion of the VHSIC program, the standard was transferred to the JIAWG in order to support the primary user programs (A-12, F-22, and LH). As part of risk reduction activities for these programs, gate level modeling of first generation Pi bus controllers from IBM, Texas Instruments, and Unisys was performed to identify inter operability problems. Numerous hardware demonstrations were also completed. The inter operability trouble reports were used to produce an extensive update to the VHSIC Pi bus specification. Subsequent to completion of F-22 DEM-VAL phase, the SAE formalized the specification (the requirements are now the same as those contained in the JIAWG specification) and established a User's Group. The SAE Pi bus specification is now in the final stages of balloting. A Pi bus handbook was started earlier this year.

Pi bus is a synchronous 12.5 MHz parallel bus designed to support message passing and fault tolerance. Two bus widths are supported, 16 bit Error Detecting (ED) and 32 -bit Error Correcting (EC). The 16 ED bus is normally implemented as a dual redundant pair. The usual method of communication is one of several block transfer modes, which can move up to 65,535 data words to single or multiple slaves via either physical or logical addressing. A bus interface message is also defined which allows communication with the Pi bus controller register space. Also, a parameter write message is defined which allows a low latency, three word transfer. Finally, a datagram is defined which is a non acknowledged block transfer used for low latency block transfers.

A single 16ED or 32EC bus requires 58 signal lines. A mixed mode of operation, in which 16 and 32 bit modules are connected within the system, is also supported, which requires 60 signal lines. The protocol supports up to 32 modules per backplane. However, due to module stub lengths and electrical characteristics of the backplane, systems often contain approximately 16 modules (or fewer).

Extensive error handling and real-time support are provided in the Pi bus protocol. A "suspend/resume" feature allows messages which cross minor frame boundaries to be interrupted. An abort sequence can be used to halt a babbling module. A 48 bit real-time timer is defined. The Pi bus specification also describes numerous other error conditions and handling procedures.

Pi bus is now reaching full maturity due to extensive simulation, hardware demonstrations, avionics DEM-VAL, and operational use. It was used for A-12, YF-22, YF-23, is now found in F-15E, and will soon be operational in EF-111, F-16C Upgrade, F-22, and LH. Bus loading on each of these systems is quite low (<8%, except for current F-16C Upgrade estimates around 13%), which allows room for future growth. When additional performance is needed, faster transceivers, denser logic, use of a 25 MHz. clock, and other factors can be used to significantly increase performance. Pi bus is also under consideration for commercial avionics applications, which will broaden its technology base.

Pi bus was designed to provide a simple, optimized mechanism for transferring large blocks of data within a system. The designers intended the bus operation to be a "send and forget" asynchronous interface, mostly used in loosely coupled, multi computing computer architectures. For this reason, Pi bus is clearly not adequate as a real time memory interface in tightly coupled systems, nor was it designed for such applications. Therefore, within the architectural paradigm for which Pi bus was designed, it offers a low risk, highly fault tolerant interconnect system.

### 6-2.2 Test & Maintenance (TM) Bus

The Test and Maintenance (TM) bus is another VHSIC Phase Two Inter operability standard. Therefore, its history and user base closely parallels that of the Pi bus.

The final VHSIC version 3.0 is used in numerous systems. However, in 1990 it became clear that some additional features were needed to support system diagnostics in F-22 and LH. The primary changes in systems requirements were an increase from 32 to 251 modules addressed (256 logical address including broadcast and multicast) and a defined sequence for bus mastership (the VHSIC 3.0 specification is, in essence, a central arbiter). As a consequence of these new requirements, there were significant changes to the VHSIC TM bus clocking, electrical characteristics, and interrupts. These requirements are documented as an F-22/LH specification, which is identical to the current JIAWG document. The IEEE has begun balloting on standard 1149.5, which is similar but not identical to, the logical layer portion of the JIAWG specification. The SAE has begun a physical layer specification which will capture requirements for the electrical characteristics.

The VHSIC TM bus is a four signal line (five for IEEE) serial data bus clocked at 6.25 MHz. Systems normally contain a dual redundant pair of these buses in the backplane. In addition, many module architectures and multi chip modules use this bus as a diagnostic port.

The IEEE and JIAWG versions of TM bus are relatively recent developments which are not of the same maturity as Pi bus. However, the VHSIC 3.0 TM bus was also modeled in the JIAWG gate level efforts, and is used in the platforms listed for Pi bus (plus F/A-18). Therefore, inter operability of TM bus systems from different vendors will continue to be a problem for some time, but this bus is supported by numerous avionics vendors.

### 6-2.3 IEEE-896 Futurebus+

The IEEE 896 Futurebus+ is designed to be used in single bus and multiple bus systems and to provide performance and cost scalability over time. Although the specification is principally intended for 64 bit address and data operation, a fully compatible 32 bit subset is provided. While IEEE 896.5 *Futurebus+ Military Profile Specifications* allows only the 32 and 64 bit versions, commercial Futurebus+ will also scale to 128 and 256 bits wide. In addition to being scalable in terms of bus width, Futurebus+ is also designed to be scalable in speed as new technology is developed. With its current Backplane Transceiver Logic (BTL) the top end performance is about 100 million transfers per second, or on a 64 bit bus, 6.4 GBPS. However, with differential ECL technology, such as is used in the German Autobahn bus, this could scale to 1.8 giga transfers per second, or 115.2 GBPS. While the protocol might be somewhat inefficient at this rate (the connect and disconnect phases would become relatively long compared to the data phase), it would still perform correctly. Futurebus+ takes its name from its goal of being capable of the highest possible transfer rate consistent with the technology available at the time the modules are designed.

Futurebus+ supports both a fast central arbiter and a slower fully distributed arbiter. Eight bits of priority are provided with both arbiters, but use of priority with the centralized arbiter slows it considerably. The priority feature is directed specifically at real time military usage, including the use of Rate Monotonic Scheduling.

Transmission of data over the multiplexed address/data highway is governed by one of two inter compatible transmission methods: (1) a technology-independent, compelled protocol, supporting broadcast, broadcast, and transfer intervention, and (2) a configurable transfer-rate, source-synchronized protocol supporting only block transfers and source-synchronized broadcast for systems requiring the highest possible performance.

Futurebus+ was designed primarily as a cache coherent shared memory bus, but it also supports large block transfers and message passing. Spice simulations of up to 26 boards on a single backplane have been run. These have shown that Futurebus+ will perform satisfactorily with at least 26 boards if some restrictions are placed on which slots are used in partially populated backplanes.

Five commodity integrated circuit vendors are currently selling or planning to sell Futurebus+ chips: National Semiconductor, Texas Instruments, Phillips/Signetics, Motorola Semiconductor Products, and LSI. LSI will sell a single chip solution, except for transceivers. They will also market a macro for custom chip developers.

The disadvantage of Futurebus+ is that it requires a large number of pins on the connector and that it does not have an error corrections mode. To achieve error correction requires dual buses which doubles the already large pin count.

#### **6-2.4 IEEE-1394 High Speed Serial Bus**

At the time of this writing the High Speed Serial Bus is in its final stages of development. It is a two conductor bus which has two variants: (1) A cable variant up to 10 meters long designed to be a peripheral bus for personal computers (replacing SCSI and other interconnects) and running at up to 200 MBPS; and (2) A backplane version using BTL logic and running at 50 MBPS with potential to 100 MBPS. Because the cable version will be used in large numbers of personal computers, it is expected to be very low cost.

Serial Bus is specified as an auxiliary bus to Futurebus+ in IEEE 896.5. It is used for test and maintenance, for software debug, as an RF module control bus, and as a low bandwidth general purpose interconnect for modules which do not require the performance of Futurebus+. For example, Serial Bus might be used as a communication path for load balancing power supplies. Serial Bus has also been selected by the military VME community as their auxiliary bus to be used for similar purposes.

Serial Bus supports real time deterministic scheduling with four bits of priority and the capability to limit the length of transfers. Serial Bus supports fault tolerance through use of dual buses. Serial Bus and Futurebus+ both follow the IEEE 1212 Control Status Register standard so that the two buses are very compatible. Information is passed using the IEEE 1212 memory mapped addressing scheme.

#### **6-2.5 VME Bus, extension to VME64 (Commercial VME Products, IEEE)**

VME, or Versa-Module Eurocard, is based on Versabus, which was developed by Motorola in the 1970s. The resulting bus standard is documented as ANSI/IEEE-1014-1987. This bus provides a memory interface, block transfer of small messages, and real-time interrupts. The VME specification defines four sub-buses within the architecture: data transfer bus, arbitration bus, priority interrupt bus, and utility bus.

The data transfer bus contains a 32 bit data path, 32 bit address (non-multiplexed), and control signals used to select various addressing modes. The arbitration bus is used to select the bus master via either priority based, round robin (rotating), or user defined protocols. The priority interrupt bus is used to provide real-time capability, such as might be needed for an error handler, a minor frame time out, timer expiration, or similar events. The utility bus provides power monitor and other health discretes.

The data transfers are asynchronous and can be performed as a block transfer (up to 256 bytes) or single word (four bytes). Up to 21 slots per backplane are permitted. All four sub-buses require 94 signal pins. The interrupt bus and asynchronous protocol can be used to perform memory transactions, as would be found in a tightly coupled computer architecture. The block transfer bus operation provides conventional message passing capability.

As mentioned in the introduction, VME is used extensively in the workstation market. Although there were significant inter operability problems in the early days of VME, market forces and continued standardization work have contributed to assure multiple vendor inter operability today. Such systems are also found in military aircraft, although applicability has been somewhat limited to laboratory environments found in aircraft operator mission stations. For tactical applications, subsetting is often used due to the harsher environmental and packaging constraints.

A 64 bit data path using the VME protocol is now in development by IEEE. This bus will be interoperable with existing 32 bit modules, but represents a significant performance gain for full 64 bit systems. Although 64 bit backplane buses for tactical aircraft present a difficult design challenge because of packaging issues, operator stations in patrol and anti-submarine aircraft will be able exploit this technology.

The VMEbus standard accommodates a wide variety of applications. It is capable of dynamic sizing to allow 8-bit, 16-bit and 32-bit devices to operate on a 32-bit address bus and 32-bit data bus. Data transfer rates up to 40 megabytes per second can be handled.

The VMEbus backplane may have up to twenty-one plug-in slot connectors in a standard 19-inch subrack. If fewer plug-in slots are used on any particular product, then the subrack may be smaller than 19 inches. In addition to the subrack, some vendors offer card cages that can be physically integrated into other products. In addition to the standard subrack, some vendors offer card cages that can be fit into other products, but which meet the VMEbus standard for plug-in cards.

Two different sized Eurocards can be plugged into the VMEbus backplane. The size 3U single height cards are 6.3 X 3.94 inches, and have a single 96-pin connector on one end of the card. The 3U card can accommodate a 24-bit address bus and a 16-bit data bus. The size 6U double height Eurocard is 6.3 X 9.19 inches, and uses two 96-pin connectors (designated P1 and P2) for a total of 192 pins along one edge. The 6U cards can accommodate 32-bit address and 32-bit data buses.

#### **6-2.6 IEEE-1596 Scalable Coherent Interface (SCI)**

The IEEE-1596 SCI is a system of rings and switches. It is intended for very high performance parallel processing - both small scale parallel and massively parallel. Rings and switches were selected as the basic communication medium, because they require only point to point links rather than multi-drop T-tapped bus lines. Point to point links

provide inherently cleaner signals and hence can run at higher speeds and lower voltages. In addition, switches provide for multiple simultaneous conversations among boards - a necessity for highly parallel systems. SCI rings, since they are insertion rings with bypass buffers, also allow a limited number of simultaneous conversations depending on the configuration of senders and receivers within the ring data flow. Because two party rings degrade into simple full duplex data links (one input and one output to / from each node), SCI has been able to define interface protocols which are applicable to both rings and switches.

With its support for both rings and switches, SCI is applicable for use in both switch based massively parallel systems such as the Butterfly machine, as well as mesh based architectures such as Touchstone. It can also support hybrids of the two architectures.

SCI has a 64 bit address space of which 16 bits are reserved for node addresses allowing up to 65,000 nodes. As its name implies, it is intended for use in cache coherent shared memory systems. It uses a directory based cache coherency protocol, because of the inherent scalability of that scheme. It conforms to the IEEE 1212 Control Status Register standard and supports the shared memory locks specified therein. Two variants of SCI are now defined: (1) A 16 bit wide (plus a clock line and a flag line) parallel version running at 8 GBPS on each point to point link; and (2) A serial version, which may be either electrical or fiber optic, running at 1 GBPS. Both the serial and parallel electrical versions use differential pair links to improve the signaling characteristics. Because the SCI protocol requires no handshakes between boards, links can extend for long distances limited only by the communications medium. Electrical links can extend to about 30 meters and fiber optic links to several kilometers.

While the baseline SCI is an accepted IEEE standard, further development is still taking place on variants. A low power version, featuring 0.25 volt signal swings, is under development. Other link widths are also under consideration, including an eight bit wide version and a four bit wide version. Higher speed serial versions are being investigated up to 4 GBPS. A proposal has been made by the Canadian Navy for a Real Time (RT) version of SCI dubbed SCI/RT. SCI/RT proposes several changes for deterministic scheduling, for improved fault tolerance, and for higher performance.

Currently three vendors are developing protocol chips for SCI. These are Dolphin Technology (Norway) doing a GaAs chip, and both LSI and National Semiconductor doing silicon implementations. Dolphin was scheduled to deliver chips in first quarter of CY-93. Convex has announced that they will use SCI in a new super computer. Hewlett Packard has announced that they will use SCI to link work stations to the Convex super computer using SCI.

## 7-0 KEY TECHNOLOGIES.

While technological capability is obviously not the only issue in system design, it is nonetheless very important. If the military avionics market is to leverage commercial technology to lower development cost and risk, then it is essential that a clear picture of the direction of such markets be established. This picture is possible on the basis of the industry briefs received by the AART, recent analysis by programs such as PAVE PACE, and periodic market previews by organizations such as IEEE.

Market sizes which are quoted in this section and in various literature should be used with some care. Even though the military avionics computer quantities are much smaller than the number of workstations purchased, the military still provides a significant percentage of research dollars which ultimately results in commercial products. As two examples, MIPS Inc. and Silicon Graphics are both now thriving commercial ventures (recently merged) which were spun off directly from DoD research. It will be a significant challenge to develop a policy which provides the market incentives for the commercial world to carry more of the burden of such research, especially when military requirements are often so different from commercial requirements. The first step in understanding this transition is to project trends in the commercial market, which is the purpose of this section. The trends are summarized in the Table 7-1 below.

**Table 7-1. 1998 Avionics Technology Status**

<b>ITEM</b>	<b>STATUS</b>
<b>Processors</b>	75 MHz., 64 bit RISC data processors; similar signal and graphics processors.
<b>Packaging</b>	(1) SEM-E dimensions, liquid flow through (2) Multiple chips hybrids with two or more processors and memory systems.
<b>Memory</b>	8192x8 SRAM
<b>Backplane Buses</b>	VME64 will be big in the commercial world and also with some military users.
<b>Interconnects</b>	Futurebus+ will emerge in the military world for some mission equipment, although many Pi bus users will remain.
<b>Software Languages</b>	Ada will still be used, C++ will begin to emerge as commercial software becomes reusable; CASE will mature
<b>Emerging Technologies</b>	(1) AD/DA converters will revolutionize mission avionics by replacing many analog devices; (2) Digital solderless microcircuit interconnects; and (3) Mission equipment (especially RF) will become modularized and standardized

## 7-1 Generic Technologies

Another technology which may have major impact is the digital solderless interconnects. This technology is used to mechanically attach multi-chip hybrids to interconnect boards in place of solder. Solderless interconnects may allow faster, and more automated, assembly with fewer mechanical defects, as well as more easily repairable systems.

### 7-1.1 Computerized Engineering Tools

Computerized systems engineering and detail design tools are evolving for hardware, software and overall systems design. These tools have the potential for significantly increasing engineer productivity, while at the same time increasing overall quality of the designs. In the hardware realm, the quality issue is addressed by checking myriad design rules and performing some of the reliability engineering tasks. Software tools, commonly called Computer Aided Software Engineering (CASE) tools, can check logical design and a whole host of other factors. It is important for systems engineers and managers to understand the tools that are available, and be familiar with both their strengths and their weaknesses.

A caution is in order regarding computerized tools. They do not replace good systems analysis, although they can enhance the effectiveness of systems analysis. Good tools will not help a mediocre design organization perform at high levels. While they may help a great deal in such organizations if properly used, they may also either wind up not being used at all (becoming "shelfware") or being misused by unskilled practitioners of the design arts. Poorly used tools can hide serious problems, not only in the design itself but also in the design organization and its management.

Yourdon<sup>1</sup> offers some observations about software computerized tools that are equally applicable to all forms of computerized systems engineering tools. He divides software tools into "lower-CASE" and "upper-CASE" categories, depending on their complexity, functionality and capability. He asserts that upper-CASE tools should be reserved for organizations that are mature, i.e., process oriented, at the Software Engineering Institute (SEI) maturity level-2 or above. Lower-CASE tools have a place in, and can be successfully used by, less mature design organizations.

One application of tools is in keeping order as a design emerges and evolves to the final system. Humans can only focus on a limited number of items at a time, so designers of complex systems must go through long check lists every time a change is proposed. As the system grows more complex the ability of human designers to complete the check list decreases, until eventually some important detail is missed. The computer-based tools can complete checklists of design rules, and do dynamic simulations that help the designer to understand and control complex interrelationships between system elements. Lack of such insight often causes seemingly good designs to perform poorly or incorrectly

Designs become much more robust when good tools are properly used. Designers now only have the time to evaluate a very limited set of tradeoffs, and they often have to

---

<sup>1</sup> Yourdon, Edward, Decline and Fall of the American Programmer, Prentice-Hall, Englewood Cliffs, NJ, 1992

accept the first design that meets the minimum design goals. Even then, it is sometimes true that educated guesses are passed off as analysis and the true situation is largely unknown. With the improved tools that are becoming available, designers will be able to examine hundreds of designs and pick much more optimum solutions. It will be possible to debug the software and the hardware concurrently before the system is ever built. This capability is projected to be fully available by the year 2000 at the latest, while interim capabilities are being constantly improved in the meantime.

It must be recognized that computerized engineering tools do not represent a "silver bullet," despite the progress that has been made, their most passionate advocates notwithstanding. Nor will they become silver bullets as they mature...unless design organizations that use them mature also. However, they are powerful tools that, in the hands of a skilled practitioner, can greatly improve the systems design process.

### **7-1.2 Emerging Technologies**

A significant technological revolution based on breakthrough discoveries occurred over the last few decades. These discoveries include the bipolar and field effect transistor, the integrated circuit, solid state, gas, and semiconductor lasers, fiber optic communications and sensors, focal plane arrays, gallium arsenide microwave devices, flat panel displays and solid state power control devices, etc.

Digital integrated circuits, semiconductor and optical memory, advanced networking (both electronic and optical), advanced packaging technology, flat panel displays, and the related processing technologies are being widely supported through both commercial and government supported research and development. These are key technologies which have dual use in the commercial and military information revolution which is taking place.

The Department of Defense, and especially the Defense Advanced Research Projects Agency (DARPA), have begun major initiatives in Microwave And Millimeter Wave Integrated Circuits (MIMIC) and Infrared Focal Plane Arrays (IRFPA). These technologies offer the possibility of extending the modular integrated avionics concept to the sensor area. Similar initiatives in optical control sensors and high bandwidth networks for sensor signal transfer are also under consideration. There is however no major commercial emphasis at this time by the industry in the sensor arena.

## **7-2 Microelectronics Technologies**

Microelectronics provides the critical difference between modern avionics systems and earlier designs. It is anticipated that microelectronic development will continue to pace the avionics industry, and will become even more important in the future. In the sections below are discussed some trends in microelectronics, along with specific microelectronic technologies.

### **7-2.1 Trends in High Density Microelectronics**

Microelectronics technology has been cited year after year by the DoD as one of the critical technologies which serves as the foundation for the electronic systems which are the "force multiplier" responsible for the US superiority in weapons systems. The US has always been a leader in this technology and, more importantly, in the design of sophisticated systems and software which leverage its speed and packaging density.

Microelectronics technology started with discrete transistors and passive components on a circuit board and quickly progressed to integrated circuits (IC's) consisting of several transistors and other components on a single slab of silicon. These IC's, usually called chips, have continued to grow in both density and speed while power per individual component goes down. Today it is possible to design and fabricate chips with  $\geq 100,000$  digital gates each and clock speeds of 100 MHz, as well as dynamic random access memory (DRAM) chips with 4 Mbits per chip. In fact, the integrated circuit technology has been doubling in density and performance at an astonishing rate of two years or less for decades. The Complimentary Metal Oxide Semiconductor (CMOS) transistors used in most modern digital circuits obey a scaling law which in effect says that the smaller you can make the components the faster the components will work and at less power. Also, the smaller you make the components the more components that can be fit on a chip.

The photolithographic technology that is used to define the ultra-small patterns on a chip is one of the keys to achieving these high performance chips. Today's state-of-the-art chips have minimum design rules of about 0.7 micrometers (millionths of a meter,  $\mu\text{m}$ ). These chips are manufactured using photolithography tools. In the future chips will be patterned using light of shorter wavelengths (i.e., deep ultraviolet) to achieve even smaller dimensions of 0.35  $\mu\text{m}$  to 0.5  $\mu\text{m}$ . When photolithography reaches its limits in terms of wavelength, process latitude, or other barriers, new technologies such as x-ray lithography, electron beam lithography and ion beam lithography, which are currently in development, will be ready to take over and push minimum design rules down to 0.25  $\mu\text{m}$ , 0.18  $\mu\text{m}$ , and 0.1  $\mu\text{m}$  or even smaller. DARPA and the Navy are currently pursuing these advanced lithographic technologies both in-house at the Naval research Laboratory and with a broad cross section of industry under the Defense Advanced Lithography Program.

As the lithographic technology permits building ever smaller chip patterns, we can expect to see memory chips increase to 16 Mbit DRAM's, 64 Mbit DRAM's, 256 Mbit DRAM's, then 1Gbit DRAM's (that is one billion bits of memory on a single chip) by the year 2000. While the computer memory market has been the key driver for increasing density, logic and analog chips will also benefit dramatically both from the smaller dimensions made possible by these new lithographic technologies and the new materials (e.g., thin film silicon-on-silicon, gallium arsenide, indium phosphide, etc.), devices and algorithms emerging to support the technology. Logic circuits will grow from 100,000 gate equivalents per chip to hundreds of thousands of gates per chip or even millions of gates per super chip (a large slab of single crystal silicon with multiple chip sites and interconnect all included in a monolithic circuit equivalent to an entire subsystem on a single large chip). These logic circuits will be capable of operating at clock speeds ranging from 100 MHz to tens of giga hertz. Such speed will enable whole new systems architectural concepts that provide more blending of analog and digital circuitry, and eliminate certain functions such as RF down converters and up converters.

Likewise, analog circuits will also benefit from the ability to shrink circuit patterns. Benefits will include higher efficiency, broader bandwidths, lower noise figures, and higher frequency operation. In both the analog and digital case, the cost of a given element (transistor) or function (gate, memory bit, etc.) can be expected to decrease; however, the cost of the large scale function (32-bit computer, memory subsystem, transmit/receive chip, etc.) will be much higher than what one normally considers a component should cost. The reason is that the value added at the historical component vendor will become higher as will the risk and cost of designing and producing these large scale functions. In fact, many of these component houses are going

into the multi chip module vending business which will permit them to sell much larger scale functions than they have made in the past.

Multi chip modules are large area metal, ceramic, composite, or metal matrix housings in which are placed substrates made of ceramics, composites, glass, silicon, polyimide, etc. with multiple levels of metal interconnect and an array of chips and other passive components. In some cases, the chips are stacked into three dimensional blocks to form extremely dense blocks of memory or logic. This type of packaging is just beginning to emerge and can be expected to incorporate digital and analog circuits as well as power supplies, photonic circuits, micro electromechanical functions, electro-optical components and electromagnetic functions in the late 1990s.

#### 7-2.1.1 What Does the Microelectronics Trend Mean?

The trend in ever higher density microelectronics is expected to continue throughout the 1990s and is expected to affect not only chip densities but also packaging densities. The costs to design and produce these super large scale functions are also expected to grow (e.g., \$1B capital investment to get into the gigabit dynamic random access memory business) which will most likely lead to fewer vendors and more interest in functional standards and open architecture systems, at least initially. In the long run, efforts in much more sophisticated simulation tools, computer-aided-design tools, computer-aided-manufacturing tools and computer-aided-test tools, coupled to the front-end of highly automated chip and module factories will lead to more application specific circuits and modules. The growth in the technology is expected to continue doubling about every two years throughout the 1990s and into the next century. The share of the market devoted to military products will continue to shrink on a relative scale to commercial applications, even for products such as microwave and millimeter wave integrated circuits where the military has been the largest user.

For the military program manager and the prime contractor, these trends in the microelectronics industry will necessitate a new paradigm for conducting their programs and business. For major programs such as a new aircraft development, it is no longer economical to develop a totally custom avionics suite because it will be considerably out of date by the time of initial introduction in the fleet. Instead, it may be preferable to develop an open architecture approach to an integrated avionics system. As stated elsewhere in this report, there is probably going to be a need for more than one architecture to meet the diverse mission requirements of the services and all the platforms, but many of the components of these architecture's can be standardized for a given time frame. The Government and industry will have to adopt and manage these standards in order to survive in a world where the commercial marketplace, not the military, dominates. It is desirable that the military systems utilize or adapt commercial technology wherever feasible by cleverly choosing products which are foreseen to be winners, and are likely to be supported by the commercial markets for a reasonable period of time. It may also be necessary to revamp our logistics strategy to plan for making lifetime buys of spare components early in the production cycle of new platforms or major updates.

To implement the new paradigm, the services and industry must be prepared to invest (preferably off-line from any major programs) in the development of advanced simulation tools and in the development of avionics test beds for these advanced architectures.

While the objective is to have open architecture, it will also be necessary to have a level of security which prevents compromise of the system or its exploitation by an adversary.

Because the services and prime contractors will be taking on more liability for the overall avionics system design and architecture, it will be necessary to develop verifiable advanced fault tolerance technology to insure functionality of all systems.

The new paradigm also will likely necessitate certain organizational changes in recognition of the fact that the aircraft is merely a seamless extension of the pilot and crew which will operate as a single entity instead of as a conglomeration of individual subsystems. Organizations will need to breakdown the outmoded historical boundaries between subsystems such as CNI, EW, radar, computers, controls, displays, software, weapons, etc. Such systems will operate with the aid of sophisticated resource managers, sensor fusion algorithms and make optimum use of human intelligence and control. To facilitate this goal, it will be desirable to maintain a significant number of on-going technology developments which build upon the successes achieved with each generation of technology rather than starting from scratch with each new program. If implemented properly, such an approach could yield significant savings to each program manager by reducing risks associated with starting from scratch, permitting an optimum selection of systems on-board versus off-board a given platform, minimizing the need for derivative aircraft types, and lower maintenance costs (years between scheduled repairs). In addition, updates would be much simpler through the use of modular hardware and software tied together using large bandwidth signal and data flow networks.

While there are a lot of advantages to these breakthroughs in microelectronics and associated packaging and interconnect technology, there are also some potentially serious drawbacks. One of these is the old problem of cosmic radiation damage to logic and memory circuits which becomes more exacerbated as the dimensions of these device and circuit patterns becomes smaller and smaller. Such radiation can cause single event upset (SEU) or single event latch-up (SELU) in circuits with sub micron dimensions, and increases in probability of occurrence as a function of altitude. This problem can be overcome through proper design and engineering, but points to a potential issue in using off-the-shelf components not designed for aircraft application. Another problem which can be anticipated in future systems using high density electronics is one of thermal density and the need to remove heat. It is probable that future systems will require that some form of liquid cooling be available. Techniques such as liquid flow through modules and racks, liquid immersion, and higher conductivity materials (including light weight composites, diamond filled composites, and metal matrix materials) are among a number of approaches being pursued to alleviate thermal density problems.

Program managers need to recognize that the thermal management problem is a potentially significant weight and reliability issue and that integrated avionics architectures with distributed hardware may necessitate providing cooling to more remote locations in the aircraft. The trend of moving more avionics towards the skin, or embedded in the skin, of an aircraft (smart skins) will continue and will necessitate new environmental protection technology. Microelectronics materials other than silicon (e.g., gallium arsenide, indium phosphide, and silicon carbide) will enable more environmentally tolerant circuits which may eventually not even require cooling.

### 7-2.1.2 What About The Threats?

The continuing revolution in microelectronics is no longer unique to the United States and this picture is not expected to change much in the foreseeable future. While it appears to be true that the US is regaining a slight lead over our nearest competitor, Japan, we will not likely see that gap increase by any significant amount. In fact, to the contrary we may see a more determined effort on the part of other nations to gain more of a leadership role in designing logic circuitry, rather than memory chips, as they now surely recognize that there is far more profit in those chips than in memory chips.

Also, with the breakup of the Soviet bloc into independent nations there is more emphasis on the US releasing more and more products to foreign buyers. This trend is probably good for maintaining a strong US industrial base, but it also means that any potential adversary will have access to modern microelectronics technology that may be more current than what is in our present weapon systems. It can be expected that even Third World nations will have easy access to the components needed to build very formidable military systems EW, radar, CNI, and processors. While these nations may not be able to field technologically superior aircraft, they will be able to build-up large inventories of countermeasures, detection equipment, communications equipment and fire control systems making the projection of air power more difficult. In turn, this increased threat density and sophistication will demand even more capability from our aircraft avionics in the late 1990s and beyond. Also, the explosion in the application of microwave and millimeter wave microelectronics in more and more commercial applications like (e.g., cellular communications, smart highways, entertainment, security sensors, etc.) will make the problem of distinguishing friend from foe far more difficult.

### **7-2.2 Random Access Memory Chips**

Random Access Memories (RAMs) are the true "commodity" parts of the digital age. All computers need some place to store data during program execution, and that is what random access memory provides. Because RAM is so universal, it is produced in very large quantities, and the ability to get more working chips from a given wafer size translates directly into profit. Primarily for this reason, advanced lithography has usually appeared first in the memory parts, and later in the more complex and less regularly connected logic devices. A second reason is that memory devices contain such a high degree of image repeatability that they are much easier to test for lithographic mistakes.

Random access memories have for a long time been divided into two main types: *dynamic* and *static*. There are other types (e.g., nonvolatile), but these two categories predominate. In dynamic memories, charge is gated onto or off of a capacitor, and the resulting potential represents either a one or a zero. Since neither the gate transistor nor the capacitor is perfect, some charge leaks, and eventually it is impossible to determine what state was supposed to be represented. To overcome this problem, present dynamic RAMs (or DRAMS) have circuits built in which periodically read each memory location and then rewrite it in a process called refresh. While refresh is going on, the memory location being refreshed is unavailable for use. Thus, in a high performance system, DRAMS cause unplanned for wait states and are avoided. On the other hand, for non-time critical memory, DRAMS are used heavily because they are much cheaper and have more memory bits per single chip die.

For static RAMs (SRAMs), the second predominate memory type, the one or zero state is stored by creating a bistable multi vibrator. Since this circuit usually requires six transistors per memory bit, SRAMs take up more room on chip. On the other hand,

each state is stable, and no refresh is required, which means that the time to access any given bit is a constant number of clock cycles. SRAMs have an additional advantage in that they are usually more immune to system noise, a particularly important feature in time or result critical systems. They also tend to be more resistant to instantaneous radiation upset, since the charge produced by a passing particle is more likely to alter the amount of charge stored than to cause a multi vibrator to change states.

For various technology reasons, the SRAM is usually faster than the DRAM, while the DRAM has higher levels of integration. At present, 16 Meg DRAMS with 50 nsec access time are just becoming available, and there are claims that 4 Meg SRAMs with 10 nsec access time can be purchased. Actually, in large quantities one can get 4 Meg DRAMs and 1 Meg SRAMs. By the end of 1994, 64 Meg DRAMs at 40 nsec and 16 Meg SRAMs at perhaps 5 nsec will become available. Beyond that, there is some question as to whether it will be more cost effective to make large single die chips or go to multi-chip modules (MCMs) which contain several die each.

There may be problems with attempting to produce memory chips having lithographic features much below 0.35 microns ( $\mu\text{m}$ ). Present technology is in the 0.8  $\mu\text{m}$  to 0.5  $\mu\text{m}$  range for aggressive producers. Since the area of a die is approximately proportional to the square of the line widths used, cutting line width in half usually reduces the size by a factor of four, and eventually (as the density of defects on a wafer decreases) leads to four times as much logic or memory per chip. In addition, as defect densities drop, the manufacturers usually go to larger areas per chip.

For memories in particular, the repetitiveness of the design makes creating die with spare parts built into the die much easier. This allows the designer to use a larger number of viable transistors in a memory chip than in a logic part. And, since memory is so ubiquitous in its use, it will continue to have large markets and be favored by the chip manufacturers for their future investment. Even as diminishing returns set in for lithography, the size and usability will continue to improve, making older memory parts quickly obsolete and harder to locate for sparing. On the other hand, the simplicity of memory devices makes it a much easier part to retrofit in a system, so memories should not create an obsolete parts problem for future systems unless chip architectures changes provide for much less noise and power margin.

An important development is the move to put substantial memory caches on the same chip with the logic in a three level memory scheme, and putting SRAM caches directly on large DRAMs to make them look more like SRAMs. But these are just cosmetic in nature: the way in which information is retrieved from memory remains the same. A large number of lines (one line for every power-of-two number of memory cells) are required to access a memory by single location.

A more fundamental change in memory technology will be the move to "smart memories", where the processor asks for a piece of data by name rather than by location. Since many pieces of data take up more than one location, this change will allow a decrease in the number of address lines required to access a memory, as well as allowing for immediate off loading of many simpler direct data operations.

With all the changes coming in the future, many of them as yet unforeseen, and the increasing size and complexity of systems enabled by the memory market, the only way to maintain a system in the future will be to maintain an accurate simulation of the system wherein one can try out changes and determine the effects of changing parts. This will be a necessity because the total life cycle of a large system is many times greater than the lifetime availability of the parts it contains. Thus, as each part or group of parts

become either unavailable or uneconomic to purchase, adequate substitutes will have to be determined through modeling in order to take into account the entire range of variables in the manufactured parts. This variability is the key factor in determining if a part is adequate, but it is too difficult in practice to physically try out all the combinations which can (and will) occur when the parts are introduced into the actual system. The ability to do this simulation cost effectively probably mandates the use of IEEE-1076 as the simulation and description language for digital systems.

### 7-2.3 Gate Array Technology

For parts which are to be built in smaller quantities, designers presently use gate arrays. These are arrays of gates which are all made identically, but are not connected. A designer just does one layer of the design, the one that connects the gates together, and the manufacturer takes some arrays out of stock, does the last layer, tests, packages, and delivers the array. However, there is another technology which is rapidly catching up with gate arrays, the so called field programmable gate arrays. These are completely fabricated and packaged, and one electrically programs the necessary interconnects to produce the desired part. Of course they are not as tightly packed as a custom part, but they can be programmed and tested in a couple of hours instead of the six months it takes to produce a fully custom part.

It is with these parts that the computer tools will really make a difference. Soon, the designer will create a mathematical design, and the tools will partition the design onto FPGAs, program them, and program circuit boards to accept the programmed parts. It will be possible to build high level systems containing many millions of gates of logic in a few weeks. At the same time, the computers will be able to calculate all the "ilities" for each design. These calculations are not difficult conceptually -- they just require an enormous amount of labor. Using the tools, systems will no longer require replacement when parts become unavailable. Using the design from the system's documentation base, replacement parts will be designed whenever necessary, and will be readily available.

At the same time, designs will become much more robust. Designers now only have the time to go through a very limited set of tradeoffs, and they have to accept the first design which meets the design goals. With the improved tools which will soon be available, designers will be able to examine hundreds of designs and pick much more optimum solutions. This ability to optimize the design will go a long way to alleviate the inefficiencies caused by using non-custom parts. The software to control the system will be designed at the same time the parts are, and will be run on very accurate computer simulations while the system is being built. It will be possible to debug the software and the hardware concurrently before the system is ever built. And all this should be available by the year 2000 at the latest.

### 7-2.4 Analog-to-Digital Converters

An area of technology that is important to the future of avionics, especially tactical sensors, is analog-to-digital (A/D) and digital-to-analog (D/A) converters. The A/D converter is a device or circuit that examines an analog voltage or current and converts it to a proportional binary number that can be input to a computer. The D/A converter is the inverse of the A/D, i.e., it converts a binary number to a proportional analog voltage or current. As these converters become faster, there will be decrease in the dependence of avionics on analog components. The PAVE PACE final report states that Gallium Arsenide (GaAs) 20 bit, 10 Gbit/second A/D will begin to emerge in 1997 and

be mature beyond 2001. The D/A will probably lead A/D by approximately 18 months (many A/D designs incorporate an embedded D/A).

### 7-3 Microwave/Millimeter-wave Monolithic Integrated Circuit (MIMIC)

The MIMIC program is a seven year DARPA/Tri-Service defense technology initiative to develop and demonstrate affordable and available MIMIC chips for application to radar, electronic warfare (EW), communications and smart weapons systems. Many people confuse the intent of the MIMIC Program with that of the Very High Speed Integrated Circuit (VHSIC) Program, a similar defense technology initiative conducted during the first half of the 1980s to counter the erosion of the US lead in silicon based digital integrated circuits. Unlike VHSIC, which was designed to push digital integrated circuit technology, the MIMIC Program was initiated when it became clear to defense acquisition managers that systems based on an emerging Gallium Arsenide (GaAs) analog integrated circuit technology capable of operation at microwave and millimeter-wave frequencies, were generally too expensive to procure. While the Demonstration and Validation (DEM-VAL) models of these systems indicated that advanced levels of performance could be achieved, the risk and cost of putting them in production was considered unacceptable due to the inability of the limited number of relatively small GaAs Microwave/Millimeter-wave Integrated Circuit (MMIC) foundries to produce chips with a reasonable yield. This problem was further exacerbated by the inability of designers to design a working and yieldable MMIC without many design passes due to the inadequacy of good device and circuit models as well as computer aided design (CAD) tools.

Like the VHSIC Program which preceded it, the MIMIC Program is comprised of four major program phases. There were sixteen Phase 0 concept study contractor teams representing forty-eight companies. Phase 0 was designed to lead into two consecutive program phases, Phases 1 and 2. These are the program phases during which the bulk of MIMIC technology development and demonstration is occurring. Work in Phase 1 was conducted from May 1988 through May 1991 by four contractor teams led by the following: a.) Hughes/GE; b.) ITT/Martin Marietta Joint Venture; c.) the Raytheon/TI MIMIC Joint Venture; and, d.) TRW. Work in Phase 1 included materials, devices, modeling, design tools, circuits, process line validation, data base establishment, quality and reliability evaluation and brassboard demonstrations.

Three year Phase 2 contracts were awarded in August 1991 to the Hughes/GE team, the Raytheon/TI MIMIC Joint Venture team and the TRW team. Work in Phase 2 includes continuing the technology developments begun during Phase 1, developing and demonstrating advanced technologies (e.g., multi-function chips, more efficient devices, additional millimeter-wave chips, etc.) and more brassboard demonstrations. Finally, Phase 3 of the MIMIC Program runs concurrently with Phases 1 and 2 and is structured to support smaller development efforts which, if successful, could have a positive impact on the Phase 1 and 2 major initiatives. Work in Phase 3 includes high frequency microwave and millimeter-wave test probes, new materials growth and epitaxy technology, advanced packaging techniques, advanced device and circuit models, computer-aided-design tools, reliability and radiation tolerance evaluation, and others. All three services and the MIMIC contractors are working toward the eventual development of a MIMIC Hardware Description Language (MHDL) to simplify the process of specifying new chip designs to foundries, and to assure inter operability of components from different designers and foundries.

Although modeled after the VHSIC Program, the MIMIC Program has benefited from lessons learned during that program. VHSIC was successful in moving the digital integrated circuits technology forward at an accelerated pace; however, many of the VHSIC foundries did not succeed after the conclusion of the program due largely to a lack of orders for VHSIC parts. This situation resulted, in part, from the guidance behind the VHSIC contract efforts which emphasized generic chips which, as it turned out, had almost no market. Fortunately, the strong commercial interest in pursuing digital technology quickly provided a home for the flow of technology developed under VHSIC to other vendors mostly in the form of Application Specific Integrated Circuits (ASIC's) and memory chips. Unlike VHSIC technology, which has an almost infinite commercial market that dwarfs the military market, MIMIC technology does not presently enjoy a large commercial market and must be focused on as many real military applications as possible. This focus has been the guiding principle from the beginning of the MIMIC Program in order to assure a market for the resulting MIMIC chips. Many of the basic requirements associated with military applications of MIMIC's (e.g., wide bandwidth, high power, etc.) are counter to most commercial requirements governed by FCC regulations (e.g., narrow band, low power, etc.).

The MIMIC Program requires each contractor team to develop and update a business plan to help forecast market opportunities and to guide program developments. In addition, MIMIC contractors are encouraged to focus their chip designs on specific application opportunities to help assure a market for the resulting products.

The VHSIC management and execution model was carried forward to the MIMIC Program. The services take turns acting as the coordinating service for planning a tri-service MIMIC procurement, then all three services award contracts against these various solicitations once they are approved by DARPA. Each service is responsible for the award and execution of the contracts assigned to it (e.g., the Navy was assigned Raytheon/TI MIMIC JV for Phases 1 & 2, the Air Force was assigned Hughes-GE for Phases 1 & 2, and the Army was assigned the ITT/Martin Marietta JV for Phase 1 and TRW for Phases 1 & 2). All services and DARPA participate in monitoring all contract efforts. Each MIMIC Phase 1 & 2 contract includes chips and brassboard demonstrators for all three services even though each service has indicated different applications of preference (e.g., wide band & EW- Navy, active array radar- Air Force, and millimeter-wave sensors-Army).

### 7-3.1 MIMIC Phase 1 Accomplishments

The four contractor teams who performed under MIMIC Phase 1 contracts were highly successful in achieving validated process lines for the manufacture of MIMIC chips. Eighty-four different chip designs were processed and thousands of chips were delivered for further evaluation. Raytheon/TI MIMIC JV and Hughes/GE both demonstrated transfer of chip designs between foundries and that these designs can be used to process functionally equivalent MIMIC's. All Phase 1 contractors demonstrated several brassboard insertions of MIMIC chips and modules. For example Raytheon/TI MIMIC JV demonstrated MIMIC Phase 1 chips in their EW Active Array brassboard and the Generic Decoy (GEN-X). In some cases MIMIC served as an enabling technology such as in seekers for the Advanced Anti-Aircraft Missile, EW Active Array brassboard, and SADARM while, in other cases, MIMIC reduced costs such as the HARM missile, GEN-X, and ASPJ. Table 7-3.1 below describes examples of MIMIC Phase 1 chips and their applications.

Table 7-3.1 Examples of MIMIC Phase 1 Accomplishments

	CONTRACTOR	CHIP TYPES	APPLICATIONS
EW	Raytheon/TI JV	Wideband LNA's, Wideband Power Amplifiers, Phase Shifter, Switches	GEN-X Decoy, STRAP, Towed Decoys, ESM, EW Active Array Brassboard, Jammers
	ITT/Martin Marietta JV	Amplifiers, Phase Shifters, Convert- ers, Attenuators, Oscillators	ASPJ & other jammers, ALQ-136, & Decoys
	TRW	Tuners, Switches Mixers, Receiver and Amplifiers	Chanelized Receiver Brassboard to Improve AN/SLQ-32 and ESM
RADAR	Raytheon/TI JV	LNA's, Power Amplifiers, Digital Attenuator	C, X, & Ku-band Active Array Radars and Shared Aperture
	Hughes/GE	LNA's, Attenuators Amplifiers, Phase Shifters, Switches	C-band & X-band Active Array
CNI	Raytheon/TI JV	Ka-band LNA & Down Converter	MILSTAR
	ITT/Martin Marietta JV	C-band Multi- function T/R Chip, Amplifiers VCO's, Converters	Combat Engagement Capability (CEC), SHF SATCOM
	Hughes/GE	LNA's, Phase Shifter, Mixer, Oscillator	GPS Antenna Electronics
SMART WEAPON	Raytheon/TI JV	Ka-band LNA, Q-band VCO and Power Amplifier	Missile Seeker SADARM
	Hughe/GE	Ka-band LNA and Power Amplifiers Mixer, Limiter & VCO	Advanced Milimeter- wave Missile Seeker
	ITT/Martin Marietta JV	Milimeter-wave Receiver	SADARM, MLRS TGW, Longbow
	TRW	FM-CW Transceiver, Milimeter-wave IC's	Multioption Fuze for Artillery, SADARM

### 7-3.2 MIMIC Phase 2 Directions

The MIMIC Phase 2 Program is comprised of three contracts -- one each managed by the Army, Navy and Air Force-- to further the achievements of previous Phase 1 and 3 efforts. The broad objectives of Phase 2 include: a.) continued cost reduction for a given RF function; b.) continued reliability and quality improvement leading to eventual Qualified Manufacturers List (QML) approval; c.) validation of new processes; d.) advanced devices and technologies; and, e.) additional MIMIC insertions and demonstrations. The brassboard and module demonstrators will again have application to EW, radar, communications and smart weapons.

The Navy is directing the Raytheon/TI MIMIC Joint Venture MIMIC Phase 2 Program team which includes Lockheed Sanders, Teledyne, Hittite, General Dynamics (Ft. Worth)(now Lockheed), Aerojet, Airtron and Consilium. Work includes more complex and compacted chips, shrinking the dimensions of the Phase 1 Metal-Semiconductor Field Effect Transistor (MESFET) technology from 0.5  $\mu\text{m}$  to 0.25  $\mu\text{m}$ , validating several new MIMIC processes, new devices, better materials, advanced packaging technology, four brassboard demonstrations and eight chip/module demonstrators. The Raytheon/Texas Instruments team is developing 28 new chips which address the needs of at least forty identified applications. The performance of these chips are beyond the Phase 1 levels of performance. Yield and cost goals for these chips are also beyond those of Phase 1. Eleven of the 28 Phase 2 chips are power amplifier chips. The goals for these Phase 2 power amplifier chips exceed the results achieved in Phase 1. Phase 2 power amplifier chips require a different process than those used in Phase 1.

The Raytheon/Texas Instruments JV team is developing Heterojunction Field Effect Transistor (HFET) and Heterojunction Bipolar Transistor (HBT) processes in Phase 2. These processes were not required to meet the Phase 1 program requirements. The goal for HFET power added efficiency is 48% at 18 GHz. The goal for HBT power added efficiency is 55% at X-band. Both of these goals are far in excess of the 15% Phase 1 result. The noise figure goal for the team's 0.25  $\mu\text{m}$  ion implanted low noise process is 1.7 dB compared to 5.5 dB at the end of Phase 1. One aspect of the MIMIC program is typified by chip cost and yield improvement goals established prior to Phase 1 for MIMIC chips as a function of time. Prior to Phase 1, overall yield for standard processes was approximately 10% and the cost per square millimeter of chip area was approximately \$20. The Phase 1 goal was 25% yield and \$2 per square millimeter. The goal for Phase 2 was established at 40% yield and \$0.80 per square millimeter, while transitioning from 3" wafers to 4" wafers. Based on Phase 1 results, these earlier projections are still valid and in some cases yield numbers have already exceeded the Phase 2 projections. However, cost is still heavily driven by volume considerations and will continue to be dependent on the realization of business according to market projections.

At the same time, the functionality of the chips under development is doubling every couple of years. For example, prior to Phase 1 typical amplifiers had one or two stages of gain. During Phase 1, typical amplifiers had three stages of gain. During Phase 2, amplifiers will have four or in some cases five stages of gain, as well as having additional circuitry such as switches built into the same chip.

The performance is also continuously improving over time. During the Phase 1 time frame the power added efficiency goal for a wideband (C-X-Ku-Band) power amplifier was 15%. The Phase 2 goal for an amplifier covering the same frequency band is 25% and the power produced is 4 times that of the Phase 1 chip. These system driven

performance requirements are driving the team to develop and validate new processes in Phase 2 that can support the improved performance.

In addition to cost, yield, complexity and performance advances, the Phase 2 program will develop 28 chip designs to insert variously into F-22 EW Array, Advanced Multimode Missile (AMMM) seeker, Anti-Air Warfare (AAW) Subarray for ships, Airborne Shared Aperture Program (ASAP), High Performance Armament System (HIPAS)/M734 Fuze, MDOT decoy, F-22 Avionics, Man Portable Radar (P-STAR), Microwave Landing system (MLS), F-22 T/R Module, Advanced Airborne Electronic Decoy/ Integrated Solid State Module (AAED/ISSM), Sparrow Missile, and Advanced AAW T/R Module. Phase 2 also includes several enhancements to applications initiated during the Phase 1 program such as GEN-X, AMRAAM, Seek and Destroy Armor (SADARM), Micro-TWT Driver Module, and STRAP. By comparison, during Phase 1 the team developed fourteen chips designs to insert into six hardware demonstrations.

The TRW team, which includes Westinghouse, General Dynamics, Alliant Techsystems, and Hercules, under direction of the Army, is developing and validating several new technologies including: a.) a 3  $\mu\text{m}$  HBT process for VCO and VCO/mixer applications to achieve low phase noise; b.) a 0.2  $\mu\text{m}$  HEMT process for a wide range of mixer, converter and low noise amplifier brassboards to achieve high gain at low noise over wide bandwidth; c.) a 0.1  $\mu\text{m}$  HEMT process for W-band transceiver and receiver brassboards to achieve a three-stage amplifier, image reject mixer, monolithic detector and voltage controlled oscillator; d.) a 0.2  $\mu\text{m}$  Power HEMT process for amplifiers; and, e.) a 0.5  $\mu\text{m}$  PFET process for use as RF down converters, mixers, phase shifters, and power amplifiers.

TRW is developing an integrated electronic warfare (EW) receiver brassboard. The primary insertion candidate for that product is the Aircraft Survivability Equipment (ASE) Radar Warning Receiver aboard the RAH-66 Comanche helicopter. The secondary insertion candidate is the Special Threat Analysis and Recognition (STAR) receiver system, which has multiple airborne applications including potential for integration with any radar warning receiver. The brassboard consists of three Integrated Microwave Assemblies (IMA's), a block frequency converter, a single conversion tuner, and a dual conversion tuner. A total of five chips will be developed. TRW is also developing millimeter wave low noise amplifier and millimeter wave oscillator chips using PHEMT technology that can be inserted into the ALR-67 ASR Millimeter Wave Receiver (MMWR). Other applications include wideband T/R module for Airborne Shared Aperture Program, X-band transmit modules for radar, W-band receiver for Multiple Launch Rocket System, W-band Transceiver for X-Rod, broad band receiver for missile seekers, EHF data link transceiver for EHF SATCOM, and Q-band transceiver for SADARM.

The Air Force is directing the Hughes/GE (now Martin Marietta) MIMIC Phase 2 Program team (which includes AT&T, Litton, Rockwell International, M/A-COM, Alliant Techsystems, EESof and Cascade Microtech). The Hughes/GE team is developing and validating several processes including: a.) continuation of the signal and power MESFET process work begun during Phase 1; b.) a 0.25  $\mu\text{m}$  Pseudomorphic High Electron Mobility Transistor (PHEMT) processes for several frequency ranges to improve performance of amplifiers, VCOs and mixers; and, c.) HBT process for high power amplifiers at X-band.

The brassboard demonstrations of the Hughes/GE team include: a.) X-band T/R modules for an active array radar brassboard; b.) receiver and transmitter modules for a Smart Target Activated Fire and Forget (STAFF) Brassboard; c.) low noise and power

amplifier modules for Single Channel Advanced Man Portable Terminal (SCAMP); and d.) low noise and power amplifiers for the Ground Based Interceptor (GBI) Communications System.

### 7-3.3 Tri-Service MIMIC Reliability And Radiation Effects Program.

Begun during MIMIC Phase 1, the objective of this program is to target specific reliability areas of critical importance to the services. The goal is to ensure with a high degree of confidence that the MIMIC chips meet, or can be made to meet, system reliability specifications. An important aspect of this objective is to promote insertion of MIMIC technology, leading to enhanced military capabilities. Through a closely integrated reliability program, the services can determine the resultant effects of material, design, fabrication processes, assembly, and packaging on the chips and modules developed under the MIMIC Program. The service laboratories working on this effort include primarily the U.S. Navy Naval Research Laboratory (NRL), US Army Laboratory Command (LABCOM), and the U.S. Air Force Rome Laboratory (RL) and Wright Laboratory (WL).

The tri-service MIMIC Reliability and Radiation Effects Program closely coordinates the reliability activities of the three Services on the MIMIC Program and assigns responsibilities based on expertise, work load, unique Service need and Service unique capabilities. Under the MIMIC Phase 2 program the 3 contractor teams will have a total of 10 foundries covering 21 GaAs processes including Field-Effect Transistors (FETs), High Electron Mobility Transistors (HEMTs) and Heterojunction Bipolar Transistors (HBTs). From these foundries four types of Standard Evaluation Circuits (SECs) from each team will be delivered. The four basic types of SECs will be : microwave Power Amplifier (PA), microwave HBT PA, microwave Low Noise Amplifier (LNA), and a millimeter-wave LNA. A common SEC carrier will be specified by DARPA, one type for microwave and one type for millimeter wave.

The approach consists of a coordinated Tri-Service effort organized to efficiently accomplish the program objectives. Each Service will assume a number of "prime" responsibilities that are centered around eight key thrusts. This division of responsibilities provides each service with specific areas to focus on, and in the aggregate provides the government with a comprehensive coverage of the MIMIC technology. The Navy will focus on RF life testing of 1 to 26 GHz MIMICs. It will test both low noise and power SECs and will be able to effectively standardize on reliability test fixtures and carriers. In a similar manner, the Army will focus on the millimeter wave MIMIC SECs and be able to standardize fixtures and carriers in the 27 to 100 GHz frequency range. The Army will also evaluate 1 to 18 GHz devices applying a life test facility developed in Phase 1.

The Air Force will focus its reliability assessment work at the fabrication process level and module level. Evaluation of MIMIC test structures will be conducted. The RL work will be closely coupled with the WL materials/device correlation work in which contractor High Density Test Reticle (HDTR) data are evaluated and analyzed. In addition, reliability life test of module level MIMICs will be accomplished through the Air Force. The tri-service reliability assessment work will be supplemented by special area evaluations of radiation effects, electrostatic discharge (ESD), high power microwave (HPM) hardening, thermal analysis, and electrical bias and RF overstress testing. Data analysis and physics of failure tasks will also be conducted.

To date the Navy, Air Force, and Army have conducted in-house activities in support of the MIMIC Reliability and Radiation Effects Program. The Navy program, conducted by the NRL, centered on the inspection of Phase 1 deliverables, revising the Tri-Service Reliability and Radiation Effects Integration Plan, demonstration of the EW brassboard, and RF life testing. Phase 1 deliverables consisting of chips on carriers, test fixtures, reports, modules and brassboards required by the contract have been received and visual inspections completed. A demonstration of the EW brassboard was held at the NRL anechoic chamber in the Tactical Electronic Warfare division. An RF life test of the MIMIC designated element circuit function (ECF) was completed at NRL.

The Air Force RL program concentrated on finite element thermal modeling, reliability life testing of C-band MIMICs, test structures thermal stressing, QML, product evaluation, and failure analysis. Thermal modeling of the ECF MIMIC and IR thermal measurements of this device were coordinated by RL and NRL. Ten C-band power amplifiers were received and life testing was initiated. Thermal stress testing of test structures was carried out to 800 hours.

The Army LABCOR Reliability/Radiation Effects program concentrated on building test fixtures for RF life testing. A new 64 position C-band and Ku-band RF life test system is now operational.

#### 7-3.4 MIMIC Section Summary

Although we are just past half way through the MIMIC Program, the technology is already finding a home in production systems (e.g., HARM, GEN-X, etc.), and MIMIC chips were found in some of the systems used in Desert Storm. During Phase 1 there were tens of thousands of chips processed and hundreds of each type were delivered to the Government for inspection and further evaluation. MIMIC has become an enabling technology for systems such as the F-22 radar, EW active array jammers, advanced missile seekers, more effective armament and others. MIMIC has also demonstrated it can reduce the cost of systems such as GEN-X decoy, ASPJ, HARM, etc.

MIMIC Phase 2 efforts are concentrating on further increases in output power, increased efficiency, wider bandwidth, lower noise figure, increased functionality, on-wafer test of power circuits, and increased yields as well as qualifying new processes. Statistical process control of the manufacturing process is expected to continue to increase the yield and lower the cost of a given function. Special attention is being given to improving visual yield of the finished chips. A significant amount of progress is expected towards achieving QML for the MIMIC foundries, with approval anticipated within a short time after the completion of Phase 2. QML approval should further reduce the costs associated with applying MIMIC technology. The number of brassboard demonstrations is increasing as are the number of defense systems planning to use MIMIC technology.

Many of the MIMIC foundries are experiencing an increased interest in applying the technology to commercial applications such as direct satellite broadcast, cellular communications, smart highways, automotive sensors, instrumentation, wireless local area networks, and others. Several of these foundries are already under contract for components for both domestic and foreign markets. These markets are projected to grow substantially over the next ten years which will hopefully help counter the anticipated decrease in the total number of defense applications for MIMIC's in the same time frame. The best evidence of the success of the MIMIC Program is that these foundries report that they are viewed as highly competitive on the world market. This situation can only

improve further as MIMIC Phase 2 progresses and should serve to continue the reduction in cost of parts for both commercial and defense systems.

### 7-3.5 Beyond MMIC

The MIMIC Program has been a major success story in demonstrating more available and affordable microwave and millimeter wave integrated circuits and it is planned to use MIMIC as the model for several other defense technology thrusts. Two areas where this result will be seen in the near future is microwave/millimeter-wave packaging and in active array demonstrations. MIMIC concentrated mostly on the chip technology and design tools whereas these new initiatives will concentrate on simplifying the design, fabrication and testing of module containing these chips into functional circuits (e.g., transmit, receive, etc.) for both military and commercial markets. Work will include new housing materials and concepts, new interconnect concepts, new module level computer-aided-design tools, new thermal management techniques, new power regulation and distribution concepts, and others. These efforts will lead to using concepts similar to the multichip module techniques used in digital packaging for microwave modules. It is foreseen that by the mid 1990s transmit/receive modules will be fabricated as arrays of monolithic modules, using novel packaging and cooling techniques, with integral radiators and high density power supplies (e.g.,  $\geq 100$  watts/cu. in. and  $\geq 90\%$  efficient). These modules will be batch fabricated and tested using highly automated equipment yielding more affordable modules. In turn, such modules will permit building active array radar, receivers, transmit arrays, wideband shared aperture arrays, etc. which are not only more affordable but also simpler to install due to their low profile.

## 7-4 Advanced Processor Technology

### 7-4.1 Microprocessor Technology

Microprocessor technology is presently moving along two courses that are represented by two generic types of device. The first generic type is the Reduced Instruction Set Computer (RISC) architecture, which is the driver behind the rash of very high performance workstations. They are optimized to support a multi-task management system like UNIX, and depend on the compilers to support the less used but more complex tasks while being optimized for the simple register-to-register operations. Normally they devote as much chip area to registers as possible. They also depend heavily on caching (moving entire blocks of memory from a global slow memory to a fast local on-chip memory) and on uniform program flow. The uniform flow is necessary in order to get a large cache hit ratio (i.e., ratio of the number of words found locally vs. the number called for by the processor). This type of processor is represented by the Sun SPARC, the Intel i960, the HP PA, and the DEC Alpha. The Alpha is probably the current leader, but new chips appear often and capability is growing rapidly. Also, as lithography (chip making technology) improves and more capability is available on chip, the different types of resources are beginning to increase. Even the definition of RISC has been changed to mean "single cycle instructions" rather than the original "few simple instructions".

The second generic type of microprocessor is the Complex Instruction Set Computer (CISC). This type is characterized by on-chip hardware to provide more complicated hardware and software operations. CISC processors do the same operations as the RISC chips, but the most frequently used operations (such as load, store, AND, OR, NOR, etc.) are not performed as fast because of complicated checking and status

register setting which must take place. As such, they place more dependence on the hardware than on the software compilers to support operations. These chips include the Motorola 680XX, the Intel 80X86 series used in personal computers, and the engines for most previous generation computers.

A third class of processing chip which is not usually accorded the title of microprocessor, is the Digital Signal Processing (DSP) chip. This chip is finding a niche as the embedded processor in controllers, and as the arithmetic processor in signal processors and workstations. It is characterized by two things: an on-board single-cycle Multiplier/Accumulator (MAC), and extremely efficient I/O capability. These chips include the TI TMS320C40, the AT&T DSP32C, the Motorola DSP96002, the Intel i860XP, etc. to name a few. There are frequently other resources available on the DSP chip such as DMA controllers, A/Ds, and D/As. For computationally intensive tasks on large data blocks, the DSP is usually orders of magnitude faster than the RISC or CISC approach. On the other hand, software of a level higher than assembly or C does not normally map well onto DSP architectures.

#### 7-4.2 Examples of Commercial RISC, CISC and DSP Chips

In this section there is information on representative examples of the RISC, CISC and DSP chips.

##### 7-4.2.1 RISC

An example of the RISC type chip is the Intel i960 family. This family of 32-bit processors was, according to Intel, designed especially for embedded applications. All members of the series share a common core architecture (RISC technology), and the core functions are object code compatible. Each processor in the series has a different set of special functions. There are both commercial and military versions available -- for instance, the 80960KB (commercial, floating point) and the 80960MC (Military, floating point) have identical architectures, and nearly identical pin-outs in the same type of pin grid package. The main difference is that the 80960MC has a military temperature range, and reduced specification. For instance, the MC is only rated to 20MHz clock and 7.5 VAX MIPS, while the "KB" is rated at 25MHz and 7.5 VAX MIPS. The core 80960 appears to be sixteen 32-bit global registers, sixteen 32-bit local registers, and at least four 32-bit control/pointer registers. It also includes a built in instruction cache (512-byte), and a built in interrupt controller and bus logic control. The "KB" and "MC" contain also four 80-bit floating point registers and a built in floating point unit capable of 5.2M (4.0M for "MC") Whetstones at full clock, doing IEEE 754 floating point operations. Floating point add/subtract takes 0.5/0.7  $\mu$ sec for 32/64 bit words; multiply 1.0/1.8  $\mu$ sec, 32/64 bit words; divide 1.8/3.8  $\mu$ sec; square root 5.0/5.2  $\mu$ sec; sine 20.3/22.1  $\mu$ sec, 32/64 bit words; etc.. The 80960 series uses a high bandwidth local bus. It has a 32 bit multiplexed address/data path, 15 lines for control of address, data, and signal lines, and two bus arbitration lines. It has a four word burst capability, which allows 1 to 16 byte transfer operations, and 43 Mbytes of sustained read/write operations.

##### 7-4.2.2 CISC

For an example of the CISC architecture, one can examine the Intel 80XX6 series (the famous "PC" chip) or the Motorola 68XXX series. These chips contain such items as address generators, as well as a large microcoded instruction set. The Motorola series

in particular contains a large number of symmetric addressing schemes which allowed high level software to easily address memory. Later chips in the series have on-board cache, cache controllers, floating point coprocessors, etc.

#### 7-4.2.3 DSP

An example of the DSP is represented by the Texas Instruments (TI) TMS320XXX series. In particular is TI's fourth generation DSP, the TMS320C40. It features high speed multiple I/O ports, and like all DSPs is optimized for the multiply accumulate (MAC) signal processing function. It has six byte wide DMA controlled concurrent ports, each capable of 20MBytes/sec bi-directional flow, which allows a large variety of multiprocessor architectures to be efficiently implemented. It also has two identical 32-bit buses for memory and or data access.

Internally, the 320C40 contains a full 32x32 MAC, a single cycle FPU multiplier, and support for single cycle external memory accesses. It will also perform an integer to floating point conversion in a single cycle, and even has a built-in barrel shifter. It has twenty-two 32 bit registers, and twelve 40 bit registers. Instruction cycle is 40 nsec at 50 MHz clock. It supports automatic module and bit reverse addressing. It supports both IEEE-754 32 bit as well as a 40 bit floating point format. Another feature of high value to signal processing and to real time processing is its ability to respond to interrupt instructions.

The 320C40 does a floating point MAC in 40 nsec at 50 MHz, a floating point divide in 360nsec. It can do eleven operations in parallel, and has an instruction set of 135 instructions. It does not have an auxiliary memory management unit, and does not run an operating system such as UNIX. It does have C compilers available, and should have an Ada compiler soon, because there is one for its predecessor chip (the TMS320C30).

Industry respondents addressed the issue of standard instruction sets as the focus of computer policy. Industry opinion is that evolution toward the Application Program Interface (API) to be the most effective approach to standardization instead of the assembly language Instruction Set Architecture (ISA). In this methodology, standard operating system interfaces and software linkage procedures for input/output, timers, diagnostics, and interrupts would be defined instead of the assembly language interface. This would free the application programmer from the necessity of understanding low level hardware features, but each processor type would require a unique operating system implementation, although the interfaces would be standard. The API approach, if fully implemented, decouples software from the hardware technology obsolescence issue, so this approach offers much promise.

#### **7-4.3 Technology Trends and Time Lines.**

The life of commercial processors can be gauged by an *Electrical Engineering Times* article (29 June 92) which states that Intel 80386 usage will decline from 76.9% (\$620M) in 1991 to 45.1% in 1996 (\$403M). The 80386 market loss will be replaced mostly by the Intel 80486, with a next generation product predicted to acquire 7.5% of the market in 1996.

Graphics processing is another area of opportunity. Reuse of commercially developed graphics software libraries can be simplified by using the graphics processor

for which such software was written. Although there is some movement in this direction, additional work is needed to prove the approach and minimize risk of its use.

Finally, although not stated explicitly by the industry presenters, it is clear that the DoD must find some way to leverage the trend for groups of companies to band together for product development. One example of this is the MIPS R-4000 processor, licensed by MIPS Inc., but manufactured and developed by Integrated Device Technologies, LSI Logic, and Performance Semiconductor. Another example is the SPARC processor developed by Sun Microsystems in conjunction with Bipolar Integrated Technology, Cypress Semiconductors, Fujitsu Microelectronics, LSI Logic, and Texas Instruments Semiconductors. Just as airframe manufacturers are pooling efforts on new aircraft designs, commercial processor designs will become joint ventures to reduce risk and share development cost.

#### 7-4.4 Key Developments For Advanced Processing

The industry briefs contained discussions of two important advanced computer efforts: *Aladdin* and *Touchstone*. The Aladdin processor is an advanced signal processing computer made by Texas Instruments for target recognition applications. This computer uses the R-4000 processor, multiple TMS320C30 signal processors, the VME System Bus (VSB), and is programmed in the Ada language. Touchstone is a DARPA initiative for supercomputer applications. Intel is building a commercial computer based on the Touchstone architecture using Intel i860 processors. Honeywell is currently prototyping a militarized version of this system.

##### 7-4.4.1 Hewlett-Packard Processor

The Hewlett-Packard was one of the early RISC architectures, originally introduced by HP in 1986 as the *Spectrum* architecture. It has a register based architecture, 32 bit instruction word, simple addressing modes, load/store operations, simple hard-wired instructions, and no full integer multiply or divide instructions. It uses the Harvard architecture: separate data and instruction paths. The latest version uses large external caches (up to 1Mbyte Data and 2 Mbyte Instruction) connected by a 64 bit non-multiplexed bus.

The latest version, the PA-7100, has a FPU integrated on the CPU, along with 32 32-bit general registers and 28 64-bit floating-point registers. The CPU can issue a floating point instruction on the same clock cycle as a CPU instruction. External cache controllers are on chip, along with a separate System-interface 32-bit multiplexed bus interface. It also has a six stage pipeline for a faster memory interface. The PA-7100 uses 850,000 total transistors.

On Alpha there are no special registers that would prevent pipelining multiple instances of the same operations (no MQ register and no condition codes). The instructions interact with each other only by one instruction writing a register or memory, and another one reading from the same place. This feature makes it particularly easy to build implementations that issue multiple instructions every CPU cycle. There are no implementation-specific pipeline timing hazards, no load-delay slots, and no branch-delay slots, and no binary compatibility issues across multiple implementations.

Alpha is unconventional in the approach to byte manipulation. Single-byte stores found in conventional RISC architectures force cache and memory implementations to include byte shift-and-mask logic, and sequence logic to perform read-modify-write on

memory words. Instead, Alpha does byte shifting and masking with normal 64-bit register-to-register instructions, crafted to keep the sequences short.

Alpha is also unconventional in the approach to multiprocessor shared memory. As viewed from a second processor (including an I/O device), a sequence of reads and writes issued by one processor may be arbitrarily reordered by an implementation. This allows implementations to use multi-bank caches, bypassed write buffers, write merging, pipelined writes with retry on error, etc. If strict ordering between two accesses must be maintained, memory barrier instructions can be explicitly inserted in the program.

Finally, Alpha includes a privileged microcode library (PALcall) which allows it to run full VMS using one version, OSF/1 using a second version (mirrors many MIPS operating-system features), and NT with a third version. Other versions can be tailored for other operating systems. This feature makes Alpha an especially attractive architecture for multiple operating systems.

#### 7-4.4.2 Aladdin

The Aladdin computer program is sponsored by ARPA and the U.S. Army Night Vision and Electronic Sensors Directorate (NVESD). The objective of the Aladdin program is to develop a very high performance miniature processor. Texas Instruments and Alliant Techsystems are developing processors for this program. Additionally, Texas Instruments is adapting its Aladdin processor for the Air Force Ultra Reliable Digital Avionics (URDA) program. The objective of this program is to develop a prototype processor that combines a data processor, signal processor, memory, and system interface on a single SEM-E card AT&T is also developing an URDA processor.

TI's Aladdin processor is an Ada programmable, modular, scalable 32-bit parallel processor. The basic building block of TI's Aladdin processor is the Basic Processor Module (BPM). The BPM consists of a 100 MIPS R4000 scalar processor, a 400 MFLOPS Vector Coprocessor (VCP), local memory (SRAM), a 128 Kbyte secondary cache, and SPROM containing the configuration parameters for the R4000 power-up sequence. Access to BPM local memory is via a crossbar switch and is controlled by 2 Processor Interface Control (PIC) devices. When multiple BPMs are stacked together each BPM interfaces to a 25 Mword/second Aladdin System Bus (ASB) for inter processor communication, external system communication, and data I/O. Each BPM also interfaces to a 25 Mword/second Input-Output Bus (IOBUS) for additional I/O bandwidth capability.

The BPM's VCP is a micro-programmable, custom designed pipelined processor based on TI's 8846,47 family of signal processors. The VCP contains two 32-bit floating point/integer multipliers and two 32-bit floating point/integer ALUs. This provides a peak processing throughput of 400 MFLOPS at a clock frequency of 100 MHz. The VCP far exceeds the capability of existing processors in executing typical signal processing algorithms. For example, a vector multiply and add takes the Intel i860 0.12 ms, the TMS320C40 0.05 ms, and the VCP 0.01 nos.

TI's Aladdin configuration for the ARPA/NVESD program is a stack of 5 BPMs with a peak throughput of 500 MIPS scalar concurrent with 2000 MFLOPS vector processing. This is accomplished in a "soup can" configuration in a volume of 4.5 inches diameter by 2.6 inches long.

TI is also adapting its Aladdin processor for the Wright Laboratories/NVESD URDA program. TI's Aladdin configuration for the URDA program consists of 2 BPMs repackaged in MCM packages to fit on one side of a SEM-E card. This provides a capability of 200 MIPS/800 MFLOPS on the single card. TI has included PiBus and TMBus interfaces on the other side of the SEM-E card. Liquid-Flow-Through (LFT) cooling is used.

TI's Aladdin processor is in the assembly stage and its URDA processor is entering preliminary design review. While TI is not currently working on adapting its Aladdin processor for commercial use, it has plans to submit an unsolicited Dual Use proposal for such an effort.

Alliant Techsystems and AT&T are also involved in developing Aladdin/URDA processors. Alliant's Aladdin computer consists of a distributed memory C30<sup>2</sup> based MIMD architecture combined with an array coprocessor with a SIMD architecture communicating over a data flow network. Alliant's Aladdin processor is designed for use in image processing and target recognition applications and is currently in prototype for use in an acoustic sensor application. AT&T's URDA processor uses a C-40<sup>3</sup> based approach.

#### 7-4.4.3 Intel's Evolutionary Processor Growth

Intel Supercomputer Systems Division (SSD), with support from DARPA, has pursued a significant research effort in the area of massively parallel processor (MPP) technology and system products over the last several years. The Intel/DARPA R&D initiatives has have evolved from early MPP developments, the Touchstone program, to the current TeraFLOPS initiative. Intel SSD products, such as iPSC/1, iPSC/2, iPSC/860 and the Paragon<sup>TM</sup> XP/s, have evolved from these Intel/DARPA initiatives.

All of the iPSC machines have been multiple instruction multiple data (MIMD) designs. The iPSC/2 and iPSC/860 have been marketed as relatively inexpensive supercomputers and attracted wide interest. It was reported by the *IEEE Spectrum* in September 1992 that over 325 machines have been sold by Intel worldwide. The iPSC/860 has been difficult to program for peak performance both at the node and parallel levels, therefore there are very few third party applications available as yet. It should be noted that programming Intel's machines is no more difficult than programming any other parallel time-to-solution machines. Intel continues its efforts to improve the iPSC/860 user's environment. It actively works on advancing the quality of its software development tools to facilitate the application. It can be expected that future developments along these lines will greatly improve the ease with which such machines can be programmed.

The Paragon<sup>TM</sup> is a MIMD machine that supports both the message passing and data-parallel programming models, and is considered one of the world's most powerful computers. Two machines have been produced in 1992, one for Oak Ridge National Laboratory and one for the Boeing Company. The Oak Ridge machine is a 512 node Paragon<sup>TM</sup>, projected to deliver over 100 gigaflops peak performance. The Boeing

---

<sup>2</sup> C30 refers to the Texas Instruments TMS320C30 signal processing chip

<sup>3</sup> C40 refers to the Texas Instruments TMS320C40 signal processing chip

Paragon™ machine will replace another earlier version MPP machine. The current Paragon™ design supports 66 to 1,000 nodes. Each node will have either two or five Intel i860 XP microprocessors and 16 to 64 megabytes of memory. In every node, one of the microprocessors controls the message passing communications to other nodes, and the others are the computational elements. Each node also has a mesh router chip to enable communications to any of the four nearest neighbors. Intel has received more than thirty orders for Paragon™ machines, including six for Europe and Japan. It is Intel's opinion that teraflop performance Paragon™ XP/S systems can be built today, but it may not be affordable. Plans are to wait for a new generation of microprocessors, memories, and other VLSI components to make it more affordable. Intel's DARPA-sponsored TeraFLOPS initiative will evolve this R&D architecture over the next five years and provide the technology for an affordable teraflop 30 to 40 million dollar supercomputer. Also, the major high performance computer issues such as languages, programming tools, processor communications and operating systems are being addressed by Intel to facilitate the application.

The Touchstone program has provided the research and development to advance the Scalable Parallel Computing Technology base for the 1990s. It has served as the catalyst for computing systems scalable over a wide performance range. The recently announced TeraFLOPS initiative between Intel and DARPA will be the next catalyst to advance the state-of-the-art of future massively parallel high performance computers. DARPA has agreed to provide \$21M over the next five years in a jointly funded research agreement with Intel to accelerate the development of TeraFLOPS supercomputers. TeraFLOPS systems are going to be developed to meet what are termed the "Grand Challenge Problems" of science, as well as important defense and national security applications. Three critical technologies will be addressed by the TeraFLOPS initiative:

1. Advanced parallel software architectural design, developed by the Touchstone program, will be the foundation for future generations of scalable, parallel supercomputers.
2. Advanced inter processor communication and computing concepts proven in the Intel iWARP real-time Supercomputer will be enhanced and implemented in new submicron VLSI components.
3. Development of a new generation of floating point RISC microprocessors derived from the Intel i860 to increase computational power and maintain software compatibility with the Paragon.

This Intel and DARPA coordinated effort is expected strengthen competitive position for the U.S. computer technology.

#### 7-4.4.4 Alpha

The Alpha processor is a true 64-bit RISC architecture (it has no 32/64 mode bit), with a minimal number of 32-bit instructions. It is not a 32-bit architecture that was later expanded to 64 bits, but rather it was designed for 64-bit service in the first place. It was designed with particular emphasis on multiple instruction issue, multiple processors, and software migration from VAX VMS and MIPS ULTRIX. Alpha is a load/store RISC architecture with all operations done between registers, based on 32-bit instruction words. It has 32 integer 64 bits registers and 32 floating 64-bit registers. Longword (32-bit) and

quadword (64-bit) integers are supported. Four floating data types are supported: VAX F-float, VAX G-float, IEEE single(32-bit), and IEEE double (64-bit).

The integer operate instructions manipulate full 64-bit values, and include the usual assortment of arithmetic, compare, logical, and shift instructions. There are just three 32-bit integer operations: add, subtract, and multiply. These differ from their 64-bit counterparts only in overflow detection and in producing 32-bit canonical results. There is no integer divide instruction.

The floating point operate instructions include four complete sets of VAX and IEEE arithmetic, plus conversions between float and integer, but there is no floating point square root instruction.

The Privileged Architecture Library call (PALcall) instructions deal with interrupts and exceptions, task switching, virtual memory, and other complex operations that must be done automatically. PALcall instructions vector to a privileged library of software subroutines (using the same Alpha instruction set) that implement an operating-system-specific set of these complex operations.

The first chip implementation runs at up to 200 MHz. At 3.3 volts and a 150MHz clock speed, the chip dissipates 23 watts of power, so it will use 32 watts of power at full speed. DEC claims that the speed of Alpha implementations will scale up from this level by at least a factor of 1000 over the next 25 years

#### 7-4.4.5 Touchstone (DARPA)

Over the last three or four years there has been a technology explosion in all areas of processor technology, emanating primarily from the commercial sector world-wide. Intel's DARPA-sponsored Touchstone program is a high performance computing (HPC) technology demonstration initiative that takes advantage of such technologies. This R&D cooperative initiative was intended to demonstrate the massively parallel processor (MPP) technology as the high performance computer (HPC) architecture, promising a large price and performance advantage over traditional supercomputers. The Touchstone is a distributed-memory multiple computer, multiple instruction, multiple data (MIMD) model using a 2-D mesh interconnection architecture which is very representative of future parallel processing trends. Each processing node is based on the use of multiple i860™ XP RISC microprocessors for high throughput computation and inter processor communication. The Touchstone program includes several technology milestones, such as DELTA and SIGMA, that resulted in the completion of the experimental Touchstone Delta machine and Intel's Paragon™ XP/S product line.

The Touchstone Delta machine was installed at California Institute of Technology in Pasadena in May 1991, as one of the world's most powerful computers consisting of 528 processors based on the i860 RISC microprocessor. The Touchstone design is a scalable distributed memory MIMD architecture arranged in a 2-D mesh using message passing for its communications. The Delta machine has achieved record 13.9 gigaflops on a highly parallel Linpack benchmark for solution of large system linear equations. The Paragon™ XP/S is Intel's current state-of-the-art supercomputer system. The system delivers scalable performance from 5 to 300 gigaflops and is marketed as a commercial product.

The Touchstone/Paragon scalable design is ideal for embedded parallel applications offering considerable throughput and growth path with continuing

Intel/DARPA support for future evolutions under the TeraFLOPS program. The Paragon™ family will result in a considerable software investment that will be supported both by Intel and the commercial users.

Use of the Intel COTS software where applicable (e.g., compilers, linkers, etc.) and the Verdex distributed ADA is a strategy for embedded Touchstone. Development of domain specific application shells and utilities for the Touchstone are also being addressed in order to lower end user costs. These include parallel expert systems, real time object oriented data bases, and data fusion and tracking shells which can significantly benefit advanced Navy tactical aircraft.

Other DARPA/DoD software research initiatives in academia, industry and government laboratories can be leveraged to support Touchstone technology applications. The Paragon™ model represents the industry leading microprocessor and interconnect technology.

There are some key developments in high performance computing, advanced packaging and other related processor technologies that may enable rapid prototyping of high performance computing for military systems in a short time at relatively low costs. One of the key developments is the militarization of the Touchstone system and related technologies. Militarization of the Intel/DARPA Touchstone for high performance military applications is already in place with the new Honeywell/DARPA co-funded R&D initiative. The program is executed with a cooperative effort between Intel and Honeywell to repackage the Touchstone design in an advanced military package, using the SEM-E card format. The new processor cards will be developed by Honeywell using the GE high density interconnect (HDI) MCM technology, which arose from the DARPA HDI initiative. The SEM-E high-density cooling design is being provided by Lockheed, which developed the SEM-E cooling system for the F-22 Program. It is Honeywell's responsibility to produce and integrate this technology in a military chassis as a scalable system and test it with the exact same software as the Intel commercial system. This strategy will enable a military user to select COTS HPC technology without repackaging if its functionality can support the requirements. It also allows use of the commercial software available from the Paragon™ system. The DARPA strategic requirements for this effort is that:

1. No new chip design for core system.
2. The system must be evolved around Intel's commercial upgrade cycle of two years.
3. Execute the exact commercial system (Paragon™) software.
4. Follow military form factors.
5. Consider unique systems requirements for future systems such as fault tolerance, security, and parallel utilities for real-time applications. As part of this initiative Honeywell is also emerging as a leader in the embedded software domain by coordinating the use of the software products being developed by the DARPA.

Under a DARPA contract GE developed High Density Interconnect (HDI) multichip module technology, which is being installed in a new TI automated manufacturing facility. The TI and GE team can provide the industry's most advanced

multichip module (MCM) foundry service supported by an integrated CAD/CAM system. HDI is a thin film multichip interconnect technology that connects complex bare chips with copper/polyimide multilayer overlay structure with extremely close chip-to-chip distances to achieve gigahertz range speeds. HDI offers high performance in today's interconnect and packaging problems due to its design using short interconnect distances, high-speed, controlled impedance, and optimum chip connections. It can be fabricated in days for rapid prototyping since the technology is available today. This is very significant since it offers state-of-the-art packaging technology that can be used as NDL therefore saving the user the non recurring costs.

## 7-5 Commercial Trends in Processor Architecture

Commercial architectural trends and efforts have evolved around open architecture and massively parallel processors (MPP) for high throughput applications. The trends to new models featuring open systems hardware and software, addressing the needs of hundreds of users, using commercial off-the-shelf technology (for both hardware and software), and enabling the application software design to lead the system design is the wave of the future. This section will address only the massively parallel efforts. MPP systems represent a small but rapidly growing segment of the overall computer market. Acceptance has been difficult in the past because of the immaturity of system software, but the situation is rapidly changing. MPP systems acceptance is being affected by the lower order-of-magnitude price/performance advantage over traditional computers and the recent accelerated software initiatives addressing parallel processors. As an example, Intel and Digital Equipment Corporation have launched a joint software initiative for High Performance FORTRAN (HPF) and other application-development and system software for massively parallel computers. HPF is an architecture-independent, high-level programming language that provides standards-based application portability and high performance. The compiler to accept HPF is being developed by the High Performance FORTRAN Forum, made up of experts from industry (Intel, Digital, Cray Research, IBM, and Thinking Machines), academia and government.

Commercial MPP computers are divided into two market segments identified as *throughput systems* and *time-to-solution systems*. Throughput systems run a single job on each processor, allowing many jobs to run simultaneously so that each job runs as a uniprocessor, but many jobs complete at the same time, Time-to-Solution systems work by dividing a single job across multiple processors, to reduce the run time of that particular job.

MPP computers are parallel computers with normally over a hundred processors capable of processing simultaneously a single problem or working in parallel on separate problems. Difficulties with the implementation of MPP systems is the synchronization of the work of the multitude of processors. This synchronization is achieved with complex hardware and software design. The software design is the most difficult to complete.

Another major consideration in the past has been the debates over the merits of single-instruction-multiple-data (SIMD) and multiple-instruction-multiple-data (MIMD) architectures. With SIMD, all the processors execute the same instructions in lockstep, but on different data. In a MIMD computer, however, each processor can be executing a different program because it has its own memory.

The market for throughput systems is more mature and widespread at large corporations throughout the world in production environments. The key to their success

is the use of commercial microprocessors and support chip technology to achieve a lower price and performance advantage.

Time-to-Solution systems are perhaps the most likely MPP computers to satisfy the high-performance architectures of future supercomputers and embedded military applications. The Connection Machine (CM-5) and Intel Paragon<sup>TM</sup> XP/S have emerged as the leaders in the industry because of their earlier availability and performance. The CM-5, which was announced 1991 and delivered in 1992, is the first machine that is capable of delivering speeds in excess of one Teraflop. Such configurations would sell for more than \$200 million. Because of this high cost, only 100 Gigaflops machines are being produced presently.

The CM-5 machine is also very flexible since it lets the users program in message-passing or data-parallel modes, or combination of the two, therefore allowing SIMD operation. The Intel iPCS/860 and PARAGON<sup>TM</sup> XP/S have been discussed in greater detail under the DARPA Touchstone paragraphs. Most of the other time-to-solution systems address the lower-end of the MPP applications. As an example, the MASPAP MP-1100/1200 machines normally compete with the low-end of the connection machine applications. Also the n-Cube/2 machine competes directly with the Intel systems. It is anticipated the MPP high-performance computer market continues to grow over the next five, ten, and fifteen years, and actually double each year over the next 5 years.

#### 7-5.1 Microprocessor Chip Progress

Microprocessor chip advancements are perhaps the most explosive technology evolving world-wide in the commercial sector that can significantly increase performance and functionality of military systems. The IEEE International Solid State Circuits Conference held in February 1992, reported on a 1000 MIPS BiCMOS microprocessor with superscalar architecture. An Intel Corporation study, titled "Micro 2000 Prediction" (1992), envisions a 1 inch x 1 inch chip with 50-100 million transistors with four 250 MHz processors offering 750 MIPS/1000 MFLOPS (peak) performance. It is anticipated that Intel will develop two new generations of processor components improving the i860 family by a factor of ten. It is also anticipated that Intel will develop three new generations of interconnect components under the TeraFLOPS program by merging the message passing models of Touchstone<sup>TM</sup> and iWARP<sup>TM</sup>. These predictions suggest that the TeraFLOPS program goals of achieving scalable multicomputer systems in excess of one TeraFLOPS performance by late 1995 is a relatively low risk. It can be expected that the TeraFLOPS program will result in three new generations of HPC multicomputers at the 300 GFLOPS, 600 GFLOPS, and 1.8 TFLOPS performance levels.

## **8-0 ADVANCED PACKAGING TECHNOLOGIES AND ADVANCED ELECTRONIC PACKAGING**

Although specialized module sizes will continue to exist, it is evident from the industry briefs and the PAVE PACE reports that the Standard Electronic Module, Format E (SEM-E) will be very much in evidence in future systems. However, higher functional densities will cause designs to exceed the 40 watts which air cooled modules cannot dissipate. Therefore, some type of liquid cooling will be needed on future modules. Module and system designers will be challenged to prove that new technologies (such as dripless connectors) can actually be supported in the fleet.

Perhaps no trend will have greater impact on module designs than the multichip hybrid. These devices utilize Wafer Scale Integration (WSI) to combine into a single package the microcircuits that previously existed in separate packages. WSI also allows three dimensional arrangement of such silicon wafers. The result is much greater functional density and simpler designs for each SEM-E module. In the McDonnell-Douglas PAVE PACE report, 12 different devices were identified with four different package sizes.

The high functional densities will allow the entire core processing architecture to be housed in two Integrated Avionics Racks (IAR), versus the dozens of subsystems in current generation aircraft. Based on the PAVE PACE reports, these IARs will hold approximately twenty modules each with five to ten growth slots. These two units will perform all the data and signal processing needed for communications, navigation, displays, and weapons delivery. Next generation systems will also modularize the radio frequency (RF) elements (frequency converters, receivers, preprocessors) into similar integrated racks.

The approach to module packaging for military systems is quite different from that of commercial systems. For the military computer module, high functional density is needed to control weight, reduce computer size, and improve reliability by decreasing parts counts and number of interconnects. In addition, module lengths, widths, and thicknesses are determined by vibration modes and mechanical rigidity requirements. These goals are achieved by using multiple layer modules (i.e., printed wiring boards), multi-chip hybrid circuits, and module dimensions such as SEM-E. This approach is probably impractical and not cost effective for commercial boards which need not meet the harsh environmental and systems limitations of military computer modules. Commercial modules tend to be much larger in size, single layer (although sometimes double sided), and multiple chip hybrids are shunned.

Due to the significant difference in requirements for electronic packaging for military systems versus commercial, it is quite likely that unique military packaging designs will be needed for the Airborne Uninhabited Fighter (AUF) applications. However, commercial modules are certainly viable for shipboard, ground based, and rack mounted airborne equipment which does not have such severe requirements.

### **8-1 Avionics Environments and Packaging**

Packaging consist of three primary areas of concern during design: connectors, thermal management, and structural. A fourth major area of packaging which may be necessary on some projects is component mounting (mainly when circuit density forces surface mounting or other advanced component mounting technology).

This section is arranged to discuss connectors, thermal management, and structural aspects. Current issues will be discussed in each area as well as technology trends and cost considerations. Repackaging of commercial technology will be discussed under this general section as well as standardization issues related to packaging.

It must be recognized that there are three distinctive environments in aircraft applications including transport, helicopter, and attack/fighter. Application of electronics in these environments affects packaging in the structural and thermal management areas. Transport aircraft are not very stringent in these respects. However, helicopters and attack/fighter aircraft, due to high vibration and drastic temperatures, require special design emphasis on the structural and thermal management techniques for the electronic equipment. With these environments in mind, application of the typical commercial equipment on transport aircraft would generally be acceptable, with regards to packaging, but would require special or added hardware to survive in the helicopter or attack/fighter environments. These extra structural and thermal management techniques are typical in military equipment designed for these environments. When considering repackaging commercial equipment for helicopters or fighter/attack aircraft, one must have cost incentives other than hardware development (such as software reuse or tool availability) because repackaging the commercial equipment can cost from 75% to 90% of the cost of newly developed hardware; making repackaging not necessarily desirable, especially at the circuit card level. Of course, repackaging at the semiconductor die level is very inexpensive compared to the cost of special die development.

## 8-2 Standardization and Packaging

Standardization with regards to packaging is effective and has proven such in the standard electronic module (SEM) program, and is documented to provide an eight-to-one development cost savings. To standardize a package, one must insure all interfaces to the package are defined and held constant (including the connectors, thermal interface, and structural interface). The most effective point to standardize a package is at the lowest repairable or replaceable unit. This fact has significant ramifications to the logistics and supply systems. The SEM program was successful with the philosophies of standardization at the circuit card level and with throw away cards.

The avionics industry is embracing standardization at the circuit card level with the SEM-E module. However, standardization of the SEM-E interfaces has not occurred: there is no standardization of connectors, thermal interface, or structural interface. The best that specifications of the word "SEM-E" buys is a card that is approximately 6"x6". Standards such as MIL-STD-1389 for SEM-E do exist, but seem to be ignored in many areas. Also, most SEM-E cards are expensive and are not designed as throw away units.

## 8-3 Commercial Avionics Package Standards

The most recent developments in the commercial avionics field is being pushed by Aeronautical Radio, Inc. (ARINC) for the Airborne Electronic Engineering Committee. This effort includes Line Replaceable Modules (LRM's) operating together to form an Integrated Modular Avionics package. The size of the LRM is determined from the Avionics Modular Unit (AMU). These units vary in width from 1 AMU=1/8 ATR to 10 AMU=1/2 Air Transport Rack (ATR); the height is fixed at 7.22 inches, and the depth at 15.00 inch. As an example, a LRM 10 AMU wide would have the dimension of 4.70 inches wide by 7.22 inches tall and 15.00 inches long. The LRM's are to be cooled by convection to ambient air. The connector shell is capable of providing 204 contacts for a

LAMU divided into three sections. The contacts are of the pin and socket type. Radial has been awarded the initial contract for these connectors. A four-bar linkage mechanism at the front of each LRM is intended as an insertion/extraction device, and is intended to prevent relative movement greater than 0.005 inch between connector halves. The substance of the thinking for LRM's is that they should be a self-contained unit with low density components and a low density connector which plugs into an Integrated Modular Avionics package to provide some environmental isolation and is ambient air cooled.

Contrast this ARINC 650 module with the SEM format-E which has the dimensions of 0.580 inch wide by 5.88 inches deep by 6.41 inch tall with a possible 396 signal contacts. The spine of the module is aluminum instead of fiberglass and conducts the heat from the module to the surrounding structure. The structural enclosure provides the environmental protection for many modules mounted together for integration, but still capable of individual replacement. This very dense electronic package accommodates the time proven blade and tuning fork connector. The tuning fork contact mounted to a printed circuit backplane through a compliant section eliminates many troublesome solder joints, and can interface up to twenty-three layers, including power and ground planes, and provide matching impedance.

The technological difference between these two methods of packaging avionics may be evident from the volume applied by the basic unit; 509 cubic inches for the ARINC 10 AMU versus 21.86 cubic inches for the SEM format-E (A difference of 20 times). The maximum weight for the ARINC 10 AMU is 16 pounds which is about ten times heavier than the SEM format-E module. What is proposed as the Integrated Modular Avionics package for the next generation transport could be adequate for a Military Air Transport Service (MATS) aircraft, but it would lose the advantage of using the avionics developed for other air services. However, the discussion about weight must also include the necessary ducting from the Environmental Control System (ECS) or liquid cooling system required by the SEM format-E due to its increased packaging density. Providing the cooling to the Integrated Modular Avionics package allows the package to be located in many areas of the aircraft where an ambient cooled package could not be located. In a combat vehicle, being able to install the electronics in a desirable location has a great advantage because exterior profile must be restricted and a large amount of equipment carried internally.

#### **8-4 Standard Hardware Acquisition and Reliability Program (SHARP)**

The Standard Hardware Acquisition and Reliability Program (SHARP) is a Navy standardization initiative, using tri-service military standards, that addresses many electronics applications in both shipboard and airborne environments, including fighter/attack aircraft. SHARP is included in the acquisition process through MIL-STD-5400, as nondevelopment items (NDI), and is also mentioned in 500G.2.

SHARP comprises four separate standard product lines which include the Standard Electronic Module (SEM) initiative, the Standard Enclosure System (SES) initiative, the Standard Battery System (SBS) initiative, and the Standard Power Supply (SPS) initiative.

The SHARP is supported by both O&MN and RDT&E funding. The O&MN line sponsors the field activities to maintain the current documentation and data bases, as well as product periodic QPL testing. The RDT&E line sponsors the transition and demonstration of new technologies in the existing standard product initiatives of SEM,

SPS, SES, and SBS. Some of the major technologies include fiber optics and photonics, composite materials, and modeling. The use of the SHARP as the technology transition approach makes it possible to migrate higher risk technologies into the fleet via proven standard product specifications.

## 8-5 Hardware Interconnects and Connectors

### 8-5.1 Computer Backplanes

A computer backplane is a rack level interconnect board into which standard modules are plugged. The physical transmission media for the backplane is typically a multilayer dielectric board with deposited or screened-on metal interconnects. In order to increase the signal transmission speeds of the backplane interconnect, controlled impedance metallization approaches are utilized. The interconnect of the backplane typically operates at or near the clock rate of the processor. The interconnect can be point-to-point in nature or may be "bussed" or shared in a serial or parallel fashion. The precise physical and operational interconnect definition consisting of connector and pin designation, the backplane media and topology, data protocols, transmission speed and signal waveforms is referred to as a *backplane bus standard*.

Most computer backplanes consist of a passive transmission media with the appropriate connectors. Massively parallel processor development programs are currently exploring alternative active backplanes. These active backplanes use active switching devices to route signals from a master bus to provide a reconfigurable interconnect system between processing elements.

### 8-5.2 Electrical and Digital Connections

There are several basic types of connectors used, including blade and tuning fork, bristle brush, box and post, tulip and hyper tack. The blade and fork connector is a major connector used in military applications with the box and post a primary commercial interconnect. Figure 8-5.2 illustrates these basic conventional connector types.

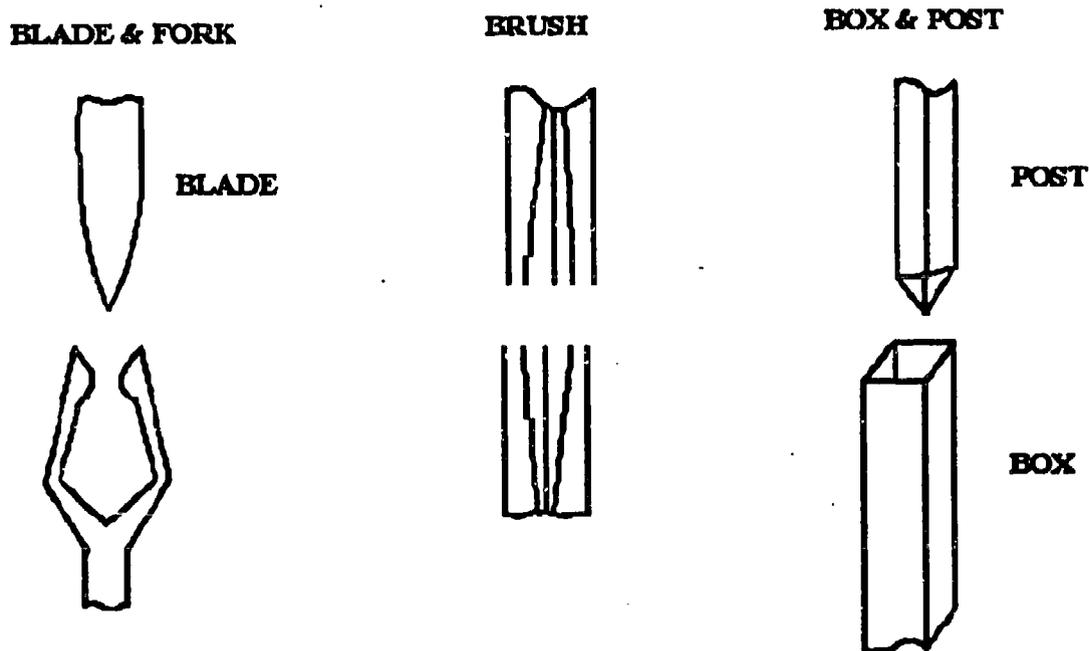


Figure 8-5.2 Typical avionics wiring connectors.

The main blade and tuning fork connector is a 396-pin version which will be used for Military Future Bus and the same pin style in a larger version will be used for commercial Future Bus, both leveraging the same contact style. These pin styles will be specified by EIA and called out by IEEE 1101.4. The brush style contact is a relative new contact technology. However, it seems in most testing to date to be equal to the blade and fork. There have been arguments for the brush based on better technology, but no testing is conclusive and both the blade/fork and brush contacts are acceptable for fighter/attack or helicopter environments. There are differences, i.e., the blade and fork has better weight and density characteristics when the brush has a lighter insertion force. No engineering evidence mandates the use of the brush style in lieu of the blade and fork styles which will be used on military and commercial Futurebus+ circuit cards.

It must be noted that circuit card connectors can, and have been, a major contributor to system failures; most due to some form of corrosion. Therefore, it is recommended that control be placed on connectors procured for severe military environments to ensure plating thickness and minimize susceptibility to corrosion. Specifically, connectors installed in attack/fighter aircraft or helicopters should be procured to military specifications to minimize system failures due to various corrosion mechanisms. Commercial connectors should be acceptable in transport aircraft depending on the criticality of the given system. A final feature that would be desired

from a military or commercial connector with high density is a solderless backplane Pin. This is required due to extremely thick backplanes driven by system interconnect density. These very thick backplanes will not allow adequate wicking of solder due to heating problems.

### 8-5.3 Technology Trends

All indications are that digital connectors will become denser, i.e., requiring more interconnects per module. Two promising technologies that allow high density are the gold dot and fuzz button shown in Figure 8-5.3. Both will allow more than two times the present density in interconnects. These contact technologies are currently being used and proven for solderless component connection to circuit cards. These two technologies offer many benefits in solderless assembler and ultra high performance electrical characteristics.

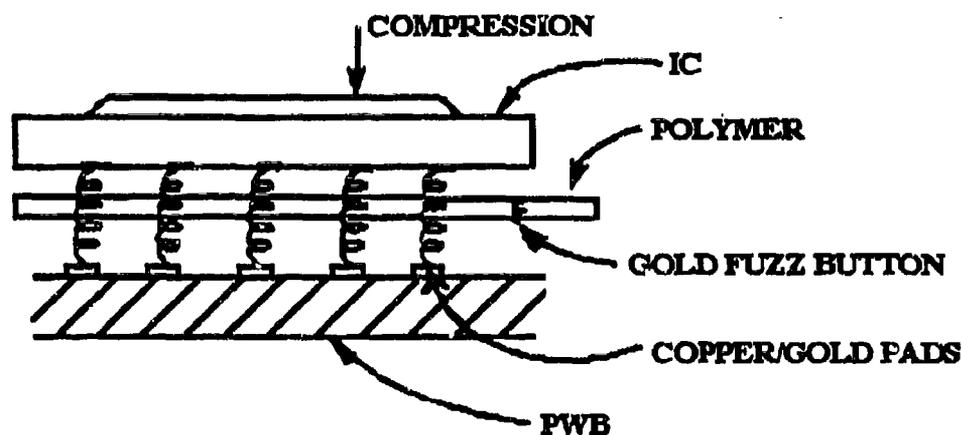


Figure 8-5.3 -Fuzz Button interconnects between an integrated circuit and a specialized printed wiring board.

It is expected that as optical interconnect technology progresses, optical technology will help slow the interconnect density requirements.

### 8-5.4 Optical Interconnects

Optical interconnect technology is beginning to become practical and cost effective for some installations. The primary circuit card optical interconnect technologies are butt type and lensed type. These optical termini are both currently available. The butt type is the lowest cost and has lowest optical signal loss. However, lensed termini are more tolerant to environment and mechanical misalignment. If losses can be tolerated, then lensed termini generally make the best solution for backplane to circuit card optical interconnect. The F-22 is pursuing a lensed termini which is V-groove ball type technology as shown in Figure 8-5.4.

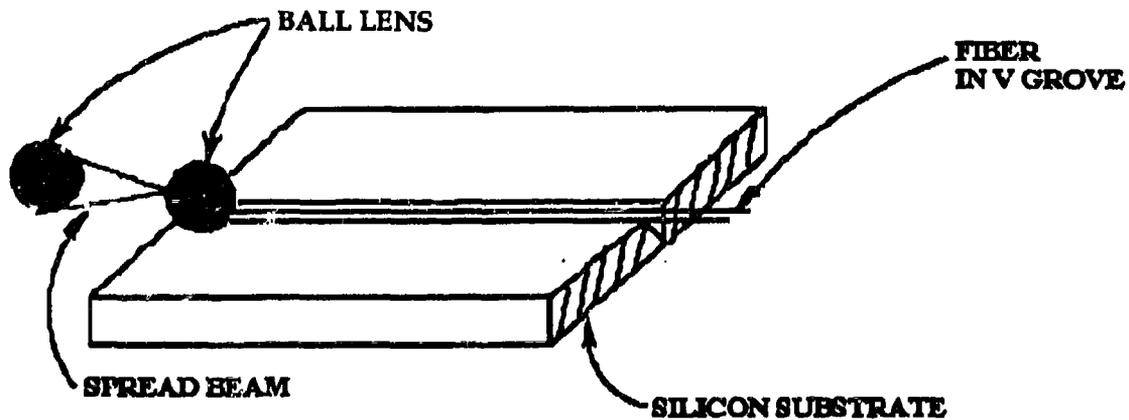


Figure 8-5.4 "V-grove" optical interconnect used in the F-22.

This type of termini is not mature, however. Risk is nonetheless regarded as low and the benefits of using multi-fibers in small areas are high. The F-22 optical termini work should be followed closely for possible standardization. It must be noted that standard termini can be procured from MIL-T-2904 and it would be effective to move this ball lens termini into existing standards.

Most experience with fiber optic termini have resulted in high cost in development and production due to the problems with fiber termination. Fiber termination seems to still be an art and not necessarily repeatable with automation of the process unlikely. To make this technology cost effective, research must be performed to eliminate the tedious polishing termination process now being done by humans. Another factor that raise cost in fiber optic application are the high cost of the fiber interface circuits, i.e., light transmitters and receivers. Experience has shown that these devices can, and typically are, sensitive to the environment which make application difficult in avionics. Effective application requires temperature compensation circuits and low noise circuit design and fabrication. These components can cost up to \$10K of the cost of a \$12K electronic module. Below are several conclusions that can be stated about current fiber optic termini.

- A. Close tolerances are required for termini alignment. Butt termini are very sensitive to misalignments in any direction. Expanded beam termini are designed to compensate for some axial separation. Both types of termini usually require an alignment sleeve for angular and side to side alignment.
- B. Connectors should provide environmental protection because the termini won't work if the interface is dirty or scratched.

- C. The connector needs to provide and maintain a compression force when using butt termini. The butt termini are usually designed to be touching. If the termini are separated by even a small amount, the light spreads before it reaches the next termini, resulting in high optical loss.

Some logistics considerations to be addressed when interconnecting fibers are as follows:

- A. Fiber type [single mode or multi-mode (graded or step index), bare fiber or jacketed].
- B. Termination procedure, installation procedure, repair procedure.
- C. Strain relief method to protect fiber from undue stresses (e.g., exceeding bend radius).
- D. Splice method for repairing in-line breaks.
- E. Special tools for any of the above procedures.
- F. Maintenance procedures, required training for technicians.
- G. fiber/cable routing on platform.

#### **8-5.5 Radio Frequency (RF) Elements**

Radio frequency (RF) connectors have been in use for decades. Today, however, RF circuit functions are migrating to module level packaging that is similar to digital packaging. This trend is evident in systems such as the INEWS and ICNIA. The avionics industry is moving towards system concepts that integrate RF type functions into racks and boxes with conventional digital processors. This approach requires that RF circuit cards (modules) have RF connectors that allow module insertion and extraction into backplanes. This feature is called "blindmatable" and places high demands on the performance of the connector. RF connectors are available for integration into modules that work up to 18 GHz with low losses. Little standardization has occurred in RF connectors. It is obvious as integration of RF functions continues that standardization of connectors must take place to allow multiple module application. Little evidence is available that commercial industry is pursuing RF applications requiring connectorization at this level. Therefore, it is unlikely that any commercial technology is applicable in these applications.

#### **8-6 Thermal Management**

It must be stressed that thermal management is more important in military than in commercial applications because fighter/attack aircraft and helicopters are expected to be immediately functional in both desert and arctic conditions. Most commercial equipment is not required to survive at these extreme temperatures. Therefore, thermal management is not considered as important. If a piece of commercial equipment has a thermal problem, one can usually add a fan or heater as required, but in the gunbay or electronics bay of an attack/fighter aircraft, one may not be able to effectively cool electronic boxes due to extremely high ambient temperatures.

### 8-6.1 Component and Module Level

The reliability of semiconductor devices decreases about two-fold for each ten degrees centigrade increase in temperature above a certain temperature. Therefore it is highly desirable to remove the heat generated in the electronic devices very effectively to an external heat sink. As the speed and density of circuits increase, this problem becomes more acute and must be solved. The device/package interface is key to this heat transfer. Exotic new materials such as synthetic diamond are being deposited on devices and packaging materials to improve this interface. The use of channels or grooves in the substrate to induce turbulent flow of the cooling air or fluid passing through these channels is also being pursued. In addition to the chip/package interface, the package to module interface must be optimized through novel materials or improved heat removal techniques. Heat pumps, liquid flow through, air flow-through or circuit immersion are concepts being explored. The optimum approach will permit the highest circuit densities and speeds while maintaining or improving the lifetime and reliability of the devices. Systems engineers must select the best materials and techniques for the intended application.

### 8-6.2 Standard Computer Modules

Standard module concepts with well defined interfaces have emerged as the preferred packaging concept for computers and other high density electronic functions in both the commercial and military sector. A module is a large area substrate (e.g., 4"x6" or 6"x9") composed of a thermally conductive material on which are mounted the semiconductor devices or "chips". The individual chips can contain from hundreds to hundreds-of-thousands to millions of transistors, and are usually hermetically sealed in ceramic packages with metal leads or "bumps". The leads of the ceramic chip packages or carriers are then soldered to an interconnect pattern consisting of deposited metal and insulator which has been defined on the module substrate surface. Thus a very complex electronic function can be implemented on a standard module which will have a standard multi-pin connector on one edge of the card. The sides of the card are usually clamped to a heat sink to keep the circuits at the desired temperature. These modules are then plugged into a "backplane" which sends signals in a parallel or serial fashion from module to module. The enclosure for these modules and backplane assemblies is referred to as an integrated rack and contains a controlled environment providing input power and cooling provided to the modular electronics inside.

### 8-6.3 Die and Component Level

Thermal issues at the die or component involve mainly removing enough heat from the die to keep die or junction temperatures low enough to ensure that thermal failure -- which is a major failure mechanism for semiconductors -- does not occur. One of the major reasons for the use of ceramics on military components is heat transfer which is usually much better than that of the plastic used on commercial components.

Trends seem to indicate the dies are getting bigger, denser and faster, which means that more power is concentrated in a small area. This power relates directly to the amount of heat that must be removed. Die stacking will also add to this thermal problem.

Diamond films which spread heat efficiently over a larger area are becoming viable, as well as liquid immersion cooling of the die which removes many of the thermal resistances between the die and the primary heat transfer medium. The use of liquid cooling, however, imposes maintenance problems which need to be fully understood prior to allowing this method. The best method for die cooling is still conduction to a

heat sink. This approach poses the lowest risk maintenance and logistics support. Liquid techniques at the die level must mature substantially in order to be viable. It must be noted that liquid cooling technology for die heat transfer has been demonstrated by SHARP and AAS&T and have achieved superior cooling effectiveness. However, the problems mentioned have yet to be solved.

#### 8-6.4 Circuit Card and Module Level

Thermal management at the circuit card or module level is where most emphasis is placed, because this level is the easiest to effect. The major cooling or thermal management techniques used at the card level include air impingement, conduction, air flow through thermal cores, and liquid flow through thermal cores. Air impingement is basically flowing cooling air over and between modules. This method is used substantially in both commercial and military applications. For more severe environments, conduction cooling cores are effective. Air and liquid flow through cooling cores are effective for use with high power modules over 50 watts, but are less mature. Liquid cooling assumes that the aircraft environmental control system can support it, which effectively limits it to new platforms or major retrofits. By far, the simplest and most cost effective method is still conduction cooling. The problem is the limitation on power level. We have seen this power rise drastically on the F-22 electronics where liquid cooling is required, but we also know of massively parallel computers in military SEM-E packages at 30 watts per module with Cray II power in four (4) modules far outperforming F-22 designs. This leads one to conclude that an extremely high performance computer architecture can be realized today without the burdens of liquid cooling on the aircraft.

#### 8-6.5 Common Cooling Methods

There are three common cooling schemes which are presently used to cool military electronic systems. These include the following: Direct air impingement cooling; Conduction of heat to an air cooled cardcage; and Conduction of heat to a liquid cooled card cage. Each of these schemes have advantages and disadvantages which must be considered before selecting a cooling approach.

Direct air impingement cools electronics by passing fan-forced air directly over the electronic module. This method of cooling can typically dissipate 20 to 35 watts on a SEM-E size module and maintain the electronics at acceptable temperatures. This method should only be used in applications where the air can be filtered sufficiently to remove contaminants in the air that may cause the module to fail. Direct air impingement is not allowed on avionics systems. Fan noise associated with this cooling scheme must be considered. The increased fan noise which is associated with increased air flow does not make this scheme a good candidate for cooling high power modules.

Conduction of heat to an air cooled card cage is a scheme that is commonly used in existing military systems. This scheme is commonly used on avionics systems where air is available at an acceptable temperature to provide sufficient cooling. Existing schemes using aluminum SEM-E heatsinks can effectively dissipate approximately 25 to 35 watts. The thermal properties of air and the increased fan noise do not make this cooling scheme a good candidate for cooling higher powered modules.

Conduction of heat to a liquid cooled card cage is desirable when there is a liquid coolant source available. This scheme is most commonly used on shipboard and submarine systems. This scheme is recommended when noise is an important

consideration. Using aluminum SEM-E heatsinks, existing systems typically can dissipate up to 40 watts per module and maintain the components at acceptable temperatures. Constructing the module with materials which have a higher thermal conductivity could allow this scheme to dissipate approximately 80 to 100 watts. This job could be accomplished using existing materials such as composite heatsinks, aluminum nitride, and diamond film. The use of these materials would also increase the module cost.

#### 8-6.6 Other Cooling Schemes

Other cooling schemes, which are not as common as the previous schemes, but offer increased cooling capabilities are available. These include the following: Hollow-core air flow through; Hollow-core liquid flow through and Direct liquid immersion in a dielectric fluid. Hollow-core air flow through provides cooling by passing air directly through a hollow module. A 0.125 inch thick SEM-E air flow through module could dissipate approximately 50 watts. To dissipate more power would require a thicker module.

Hollow-core liquid flow through provides cooling by passing liquid directly through a hollow module. Liquid quick disconnects allows the liquid to enter and exit the module. A SEM-E size module can dissipate between 200 and 500 watts. This scheme is capable of dissipating heat for densely packaged modules which are now resulting from the use of multichip modules, chip-on-board packaging, and the reduction of integrated circuit feature size. The placement of more electronics on one module results in smaller and lighter electronic systems. The use of quick disconnects allows the removal of individual modules for maintenance purposes. This cooling scheme may have a greater risk of leaking fluid than liquid cardcage cooling. There are now several studies investigating the reliability of the liquid flow through scheme. The thermal performance of this scheme will be dependent strongly on the liquid used and the module flow rate. This cooling scheme is best suited for systems with high power modules or for systems that may have high power modules in future upgrades.

Direct liquid immersion cools electronics by submersing them in a dielectric fluid. Fluorinert, a liquid manufactured by 3M Company, is used in this manner in some commercial applications. This cooling scheme has not yet been applied to military systems. This scheme is best suited to chip-on-board packaging schemes where the actual electronic chip will be in contact with dielectric liquid. Because the Fluorinert has relatively poor thermal properties, boiling must result to provide a thermal capability similar to that obtained by liquid flow through schemes.

#### 8-6.7 New Cooling Technologies

The two main cooling technologies under exploration for modules include composite cores and liquid immersion at the module level. Composite cooling cores use carbon fibers to effectively direct conducted heat off the module. With this conduction cooling technology, and the new massively parallel computer technology such as nCUBE, an argument can be made that we need not use liquid cooling for these computing elements. (Note: liquid cooling may be more of a necessity in the high power RF area than in digital or signal processors.)

AAS&T is exploring liquid immersion cooling at the module which may allow 1,500 watt modules. Currently computation requirements can be solved by proper architecture, and not necessarily speed and increased power. It therefore seems that this

1500 watt module would most likely be used in the RF area (such as a high power transmit-receiver module using very dense MMIC technology).

### 8-6.8 Systems Level (Enclosure/Integrated Rack)

Thermal management at the system or box level usually involves flowing air or liquid over or in the box to remove heat. Air impingement, conduction to air heat exchangers, conduction to liquid heat exchangers, air flow through, liquid flow, and free air convection. In general, these methods support the cooling technique utilized at the module level. Air impingement is typically not desirable due to contamination in military environments; however, as mentioned previously, it is heavily used in commercial applications. Free air convection is most desirable due to its simplicity, i.e., no fans, pumps or ECS connection, but it typically is limited to current technology. System power requirements are becoming too high in military applications for free air convection to be effective. One of the most common used methods is conduction to air heat exchanger. In this method, heat is conducted off the module to a heat exchanger located in the box which is cooled by passing ECS air through it. This method isolates the module from contamination and is efficient. Conduction to liquid heat exchanger cooling is identical to air except liquid transfers heat, allowing smaller heat exchangers and more power in the box which equates to more power in less volume. Again, restriction to new aircraft is an issue due to the liquid ECS requirement. Air flow through technology and liquid flow through technology for boxes are similar to their heat exchanger counterparts, except the heat exchangers become manifolds to the modules where the module core becomes the heat exchanger which is more efficient. Generally, the following order can be used to determine efficiency:

1. Free air convection (for lowest power disipation)
2. Air impingement
3. Conduction to heat exchanger
4. Flow through air
5. Flow through liquid (for highest power disipation)

Technology of semiconductors seems to be pushing us toward liquid solutions, but the push is also driven by the F-22. Air cooled massively parallel options do exist and could be exploited on both new and existing aircraft. With regards to commercial enclosures, their utilization potential would be restricted to those environments where free air and air impingement is acceptable, which would likely be transport aircraft only - not attack/fighter or helicopter.

### 8-7 Structural

In general, structural problems are solved at the circuit card and box (enclosure) level. Structural problems usually arise from shock, vibration and thermal expansion, with shock and vibration being the major contributors if surface mounted components are not used. Shock and vibration environments are substantially different for commercial and military applications, with military being much more stringent. Further distinction can be made in these environments between transport, fighter/attack aircraft and helicopters. Transport aircraft typically have low shock and vibration requirements, except propeller aircraft at the engine. In general, fighter/attack aircraft and helicopters

have high vibration levels. They also have significant shock requirements if used for carrier landing. Commercial equipment usually will require some form of protection to survive (e.g., vibration isolation). However, in the more severe environments it is unlikely that any added protection would be sufficient.

In general, military modules are designed with a structural reinforcing member, usually an aluminum web or housing, where commercial equipment have no module reinforcement other than the printed wiring board. Also, commercial boxes and enclosures have little structural integrity to survive in fighter/attack aircraft or helicopters. Enclosures for these are designed to insure that vibration and shock cause no damage and they usually have substantially more reinforcement structure to obtain this goal.

## **8-8 Advanced Power Systems**

Technology for advanced avionics systems is moving towards smaller device geometries, higher functional density, and higher power densities. Much of the newer devices will operate from either 3.3 VDC or 1.5 VDC power supplies, rather than the 5.0 VDC that has been standard since the mid-1960s. Higher power levels, coupled with lower operating voltages, results in higher current requirements and more difficult voltage regulation problems. Power conversion technology for advanced systems may be substantially different from the SEM power supply modules.

### **8-8.1 Power Supplies and Power Distribution**

Equipment under E&MD presently are in many instances using the SEM E packaging scheme. Power supplies are also being packaged in SEM E modules. System power requirements vary from hundreds of watts to several thousands watts, and in some cases up to tens of kilowatts. The connectors for SEM modules have from 250 to 396 pins that are each rated at around 3 to 5 amperes. To conduct a high power supply current from the module requires using many of these pins connected in parallel. As the system power requirements increase (usually with a corresponding weight decrease), more and more pins must be used. The SEM E connector is one of the limiting factors in increasing the output current and power for a SEM E power supply.

Assuming that the power supply module is adequately cooled, presently available SEM E power supply connectors limit the output current to about 100 amperes. The current is limited by the number of pins required for control signals, status signals, test points brought out through the connector, power dissipation in the connector, and the current handling capacity of each pin.

SEM E power supply modules with higher output currents must utilize special connections. These connections might be a copper bus bar that is bolted to a larger distribution bus (Texas Instruments is using such a technique in the F-22 2.5KW SEM like power supply module for the radar array).

### **8-8.2 Modular Distributed Power Systems Architecture**

Modular distributed power systems consist of SEM E or SEM-like power supply modules distributed in an avionics enclosure among the "load modules" (processors, memory, interface modules, other). The power supply modules can be paralleled to share current to the load modules. Different levels of redundancy can be accomplished with such paralleling. Figure 8-8.2 shows a typical on module power distribution system.

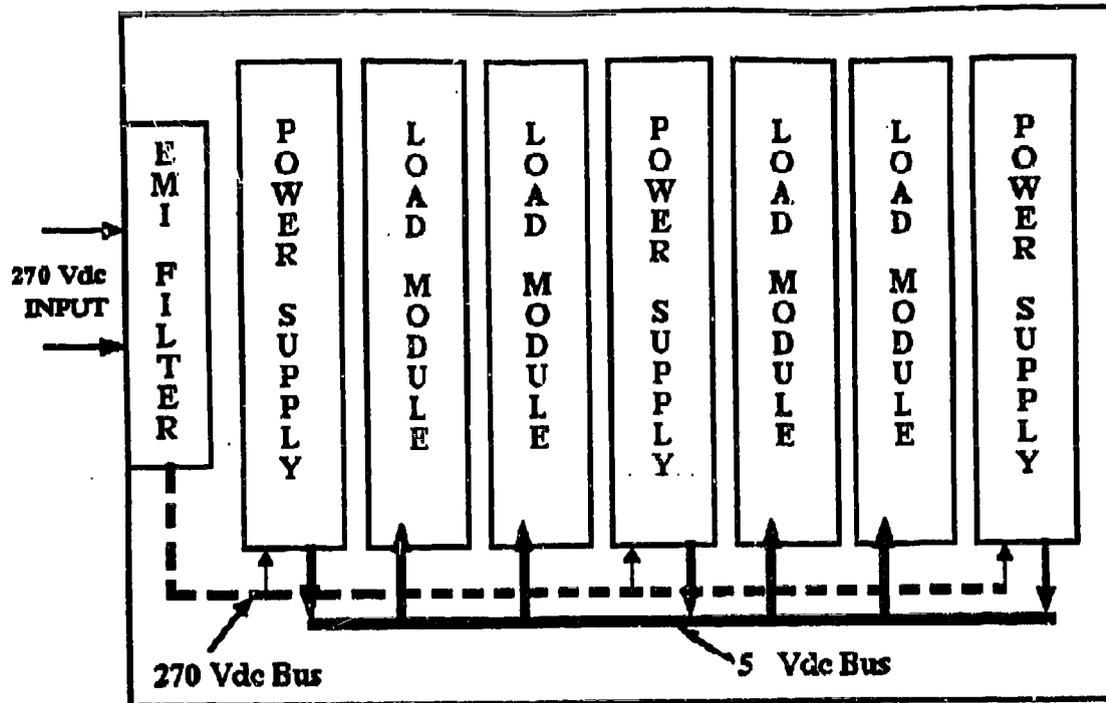


Figure 8-8.2 Typical modular distributed power system.

### 8-8.3 On-module Distributed Architecture

On-module power systems are conceived to consist of very small, very high density power converters that will be mounted on each load module to provide the proper voltages for that module. Some intermediate voltage will be distributed to each module in the avionics enclosure. This voltage might be 50 VDC. This intermediate voltage is then converted on each load module to 5.0 VDC, 3.3 VDC, 1.5 VDC or other voltage required by each module. The aircraft power into the enclosure is filtered then converted to the intermediate voltage for distribution to each module. Figure 8-8.3 shows a typical on module power distribution system.

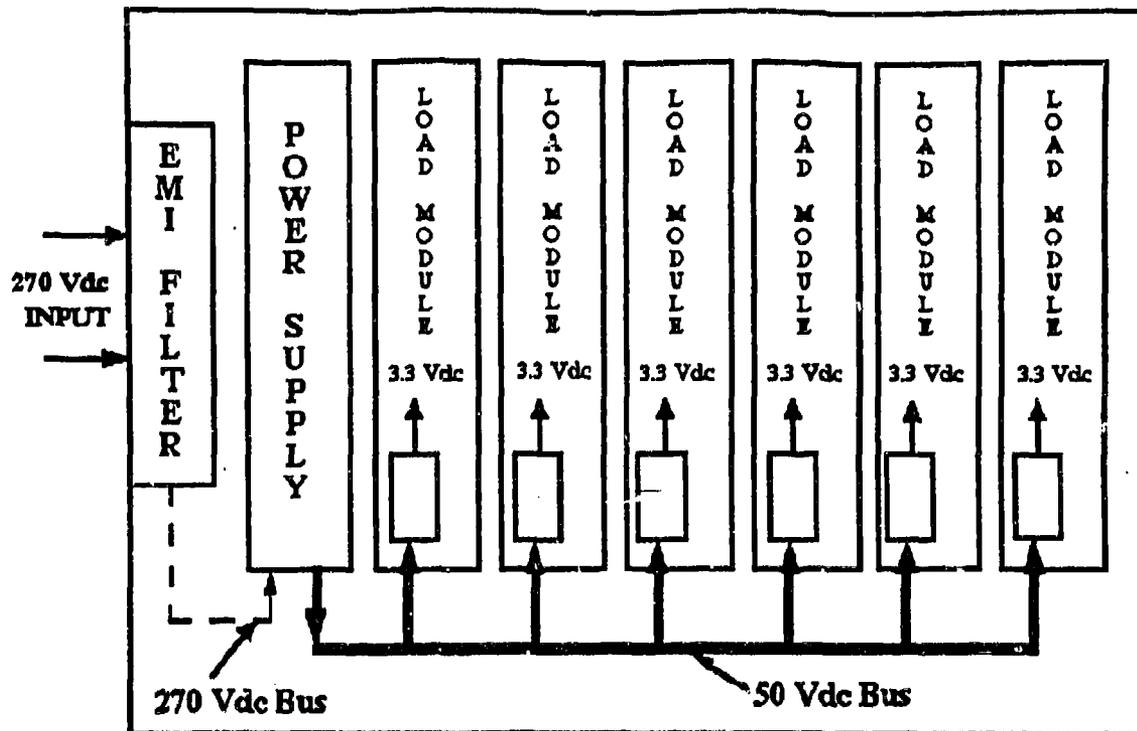


Figure 8-8.3 Typical on-module distributed power system.

#### 8-8.4 On-Module vs Modular

For low voltage systems (3.3 or 1.5 VDC) losses through the connector, through the backplane, and again through the load module connector are critical. The on-module distributed power converters should provide a well regulated voltage at the point of use. Also if the power requirements of each load module increase (or stay level) and the voltage is decreased to 3.3 or 1.5 VDC, then the on module converters should help reduce the pin count required to carry the increased current levels.

Existing SEM E power supplies range in output power density from 10 watts per cubic inch to 25 watts per cubic inch. Liquid flow through power supply modules are expected to achieve 100 to 150 watts per cubic inch.

Adding an on-module power converter to each module will reduce the available board area for that module; however, more slots will be available since the number of power supply modules will be reduced. If a 200 watt module is to accommodate an on-module converter, then the module will be required to dissipate the additional power dissipated by the converter. Existing SEM E power supply modules have efficiencies ranging from 80 to 85 percent. If the on-module converters achieve the same efficiencies, then the additional power dissipation will range from 35 to 50 watts.

### 8-8.5 Technology Required for On-module Power Converters

The technology developments required to realize the goals of the on-module power systems include: 1) power devices (switch devices, rectifier devices), 2) magnetic components, 3) circuit techniques, and 4) advanced packaging technologies.

The power losses in the power switch and rectifier devices becomes more critical when the output voltage decreases to 3.3 or 1.5 VDC. The voltage drop across the rectifier circuits become a significant percentage of the output voltage. Lower drop devices or rectifier circuits need development.

Approaches to achieving higher density packaging include innovative packaging techniques and high frequency power conversion topologies. Some power supply developments are aimed at increasing the conversion frequency into the megahertz range. These topologies include some type of resonant circuits (for zero voltage or zero current switching). Theoretically, as the frequency increases magnetic core materials decrease in volume to the point where the core is no longer required. The circuits are expected to be packaged using RF packaging techniques (strip line circuits).

Rectifier devices become more critical when the output voltage decreases to 3.3 or 1.5 VDC. The voltage drop across the rectifier circuits become a significant percentage of the output voltage. Lower drop devices or rectifier circuits need development.

### 8-9 DC Power Supplies

With the rapid increase in gate packing density and advanced packaging techniques such as multi-chip modules (MCM), electronic systems continue to shrink in size. Power supplies, however, have not kept pace, even though notable improvements have been made. They are made of magnetic and other passive components which cannot benefit from the advances in the semiconductor technology. In today's advanced systems, power supplies are packaged with conventional components. Typical DC power supply densities range from 1 to 3 watts per cubic inch (w/cu. in.) and represent about 20% of the system volume. However, with submicron logic devices and MCM packaging techniques, size and performance requirements will dictate 100 w/cu. in. density and 95% efficiency by the late 1990s. If these improvements are not made, the size and weight reductions promised by the submicron devices and MCM packaging will not be realized.

In order to accomplish the dual goal of 100 w/cu. in. power density and 95% efficiency, NRaD, in association with TI, has developed a topology that substantially reduces switching losses by raising the operating frequency (10-25X) into the 2 to 5 MHz range. The additional specification of 1.5 V and 3.3 V supply voltages, (compatible with 100 to 300 nm VLSICs), was a technological challenge in the development of the output rectifiers to maintain the 95% efficiency. Traditionally, a silicon Schottky diode with 0.4 V drop is used. The NRaD-TI team has developed a GaAs vertical field effect transistor (VFET), reducing the rectifier voltage drop to 0.014 V (from 0.4 V) and provided rectification efficiencies of 99% (instead of 80%). The potential of GaAs VFETs for microwave power amplification in the 0.8 to 4.0 GHz range is projected to provide 100 watts at 1 GHz.

Program managers need to support the continued need for power supply efforts such as the Navy/TI effort describe above or other promising approaches. It is feasible to increase the power supply densities to several hundred watts per cubic inch and eventually to one kilowatt per cubic inch.

Such power supplies packaged in an MCM format will lend themselves to a more distributed power supply and conditioning concept in advanced modular avionics architectures. Such power supplies are likely to be spread throughout an aircraft in the future. Program managers need to be aware that these advanced power supplies may not be compatible with retrofit into older aircraft due to a mismatch in prime power requirements. Also, subsystems which incorporate these distributed power supplies may require substantial redesign to operate in older aircraft.

## 9-0 EFFECTIVE SOFTWARE DEVELOPMENT AND MANAGEMENT

Today's modern Naval aircraft weapon systems incorporate complex avionics systems whose core functions are highly software-intensive and are very dependent on the real-time processing of information and data. One of the lessons that has been learned over the last 30 years of building Naval avionics systems is that when designing an avionics system, a systems engineering approach must be utilized. The initial architecture planning for an avionics system must consider the requirements that the weapon system must meet. The systems engineer must determine the architecture for the avionics system and decide what functions should be implemented in hardware and what functions should be implemented in software.

The delegation of functions/tasks between hardware and software is a systems engineering activity which must be based on a knowledge of the capabilities of available hardware and software technology/components when the system is designed. Functions which typically don't change over the life of a system or are time critical are normally implemented in hardware. This hardware could be strictly analog or may have deeply embedded software, i.e., firmware, that is not expected to change over the system's life. Functions which will be updated or changed frequently are usually implemented in software. Those functions/tasks where the software implementation would result in inadequate/marginal performance should be implemented in hardware. To achieve maximum flexibility, as many functions/tasks as possible should be implemented in software.

In today's systems the development of the software is often more of a cost, schedule, and risk driver at the program level than is the avionics hardware. Many program managers, both government and contractors, underestimate the size and complexity of the software development tasks and assume them to be low risk. Many of the DoD standards and accompanying guidance for managing software developments are not specified as requirements in contracts or if specified, either waived or ignored during the performance of the contracts. In the past, contracts have been put into place with incomplete software development requirements; which limit the government's access to software cost, schedule, and performance information; and which restrict the government's ability to require corrective actions, even when critical problems become evident. Shortcuts have been taken during the development phase which resulted in increased life cycle hardware and software maintenance costs.

There have been numerous studies and reports over the years that have addressed the software problems/issues involved in developing and maintaining software for DoD systems. These studies were accomplished by highly knowledgeable and credible individuals and organizations and provide a basis from which specific actions should be taken at the DoD level. Annex E to the draft 1990 DoD Software Master Plan presented a consolidated perspective of the status of related DoD activities at that time and provides a comprehensive list of software issues which should be addressed. DoDINST 5000.2 sets forth requirements in Part 6, Section D, Computer Resources, which program managers must address during the acquisition process. These computer resources requirements, if properly considered/applied during the acquisition process, could result in the majority of the software problems/issues which were enumerated in the draft 1990 DoD Software Master Plan being addressed during the acquisition cycle of a weapon system.

## 9-1 Real-time Computer Resources Development Dependency on the Systems Engineering Process

Although software is perceived by some as distinct from hardware, it does not stand alone, and must be viewed as a part of the weapon system. In order for software to perform its function, the necessary hardware and interfaces must be provided in the system design. The software engineering process is part of the overall systems engineering process which is in turn part of the weapon systems acquisition process. There are unique characteristics associated with software that preclude its being developed and maintained in a manner that is satisfactory for hardware. Software has no process corresponding to the hardware manufacturing process. For hardware, all of the baselined equipments must be produced on the manufacturing floor per the approved engineering drawings; whereas for software, it is only duplicated and incorporated in each of the baselined weapon systems. During the Post Deployment Software Support (PDSS) phase, new functions can be added to the weapon system through software only changes provided the system was initially designed to accommodate known/projected future requirements.

Successful weapon system development requires a systems engineering, requirements oriented approach to correctly resolve and perform engineering tradeoffs involving the distribution of functionality between hardware and software for computer resources which are deeply embedded within the weapon system. The selection of a hardware architecture must provide the necessary computation power/memory and required reserve capacity. The necessary engineering processes and procedures, both hardware and software, for accomplishing the required engineering tasks from initial design into analysis, implementation, integration and test, and through production should be in place at the contractor's facility or must be established. The hardware/software tradeoffs which are made during the development phase are critical if life cycle hardware and software maintenance costs are to be minimized.

Specifications for computer resources should be generated only after a methodical systems engineering process has been accomplished including requirements traceability from system performance requirements down to detail design specifications. This process includes aircraft/weapon unique critical performance requirements such as environmental, space, configuration and access limitation, electronic compatibility, carrier suitability, power, weight, and cooling. Factors which must be considered when making the hardware/software determination in real-time avionics systems are: available technology, update requirements over the life of the weapon system, ease with which new functions can be incorporated, known/projected future requirements, security, redundancy, degraded modes of operation due to battle damage or hardware failure, error handling in the event the software performs out of specification, loss of electrical power and start-up requirements when the power is restored, timing constraints, bandwidth of buses and input/output channels, and maintenance concept.

Many of the problems found in existing avionics weapon system software were driven by the "systems engineering" requirement to optimize hardware resources for things like size, weight, and power. Memory has been sized too small and processor speed has not been adequate to perform all the required functions/tasks in an unconstrained manner. One of the prime goals of systems engineering is to separate the requirements from the "nice to have" when determining the hardware/software requirements of the weapon system being designed. In order for the software to be properly designed, the necessary hardware/buses/interconnects must be provided for in the initial selection of the computer resources. Hardware decisions should not be finalized until the software design is mature enough to minimize the risk of producing an

avionics system which does not or just marginally meets the weapon system requirements.

Many developers mistakenly assume the application of computerized management tools, Computer Aided Software Engineering (CASE) tools, and analysis tools will provide or improve the implementation of a more definitive system development capability. If the engineering capability did not exist at the developer's facility in a well established manual form before the application of automated tools, the subsequent effort to create an automated engineering environment will be more difficult, take longer, have fewer benefits and cost substantially more. An important capability that must exist at the system level is the ability to systematically track requirements from one level of development to another, from one stage of the system life cycle to another, and between hardware and software Configuration Items (CIs). Computer-based tools to support the systems engineering process for today's complex systems are being developed and used. Current tools include the JIAWG/F-22 Software/System Engineering Environment (S/SEE) (Boeing D&SG) and *Teamwork-T<sup>m</sup>/RqT* (Requirements Trace), and *Teamwork/SA* (Alliant).

## 9-2 Computer Resources Open Systems Architecture

In-service avionics systems were built using the latest technology that was available at the time they were designed. Retrofitted systems often were no more than extrapolations of the same architecture that was used when the system was first put in service. Funding was never available to completely re-design an avionics system and re-equip a model/type of aircraft. As a result, new computer hardware was added and programmed to perform the required functions within the limits of what the computer hardware could handle and funds could provide. The computer hardware often fell short of providing the long term growth computer resources reserves which were necessary to perform PDSS. The results were that during PDSS, the software was difficult to maintain and enhance.

Even with the new technologies that are available today, the lack of funding still continues to heavily influence avionics system up-grading and the ability to migrate to the open system concept. The architecture of new aircraft avionics systems which are being developed today should be capable of future growth and technology insertion based on an open system design.

### 9-2.1 Understanding Open Systems

To have an avionics system based on the open system concept, an understanding of what is required is necessary. Open systems mean multiple standards. Just choosing existing standards does not guarantee portability or interoperability, nor does it guarantee easy integration of multi-vendor products.

In order to have an avionics system which is based on the open system concept, its design must be based on a set of open system standards. This is meant to ensure multi-vendor hardware/software interoperability and portability. Unfortunately, the necessary avionics standards are not yet defined for all the functional areas needed for real-time avionics systems. Additionally, existing standards frequently contain multiple options and systems which were built using those standards could contain hardware/software incompatibilities. For example, depending on which options of a

standard were selected, two avionics systems could result that are compliant with the same standard but contain hardware/software that can not be interchanged.

The requirements for the avionics system must include the requirement that the product baselined avionics system contain provisions for expansion or upgrading through the incorporation of additional or higher performance hardware/software. If this is to be feasible, all aspects of interfaces, including physical, electrical, functional, timing, protocols, and control must be defined. Any software control functions required to accomplish this must be included in the initial design.

### 9-2.2 Hardware Architectures

For a discussion of system/hardware architectures, see Parts 2, 3, and 4 of this Volume.

### 9-2.3 Software Architectures

Avionics systems must be designed to accommodate new mission needs, threats, and technology. An avionics system software architecture is a key determinant of the avionics system adaptability. A well-conceived and well-maintained architecture allows for reusable components (both hardware and software) to be smoothly integrated in the original development, custom components to be smoothly integrated via standard protocols, and improved components to be incorporated as replacements or enhancements are needed. To be useful, the software architecture must first be established and include provisions for change; second, it must be controlled and maintained throughout the avionics system life cycle.

The software architecture which is implemented in an avionics system is very dependent on the system architecture selected and the resulting hardware used to implement it. If the selected system architecture/hardware provides an open system architecture and the necessary computer hardware resources, a software architecture can be implemented which allows for future growth and technology advancement.

Attention to software architecture begins with the very first discussion of the weapons system scope and concept, and extends through the maintenance phase of the avionics system. The periods in an avionics system life critical to establishing and retaining a good architecture extends from the formal notification to industry of NAVAIR's need for the weapon system, through the evaluation of industry's response, the sequence of design reviews during the developer's design and implementation, and the long term maintenance of the avionics system.

Some attributes of the software architecture may overlap with concepts which would be considered part of the avionics system architecture. Those attributes of the avionics system architecture need to be established before the software architecture can be evaluated and controlled. The specific content of what is controlled under the software architecture for a specific aircraft avionics system must be established. Attributes should be considered architecture when they express relationships that contribute to the avionics system long-term tolerance to changes; they should be considered design when they are implementation specific.

General areas of consideration to consider when establishing an avionics system software architecture include the following:

### 9-2.3.1 Software Control/Partitioning

The software architecture must make provisions for several levels of control/tasks. These include:

- **Operating System Task** - functions and services such as provided in the Portable Operating System Interfaces (POSIX) should be provided.
- **System Manager Task** - functions to be performed for both software and hardware include:
  - 1) initialization/start-up
  - 2) integrated diagnostics both on-line and on-demand
  - 3) reconfiguration
  - 4) degradation/error handling/exception handling
- **Applications Software Tasks** - specific functions such as navigation, sensor management, communications, weapon launch, displays, database management, etc.

### 9-2.3.2 Flow of Data

The software architecture must provide for the orderly flow of data throughout the avionics system. Consideration must be given as to how data storage and retrieval will be accomplished, how data structures will be modified, and what processes may use the data.

### 9-2.3.3 Timing and Throughput

The software architecture must provide the necessary structure to meet the real-time requirements of the avionics system.

### 9-2.3.4 Interfacing Layering and Protocol Standards

The software architecture must provide for the establishment of open system protocols and interfacing layers. Interfaces which may need to be provided for include: pilot/crew/maintenance personal, operating system, display/graphics, networking/communications, sensors, security, data base management, and software support environment.

### 9-2.3.5 Software Security and Safety

The software architecture must provide security to guard against the compromise of the weapon system from both external and internal threats. As the threats and the complexity have increased, so has the burden of the software to ensure security with more complex, multilevel algorithms. Systems safety requirements can be met with a combination of hardware and software. Software is responsible for warning the pilot/crew of unsafe conditions, and allowing or assisting the pilot/crew to rectify the situation or take alternative courses of action. Software may be required to restrict operation of certain physical subsystems for their own preservation as well as that of the pilot/crew/maintenance personnel. The software architecture must permit software to be designed so that it does not create hazards through its normal or abnormal operation.

### 9-2.3.6 Software Portability

The software architecture must provide for the future incorporation of new software, easily merged with previously developed "legacy software", whether it was created by either the government or industry. This appears to depend on the use of both good systems engineering and the use of an open system approach.

### 9-2.3.7 Reuse

The software architecture must provide for the use and development of reusable software and the use of reusable hardware components. The software developed under this architecture should be available for use in other avionics systems using an open systems approach.

### 9-2.3.8 Verification and Validation Test

The software architecture must facilitate the comprehensive testing of software functions and those hardware items which were designed to allow testing to be accomplished by software.

### 9-2.3.9 Technology Insertion

The software architecture should facilitate the future insertion of both new hardware and software. Software changes such as new functions using Ada 9X, new operating system technology, new security technology such as better isolation/partitioning of memory using software techniques, enhanced performance of new signal and data processing algorithms, etc. must be provided for.

### 9-2.3.10 Growth

The software architecture must support both the increase in the size of existing software components and the addition of new software components. It must support and facilitate changes, additions, deletions, and system growth for the life cycle of the weapon system, whether they occur due to hardware and/or software modifications. The architecture must enable software components to be designed so that, in addition to be usable in the current system, they are also transportable to new higher speed hardware.

### 9-2.3.11 Embedded Training

The software architecture should provide for the incorporation of functions/tasks such that air crew training can be performed on the aircraft.

## **9-3 Computer Software Development**

Software provides system designers with a flexible means to adjust system elements and alter performance capabilities. Such changes don't impinge negatively upon system weight, power, or apparent reliability, since no hardware changes are required. In addition, there is no physical "wear-out" mechanism, so software appears not to require the same logistics tail as hardware. However, when software changes are made without regard for the effects on the avionics system, the side effects caused by such changes can lead to severe problems at the system level.

Once the avionics system and software architectures have been determined, several software design techniques exist which may be utilized to implement the detail implementation of the design. Both structured design and object oriented design techniques should be considered.

### **9-3.1 Software Design Methodologies**

#### 9-3.1.1 Structured Design

Structured design techniques have been used over the past several decades for developing software systems. Notable shortcomings have prompted the emergence, almost at the grass-roots level, of alternative engineering approaches that have demonstrably better results in producing software-intensive systems. While the new techniques must be examined for usefulness, it must be remembered that the present avionics software and systems engineering standards were built on engineering principles strongly aligned with structured engineering techniques. Structured design techniques should not be dismissed as unsuitable or unreliable until something which is proven to be better is available. The present techniques have been successfully employed on numerous avionics system programs and will continue to serve a wide segment of the avionics community. Thus it is probable that a blend of old and new design techniques will eventually emerge.

#### 9-3.1.2 Object Oriented Design

One new technique that is emerging is the object oriented methodology. Three software development phases must be considered in the application of the object-oriented technique; Object-Oriented Analysis (OOA) for software requirements analysis, Object-Oriented Design (OOD) for software design, and Object-Oriented Programming (OOP) for producing the software. The basic assumption of an object oriented development is that there will be an object oriented run-time environment executing an object oriented executable program. Ada is the programming language mandated for use in all DoD computer resources application; however, Ada is not an object oriented language, but rather an object based language. As such it supports some object oriented concepts but not all. This introduces constraints in the application of object oriented techniques for DoD computer resources applications. Industry and the Ada community have developed various strategies for dealing with the incompatibilities of object-oriented technologies and Ada.

The basic strategy is to use an OOA methodology such as Project Technology's Object-Oriented Real-Time Analysis or Rumbaugh's Object Modeling Technique, to perform software requirements analysis. These methodologies are supported by CASE tools such as Cadre's Teamwork or IDE's Software Through Pictures.

In moving from requirements analysis to design, two approaches are employed when Ada is the target language. One approach is to continue the Object-Oriented development into design using an OOD methodology such as Hierarchical Object Oriented Design (HOOD), Project Technology's Object Oriented Design Language, or Rumbaugh's Object Modeling Technique (applies to both analysis and design). These methodologies are supported by CASE tools. Due to Ada being the target language, there are limitations to the full application of Object-Oriented techniques which must be factored as constraints during design. A second approach is to switch to an Ada oriented design method such as Booch's or Buhr's Ada Modeling techniques. Finally, Ada is used to implement the design. It should be noted that Ada 9X will provide support of object-oriented concepts, but the implications of this capability to real-time software development is unknown at this time. Examples of commercial languages which support OOP are C++, Smalltalk, and Eiffel.

Even though there are methods to realize object-oriented solutions using Ada, any contractor/government agency attempting to employ this type of development approach will incur the following risks especially for complex real-time avionics software. First, object-oriented solutions using Object-Oriented languages with the associated run-time environment have been shown to not be particularly well suited to real-time applications. This occurs primarily because a context switch is required to operate on each instance of an object. When many objects are involved the number of context switches introduces overhead processing which can significantly affect processor loading. In addition, many commercial object oriented languages and environments do not provide scheduling, prioritization, and synchronization of software components which are critical requirements for real-time software. This issue is currently not a large problem with Ada's run-time environment since it is not object-oriented, but could become a problem with Ada 9X depending on its object-oriented concept implementation. Second, object oriented languages and run-time environments are not particularly well suited to error propagation/resolution due to the concept of inheritance. Inheritance basically allows general data definition via superclasses with refinement through subclasses to object instances. An object oriented environment works its way through the class structure to execute specific object instances. An error detected at lower levels cannot be propagated because the higher levels have an abstracted knowledge of the data. Thus the higher levels do not have knowledge of lower level errors. Resolution of this issue is not trivial. This is of particular importance for avionics software where safety associated with software integrity is a critical issue. (Note: This is one of Ada's strong suits with its typing model and exception handling capabilities. It is unclear how Ada 9X deals with this issue.) Third, DOD-STD-2167A is oriented toward a functional decomposition of software into Computer Software Components (CSCs) and Computer Software Units (CSUs) which does not particularly mesh well with an object-oriented development. Issues here primarily deal with CSC/CSU identification and associated integration test.

Object-oriented methods represents a basic paradigm shift relative to functional decomposition methods both in terms of software engineering techniques and software management. Generally, retraining is required for all personnel involved with a software development including system engineering, software engineering, configuration management, and software quality assurance. Evidence clearly indicates that object-oriented techniques for design, analysis and coding provide substantial improvements in many aspects of software engineering. However, object oriented techniques have their

own special problems as discussed above and have not been validated on large software programs such as are typical in avionics weapons systems.

### 9-3.2 Software Standards and Practices

Avionics system software is managed, designed, developed, tested, and documented per DOD-STD-2167A and DOD-STD-2168 and their associated data item descriptions. A Software Development Plan (SDP) is required as part of contract proposals. Navy field activities who are performing PDSS also normally provide SDPs as part of their Work Unit plans. SDPs should contain planning for continuous software engineering process improvement and computer resources risk management. For avionics systems in the PDSS phase that were developed using other than DoD-STD-2167A, major software block upgrades to existing weapon systems software are designed, developed, tested, and documented using DOD-STD-2167A and DOD-STD-2168.

Present standards efforts require that certain approaches be followed when designing and building avionics systems software. Present requirements mandate that a specifications-driven approach be used that develops and manages documents as the primary means of controlling and managing the project requirements, design, and product delivery.

Certain military standards (including DOD-STD-2167A, MIL-STD-499, DOD-STD-2168, etc.) which affect the development of software are currently being updated. The planned updates will provide improvements in managing and developing software. These new standards differ from the traditional structured design practices that permeated every aspect of previous standards documents and actually inhibited utilization of new methods.

DOD-STD-2167A, "Defense System Software Development", dated 29 February 1988, is the current standard for developing avionics system software. It calls for a detailed process with specific deliverables. NAVAIR is responsible for tailoring both the requirements and deliverables for a specific avionics system software. DOD-STD-2167A supports a top down, structured method (sometimes called the "waterfall"), however it can be used for other methods such as spiral, evolutionary acquisition (see DODINST 5000.2), etc. The spiral development methodology is used to overcome problems during a development such as incomplete requirements at project initiation. The method revolves around building enough to test and compare against high level requirements. With the knowledge gained, more detailed requirements are generated and the spiral methodology is repeated.

A new standard for software development MIL-STD-SDD, "Software Development and Documentation", is being developed and will supersede DOD-STD-2167A. It merges DOD-STD-2167A and DOD-STD-7935A, "DoD Automated Information System (AIS) Documentation Standard", dated 31 October 1988. DOD-STD-7935A covers all types of technical documentation for AISs, applications computer programs, and revisions thereto; as well as the use of existing or developed standards for each document type. MIL-STD-SDD will be a standard that is applicable to the software development process throughout the system's life cycle and will be applicable to both weapon system software and AIS developments. It defines standard terminology, top-level activities, tasks, and products for a software development or PDSS. It provides for transition from the development phase to PDSS. The standard does not provide or encourage any life cycle model (waterfall, spiral, evolutionary, or other). The standard contains the building blocks needed to create a life cycle model for a software project.

The standard is independent of any software engineering or management method. Application of the standard does not depend upon any computer design/programming language. It streamlines the formal review/audit process so that non-waterfall models can be employed. It requires that software "components" be partitioned out of a system design effort and, when put under Configuration Management (CM), be called computer software configuration items (CSCIs). It requires the evaluation and incorporation of reusable software when it meets user needs, complies with data rights, and is cost effective. It requires software metrics at the general level without specific realization of those software metrics as a requirement. MIL-STD-SDD is in draft form and final approval is expected in September 1993.

It is imperative that NAVAIR participate in the standards revision process to insure that NAVAIR's requirements are included. Efforts are on-going to improve the standard engineering practices and make allowances for moving to electronic data and controls including data base driven requirements and executable models for design and analysis. This is not just placing documents in electronic form, but rather allowing contractual requirements to reside in a configured data base which is accessible by both the government and the contractor.

### **9-3.3 Commercial-Off-The-Shelf(COTS)/Non-Development Items (NDI)**

COTS/NDI is the preferred method of satisfying computer resources requirements. Reuse of previously developed and available computer resources or commercial computer hardware and software can reduce program costs, shorten acquisition time, and reduce program risk. The following order of preference should be used when developing or modifying avionics system software:

- Use COTS software without modifications, although NAVAIR will not maintain COTS software.
- Reuse existing Ada code maintained by NAVAIR, DoD, or other government agencies.
- Reuse and upgrade existing software maintained by NAVAIR, DoD, or other government agencies, if it is existing code written in a high order language (HOL), and the total number of lines to be modified and/or added is less than 1/3 of total compileable source code for the system.
- Develop new code using Ada.
- Request a waiver and develop non-Ada code.

If COTS/NDI software is being procured, rather than being developed, the programming language used by the developer of the COTS/NDI product is not of vital concern, unless it is expected that the COTS/NDI product will be included as part of another application. If the COTS/NDI software will be used as part of a larger system, the designer must ensure that the COTS/NDI will be supported throughout the system's life cycle, and that upgrades to the COTS/NDI will not affect other parts of the system. If the system is critical, data rights may need to be acquired. If data rights cannot be obtained, this may be justification for not using COTS/NDI computer resources.

### 9-3.4 Languages

Software languages are an efficient way to tell a computer how to organize its hardware logic. The development of software has progressed from writing ones and zeros and putting them into the machine with front panel switches to very sophisticated Higher Order Languages (HOLs).

At present the best language for large complex software systems is Ada. As a follow on, Ada 9X will provide additional efficiencies and the object oriented paradigm without increased dangers. Ada is the most portable HOL available due to the existence of the Ada Compiler Validation Capability (ACVC). Certification through the ACVC results in a higher degree of portability by validating conformance to specific features of the standard and identifying those elements that are nonstandard extensions. The Ada Joint Program Office (AJPO) maintains a list of validated Ada compilers; as of January 1993, there were over 500 validated Ada compilers. This list is available from the Ada Information Clearinghouse (AdaIC), either from the following address or by downloading it from the AdaIC Bulletin Board at (703)614-0215 or (301)459-3865. The list is located in the file "VAL\_COMP.HLP" in the Ada Information Files directory.

Ada style guides are desirable for their contribution to the quality and consistency of Ada code. The AJPO has endorsed a Software Productivity Consortium (SPC) publication as a suggested Ada Style Guide for DoD programs: SPC-91061-N, Ada Quality and Style: Guidelines for Professional Programmers, Version 2.0, 1991. This style guide should be specified when procuring software.

Other HOLs include ATLAS, BASIC, C, C++, CMS-2, COBOL, FORTRAN, and JOVIAL. Some languages, such as LISP and PROLOG, facilitate the use of computers to mimic intelligence.

DoD has been a user of software languages throughout, and has contributed heavily to their creation in many cases. DoD sponsored and paid for the development of the Ada language (called Ada 83 because its American National Standards Institute (ANSI) standard definition was promulgated in 1983) in an attempt to make large programs testable, reliable, and portable. DoD was very successful and present estimates are that for developments of systems requiring over 10,000 lines of code, it is more economical to use Ada because of the savings in debugging time alone. Additional savings result from the increased reliability of the software and from the decrease in life cycle maintenance costs.

In order to promote an engineering environment, it is necessary to limit and control the languages allowed in a system. The latest trend in computer languages has been towards "object oriented" languages. These are languages in which one creates "objects", which are pieces of code having a definite functionality. An object contains the definition of the data structures and what operations that can be performed on the data. This allows objects to be placed in a library and reused by other programs, instead of having to continually reinvent them. The usefulness is somewhat limited by the fact that objects may "inherit" from other objects, so that an object's true definition may be heavily nested. That means that to use it, one has to have the entire library. To understand exactly how an object performs its functions requires a study of the entire string of antecedent objects. For printer and display routines, this is satisfactory, but for real-time avionics systems there is some question as to its usefulness. In fact, the current version of Ada fully supports two of the four characteristics of an Object Oriented Programming (OOP) language. That is, abstraction and encapsulation are supported,

inheritance and polymorphism are not. Inheritance can be simulated in Ada 83 with little difficulty, but simulating polymorphism is painful to the point of being impossible. Ada 9X will include inheritance and polymorphism. Furthermore, the proposed forms of these new features have also been stable for some time. Although not strictly an OOP concern, Ada 9X also supports the optional definition of constructor and destructor routines for any user defined type. Thus, Ada 9X will be an object oriented language. Its forms of inheritance and polymorphism are different from the forms in any other OOP language, but then no two of the other OOP languages use the same forms either. One of the key features of Ada's version of these features is that they are being added to the language in a type-safe manner. Thus, it will still be possible to get the extensive compile-time checks supplemented by additional run-time checks that are one of Ada's prime characteristics.

Finally, it should be noted that the entire question of interfaces between software and hardware is an area of active research. As an example, a class of machines called "RISC" (for Reduced Instruction Set Computers) has arisen in the past five years. They are designed to exploit repetition normally found in large, complex programs. Non-RISC computers have hardware resources which are only infrequently used (but promote efficiency when used) requiring complicated instructions to control those resources. An example might be a floating point unit on a machine doing only integer arithmetic. The way around including this seldom used equipment was to embed special operations like a floating point multiply in the compiler software, and to carry it out by performing a series of simple single cycle operations. By doing this, the more often used instructions could be carried out much faster because the hardware cycle was made much simpler and shorter. This in fact works well in most applications. But, the so called "RISC" machines have started to add resources as the hardware technology has grown. Thus, it will be necessary for NAVAIR to maintain an active monitoring of this whole area of computer research in order to keep up with the best available commercial equipment and technologies.

Most commercial operating systems, including the UNIX operating system, are written in C. Additionally, C is the most commonly used language for COTS software, communication programs, and bindings to existing systems. In systems where COTS software is to be used extensively, the amount of non-COTS code that has to be developed and the interfaces to the COTS software need to be considered when evaluating the long-term cost/benefits of using C versus Ada as a development language. In most cases, developing Ada links to existing C bindings has proven to be an effective development method. Furthermore, in applications where concurrent processing is required, Ada's inherent implementation of concurrent methods is preferable to C, since concurrent processing in C is handled by reference to UNIX operating system calls. Ada's concurrency methods are independent of the operating system.

C++ contains object-oriented extensions to C. The same considerations described previously for C also apply for C++. In addition, C++ is not as widely available as Ada. FORTRAN, JOVIAL, and LISP have been used traditionally for real-time and scientific processing. The availability of Ada compilers and cross-compilers make Ada a cost-effective alternative. Although FORTRAN-90 contains added capability over FORTRAN-77, it does not contain any capabilities that make it preferable to Ada for large systems.

COBOL is used extensively in data-intensive, non-embedded systems such as business and legacy software. Ada 83 has been shown to be effective in these applications as well as less costly to maintain. Furthermore, Ada 9X will include features, such as fixed point arithmetic, that have been identified as the cause of

portability issues in these applications. The re-engineering of COBOL programs to Ada has been proven to be more cost-effective than maintaining the existing systems in the long term.

### 9-3.5 Software Engineering Environments/Software Tools

The automation of the software development process using Software Engineering Environments (SEEs) involves having an established, repeatable, and consistent software process. Procedures and methods that establish the engineering approach along with all necessary engineering events, controls, tools, and management functions must be embraced by the SEE. SEEs must also exist within the broader context of the systems engineering process.

Depending on the particular avionics system being developed, a broad range of software tools, ranging from database management to software design tools to project management tools, may be required to support the effort. Software tools can be grouped into general types such as management tools, development tools, and laboratory tools.

Management tools include tools to perform planning, scheduling, requirements traceability, configuration control, and documentation. Development tools include structured analyzers, editors, compilers, code generators, debuggers, and emulators. Laboratory tools include automated testers, data manipulators, simulators, real-time non-intrusive testers.

Many of the tools required for developing avionics system software exist today; however, there is little integration between and among these tools. Future SEEs must be seamlessly integrated, easy to use, support a distributed environment, support graphical interfaces, and have a cost effective licensing policy which recognizes the true patterns of tool usage.

### 9-3.6 Operating Systems

Operating systems are very general pieces of software which provide certain interface and commonly used services to other applications software, thereby hiding the details of the computer from the applications software. Examples include VMS (DEC), UNIX (Bell Labs), MS DOS (Microsoft), OS/2 (IBM-PC), etc. The types of services provided include Input/Output (tape, disk, video terminal), memory allocation for applications, loading and running applications software, switching of jobs under multi-tasking systems, accounting, file storage and retrieval services, print formatting, and general interfaces. Although very useful for general purpose computers, they take up too much memory and execution time, and sometime cause non-determinism for real-time systems. For that reason, real-time systems tend to be "complete" and perform all their own services by displacing the operating system once they are loaded.

Since the interface between operating systems and applications programs differs for each operating system, there are definite problems with portability of applications between platforms running different operating systems. To alleviate that problem, a new standard (POSIX) has been proposed and is being developed. Although it will not provide a standard operating system, it will provide a set of standard services and a standard method for applications programs to evoke those services, thus allowing the applications to be ported to any computer utilizing a POSIX-compliant operating system. POSIX will probably have to undergo several revisions before it is mature enough to be a

really firm standard, but several software and computer houses are already advertising "POSIX Compliant".

Because of the increased complexity of the avionics systems and associated computer resources, it is becoming more and more difficult to start from scratch when developing modern avionics systems. For real-time systems, developers have started to utilize so called "real-time kernels": operating systems which have all the non-essentials stripped out and are carefully programmed to make sure that all provided operations are deterministic in nature (especially regarding their completion time).

Although operating systems cause real-time problems, the move to distributed computational systems will require the use of some sort of operating system just to maintain the message traffic between applications on multiple computers. Although this will be less efficient than a tightly coupled system, the added time performance provided by the increased hardware operating in parallel can more than compensate for the decreased efficiency. The most touted parallel operating system at present is "MACH", which is a reduced UNIX system that runs cooperatively on multiple heterogeneous platforms.

### 9-3.7 Testing

Testing is an important part of software development, is seldom adequately done, and never completely done. That is because all computers are finite state machines (they have a finite number of gates each having only two states, so the entire machine can always be represented by a binary number whose length equals the number of gates in the machine), but the progression from one state to another is not well defined. Assuming the computer hardware is functioning correctly, the software is responsible for defining the transition from one state to another. Thus, the flexibility provided by the software is the true enemy of adequate testing.

For a home computer to go into an unknown state where something totally unexpected happens may be annoying; but, for an aircraft flight control computer to malfunction during a carrier landing creates hazards for both the pilot and his shipmates. Also, some computers carry information which needs to be kept secure, and operational systems need to have their applications software secure from either unwanted or unexpected changes. Unexpected sequences in a computer can open the entire system to unwanted intrusion, and subsequent damage or destruction of the operational software. For these reasons it is very important that a military computer in an avionics system does only predictable things at all times.

Since it is clearly impossible to exhaustively test all states of a piece of computer hardware, the only solution is to test all the pieces and connections between those pieces. Since software and computer hardware have the same basis, the same rational holds true for software. The only way to test large software programs is to create small well defined functional blocks and test them completely. If the structured engineering technique is used, these blocks will be created in a top down manner, so that the requirements for each block and the necessary connections and tests are well known.

Thus, the testing seems easy, but tedious. In fact, the necessary tools to partition software into rationally functional blocks does not yet exist for the most part. Those which do exist will often give very different answers for slight differences in initial inputs. But this partitioning problem is exactly synonymous with the systems engineering problem of breaking requirements at each level into simpler sets of

requirements, and then testing them for consistency and completeness before proceeding to the next lower level. On finally reaching the implementation level, the entire system can be validated by only running selected individual strings at the lowest level while running most of the system at its highest level. This is a variation of "correct by construction." In order to perform software testing adequately though, it is necessary that testability be built into the process from the start, and be given at least equal status to every other part of the system design and build.

### 9-3.8 Software Reusability

Software reuse includes the reuse of software designs, systems architectures, software components, requirements and test documentation, tools, test data, interfaces, concepts, and source and object code. Realistic reuse leverages the best solutions, designs, and architectures for use in avionics systems. Contractors are now realizing that reuse of these elements are just as valuable if not more so than the actual source code.

The reuse of these different products during software development rather than producing them from scratch can reduce the software cost considerably. In addition, software which has been frequently reused has undergone much better testing than the average project can provide. This provides for a much better life cycle maintenance, aided by the larger user base.

Using previously developed avionics software makes economic sense just as use of previously developed avionics hardware in other model/type of aircraft makes economic sense. In the past, there was little standardization in "deeply" embedded software across aircraft platforms, and due to the differences in languages and compilers used on different platforms, very little of the software was reusable. With new avionics system software being written in Ada and the drive toward open systems, opportunities for economical software reuse will exist.

In the past, software reuse has been limited to modernization within a single aircraft model/type. The requirements and design from an earlier aircraft model/type were used as a baseline, and changes and additions made, rather than starting a whole new development for each upgrade. Reuse of software in new aircraft programs from other existing aircraft models/types was not feasible.

There were a variety of reasons why software reuse was not practiced on a large scale in NAVAIR as well as other DoD systems. The two greatest were cost and ownership and these two problems still remain. In order to take advantage of what has already been created, a detailed analysis (called domain analysis) is required to review what other similar systems have built, and how they match the requirements of the new system. Most aircraft programs have limited time and budgets, and cannot afford this extra step. And, the previous software did not receive the extra amount of effort and dollars to assure that it would be designed and documented so that it could be reused by another aircraft weapon system.

Perhaps the biggest single impediment to software reuse are the legal issues surrounding ownership. If developers find proprietary software that fills a requirement, they must negotiate with the builder. If the buyer then adds functionality to the software, he can resell it in some cases (derivative rights) with royalties back to the originator. The issue of warranty if a system problem occurs further confuses the situation. Thus most developers have found it to be easier to build from scratch than to deal with these issues.

Because of this and for reasons of life cycle support, DoD has in many systems chosen to own all rights to the system built by a given contractor. This affords DoD the ability to use another contractor or Navy field activity to support the system in the maintenance phase. But the initial contractor is not allowed exclusive rights to resell any part of the system, so it is not built to be reused. Nor is there much effort on the part of contractors to build new systems or subsystems on their own money, since they face the prospect of having the government force them to relinquish ownership. But even if the DoD relaxes its ownership requirements, there are not many military systems for a prospective industry player to use to amortize costs. This argues strongly for close ties to commercial endeavors which need similar components.

Given that software reuse is a viable means for reducing risk and cost, there is no support environment to facilitate software builders in developing software that is reusable later, or finding and integrating existing software for their current application. The DoD Software Reuse Initiative is coordinating a variety of efforts to build this infrastructure. Among these are the creation of libraries and support tools (classification, browsers). Computer storage of software is relatively easy, but retrieving software elements has proved to be much harder. Classification schemes are in development to tag software. Just as library browsers are available to aid in quickly scanning libraries for books and articles, so are software browsers being developed to aid developers in retrieving applicable software. As with any retrievable items, catalogs of software elements are required, and must be computer based because of the large number of elements. Since most companies have their own proprietary libraries, and since DoD is sponsoring several library efforts, there is a need for interpretability among libraries. Indeed, DoD is also sponsoring a joint effort to provide for interpretability of libraries.

The lower cost and risk of a previously developed software (either NDI or COTS) application puts great pressure on the Navy program managers to consider its use. It should be noted however, this lower cost comes with important problems to be solved. First and foremost, the limited rights that come with COTS/NDI may preclude visibility into the software to allow safety analysis required for airborne systems, and precludes detailed analysis to ensure that necessary security was maintained to limit penetration from outside (as from planted trap doors or viruses). These earlier design slips could lead to damage to, or even grounding of an entire new model/type of aircraft.

The second major issue is one of life cycle support. In the past, the Navy has used field activities to support and maintain avionics system software once the system is in production. Typically these activities developed upgrades based on fleet requirements and deliver them in 18-24 months. With limited rights, the Navy may be seen to place itself in a sole source position for upgrades to the COTS/NDI software. Depending on the level of reuse, the Navy's ability to respond in a short time frame to fleet requirements could be severely limited.

Code reuse has been invoked as a means to reduce costs and schedule for future programs. Along with other problems, a lack of consistency in the use of "reuse" terminology coupled with confusion throughout government and industry in understanding the principles inherent in making and incorporating reusable code has produced a credibility gulf that has inhibited the maturing of reuse technologies. Establishing recognized nomenclature and conventions for reuse are both necessary for government and industry to collectively define the reuse problems and identify strategies to begin resolving them.

### 9-3.9 Rapid Prototyping

Rapid prototyping serves the same purpose for software as for hardware. It is used to determine if questionable or key assumptions are valid. In software, as in hardware, it is also used to determine if the requirements which have been assumed are valid when taken together as a group. There are software tools such as STATEMATE available now which allow one to block in the requirements and assumptions, and determine if the system operates in a correct and timely fashion under those assumptions. Even more important, such tools allow one to recover from assumptions which later prove to be incorrect by providing control of all the variables in a software system and allowing for rapid creation of alternate paths or divisions of code functionality and subsequent hardware mappings.

## 9-4 Computer Resources Management

The development of computer resources, if not properly managed, will result in cost overruns and schedule slippages. For the computer software portion of the development to be successful, a structured software engineering discipline must be applied, from requirements definition through final test and acceptance, similarly to those of other engineering disciplines in the engineering community. Both government and contractor personnel supporting software intensive programs must have sufficient technical knowledge of the software engineering development process.

Government and industry alike must place greater emphasis on the preparation for and the building of avionics weapon systems which require large scale software developments. This includes training of personnel, establishing software and design repositories, engineering and software environments, integrated design processes, definitive development methods and procedures, etc.

In the past, program offices have underestimated the resources and costs associated with supporting the design, development and support of software systems. Significant focus has at times been placed on the design of software itself, but not on the infrastructure and support elements (software tools) required to ensure a successful software development.

### 9-4.1 Process

A process for developing software must be developed and followed in order to ensure a successful software development. Many actions are in process or being taken to improve program software development. One of which is the efforts at the DoD Software Engineering Institute (SEI). SEI has developed a Capability Maturity Model (CMM) to address the discipline required. The CMM addresses the disciplines/processes which should be in place for an organization to produce reliable software in a timely and cost effective manner. The CMM as structured has five levels of maturity and describes processes which must be in place to reach the next higher maturity level. The five levels are Initial (Level 1), Repeatable (Level 2), Defined (Level 3), Managed (Level 4), and Optimizing (Level 5). These processes must be integrated into the overall systems engineering process and not stand-alone software development processes. The key benefits that an organization gain by raising its process maturity level are more accurate predictions for software product size, cost, quality, and schedule; reduced variability; and improved software process results. In addition, an organization benefits from better technical and management visibility into their software development process. More

accurate predictions are most useful for controlling the process and producing products on time and within costs projections.

The CMM provides an invaluable method for assessing the ability of a software developer to satisfactorily accomplish a software development effort and to identify areas where an organization's software process needs improvements. The model rests on the premise that software process maturity is a credible indicator of capability. The concept implies that the productivity and quality resulting from an organization's software process can be improved over time and presumes that improvement comes through consistent gains in the discipline achieved by applying the CMM. The implication is that as an organization gains in software process maturity it institutionalizes its software process both by means of policies, standards, and organizational structure and by building a corporate culture that supports the methods, practices, and procedures of the business. In this way, the software process (with its methods, practices, and procedures) endures after those who originally defined them have gone.

Finally, each higher level of process maturity is taken as indicating both greater control of an organization's software process and greater consistency with which the process is applied in projects throughout the organization. Hence, the results of applying the process are expected to be more predictable at successively higher levels. The CMM serves three important needs of a software development organization. It provides:

- 1) An underlying structure for reliable and consistent assessments.
- 2) A framework designed to help software organizations characterize the state of their current software practices in consistent terms, set goals for improving their software process, and set priorities for instituting their process changes.
- 3) A guide to organizations planning their evolution toward a culture of engineering excellence.

A well defined and planned software measurement program is required to progress through the levels of the CMM. The basic measurement set at Level 2 is size (lines of code), effort (labor hours), schedule (calendar time), and quality (number of defects). Level 3 functions add to the basic set by defining and institutionalizing the organization's software development process and on estimating for a project's defined software process. Level 4 then focuses on further identifying and quantifying the organization's software development processes, selecting process and product data to be collected, analyses to be performed, process and product metrics to be used in managing a project, and defining quantitative goals for product and process quality. Level 5 focuses on the optimizing process by incorporating the lessons learned from continuing process measurements and development experience.

## 9-4.2 Metrics

A developer must establish project goals and software metrics for project control and process improvement. Software measurables are directly observable quantities that you can count, such as lines of code, or that you can otherwise measure, such as labor hours and labor months. A software measurement is a number assigned to an observable, i.e., a quantitative assessment, of a software process or product. A software metric is a number assigned to a quantifiable concept that relates to a software product or to the process that created it. A metric is not always observable. A metric may be a single measurement (lines of code) or it may be a function of one or more measurables (lines of code per labor months). The main categories of metrics are:

- 1) product size - counts of: initial, undefined, added, and deleted requirements; design statements, document pages, process "bubbles", data entities, and boxes or arrows in hierarchical input-output charts; source statements, comments, function points, object code instructions, and words of memory; tests, test procedure steps, and pages of test documentation; entities such as CSCIs, CSCs, CSUs, algorithms, function and feature points, inputs/outputs, logical files, internal interfacing hardware components, and external system interfaces; documents, types of documents, and pages of documentation; and status of ECPs, STRs, authorized and staffed personnel positions, and percent of budget spent.
- 2) product costs - initial estimate and final actuals, budgets and amount expended
- 3) schedule - elapsed times in weeks, months, or years
- 4) quality - number of defects, defects per unit, cost per defect, average number of days to fix defects, effort to verify the product, effort to reuse the product as a function of effort to develop the product, number of unique inputs and outputs and the number of assignment statements, etc.
- 5) product application environment - system resources - real-time limits, complexity, throughput, memory, input/output
- 6) development environment characterization - modern programming practices/methods, risk management, supporting workstations, software tools, requirements stability, process stability, simultaneous hardware/software development, standards enforcement, type of contract
- 7) development constraints - severity of cost, schedule, and staffing constraints, and development personnel characterization (experience, application familiarity, and language familiarity)

Those who will use information and what information is required must be a foremost consideration in arriving at the specific metric(s) to be collected. Many different users of information exist. A given user will only be interested in collecting and analyzing those metrics that apply to his areas of interest. Users of information may fall into one or more of the following groups: software engineering process, software quality assurance, systems engineering, program/project management, software management, software engineering, software development, software/system test, and accounting/finance.

Arriving at the actual metrics to be collected is aided if the following process is used. State the goal(s) of the development. Use the goals to determine who are the users of the information are and what they need to know. The users of the information will be both contractor and government personnel. Having determined the information which will be required, decide what questions users are going to ask to determine that the goal(s) have been satisfied. The specific metric that must be collected and the things that are to be measured can now be determined. The method(s) for collecting the metric(s) can now be established.

#### **9-4.3 Software Experience Database**

Every organization, including contractors and government, should establish and maintain a software experience database. This database should contain the measurements, metrics, and other information that can be used to support project control and software process improvement. This database can also be used to support software CIs which have entered the PDSS phase.

The establishment and maintenance of this database should be mandated by and supported at the highest organizational level. The information contained in the database can be used to form the basis for improving software standards, improving the planning and proposal process, estimating and managing costs, developing and refining estimation models, determining size and unit cost of future software, evaluating product quality, managing risk, and improve the development process.

The software experience database is an important component of the quantitative software management process. It is a powerful tool for improving organizational performance. Information contained in the database should be readily available to all organizational elements participating in the avionics system/software development process.

#### **9-5 Software Science and Technology Thrusts**

It is imperative that NAVAIR participate in Science and Technology Thrusts which are being pursued to insure that NAVAIR stays abreast of and can influence current technology developments. Areas in which NAVAIR should be concerned include software and systems engineering, human-computer interaction, artificial intelligence, parallel and heterogeneous distributed systems, real-time fault-tolerant software, high-assurance software, reusing software, software security, advancing programming techniques, re-engineering, improving software maintenance, and improving software engineering environments. Work in these areas is underway both in industry and the government through ongoing efforts like "Software Technology for Adaptable, Reliable Systems (STARS). NAVAIR should participate in the activities underway at the Software Engineering Institute (SEI), the Software Productivity Consortium (SPC), and the Software Technology Support Center (STSC).

## 10-0 REFERENCES

Checkland, Peter (1981), *Systems Thinking, Systems Practice*, Chichester, England: John Wiley.

Rechtin, Eberhard (1991), *Designing and Building Complex Systems*, Englewood Cliffs, NJ: Prentice-Hall.

DEPARTMENT OF THE NAVY  
NAVAL AIR SYSTEMS COMMAND

ADVANCED AVIONICS ARCHITECTURE  
AND TECHNOLOGY REVIEW  
FINAL REPORT

**\*\* VOLUME 2: AVIONICS SYSTEMS ENGINEERING \*\***

6 August 1993

NAVAL AIR SYSTEM COMMAND  
SPACE AND NAVAL WARFARE SYSTEMS COMMAND  
NAVAL RESEARCH LABORATORY  
NAVAL AIR WARFARE CENTER, AIRCRAFT DIVISION-  
INDIANAPOLIS & WARMINSTER  
NAVAL AIR WARFARE CENTER, WEAPONS DIVISION-CHINA LAKE  
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY  
NATIONAL INSTITUTE FOR STANDARDS AND TECHNOLOGY

**This page intentionally left blank.**

# Table Of Contents

1-0	BACKGROUND .....	2
	Importance of Systems Engineering (SE).....	2
	Role of MIL-STD-499B.....	2
	Focus on Open Systems.....	3
1-1	The Role and Scope of Systems Engineering.....	3
	1-1.1 Systems Engineering (SE) Definition.....	4
1-2	Avionics Application of Systems Engineering.....	4
2-0	SYSTEMS ENGINEERING POLICY .....	5
2-1	Specific Systems Engineering Responsibilities.....	5
2-2	Specified Tools.....	6
3-0	THE SYSTEMS ENGINEERING FRAMEWORK .....	7
3-1	Systems Engineering Process .....	9
3-2	The Principal Systems Engineering Activities of the Systems Engineering Process.....	10
	3-2.1 The Primary Tasks of the Systems Engineering Process.....	10
	3-2.2 Inputs To The Process : Requirements .....	11
	3-2.3 Requirements Analysis .....	11
	3-2.4 Functional Analysis.....	11
	3-2.5 Synthesis.....	11
	3-2.6 System Analysis and Control (Balance).....	12
	3-2.7 Outputs of the Process : Decision Data Bases.....	13
4-0	SYSTEMS ENGINEERING THROUGHOUT THE LIFE CYCLE .....	14
5-0	PLANNING FOR SYSTEMS ENGINEERING.....	17
5-1	Management Plans.....	17
	5-1.1 Systems Engineering Management Plan.....	17
	5-1.2 Government SEMP .....	17
	5-1.3 Contractor SEMP .....	18
5-2	The Use of System Effectiveness Metrics and Overall System Cost Criteria in Planning .....	18
6-0	SYSTEMS ENGINEERING OUTPUTS .....	20
6-1	SE Baselines Used in Acquisition Management.....	20
6-2	Specifications.....	20
6-3	Specifications used in Systems Engineering.....	21
7-0	REQUIREMENTS ANALYSIS .....	25
7-1	The Requirements Analysis Process.....	25
7-2	Inputs to and outputs of the Requirements Analysis portion of the Systems Engineering process. ....	25
7-3	The Role of Trade Studies in the Development of Functional and Performance Requirements. ....	26

# Table of Contents (continued)

7-4	The Importance of Requirements Analysis to Systems Engineering.....	26
8-0	FUNCTIONAL ANALYSIS, SYNTHESIS AND WORK BREAKDOWN STRUCTURES.....	28
8-1	Evolution of the Functional Architecture to a Physical Architecture.....	28
8-2	The Evolution and Uses of the Engineering WBS.....	28
8-3	Roles of the WBS in the Program Office.....	29
9-0	TECHNICAL RISK MANAGEMENT.....	30
9-1	Defining Risk.....	30
9-2	The Technical Risk Management Process.....	31
10-0	TRADE STUDIES AND EFFECTIVENESS ANALYSIS.....	32
10-1	Trade-Off Analysis.....	32
10-2	Elements of the Trade-Off Process.....	32
10-3	Effectiveness Analysis.....	33
11-0	TECHNICAL PERFORMANCE MEASUREMENT.....	34
11-1	General.....	34
11-2	Technical Performance Measurement (TPM) Profile.....	34
12-0	TECHNICAL REVIEWS AND AUDITS.....	37
12-1	General.....	37
12-2	Reviews and Audits Under MIL-STD-499B.....	37
13-0	CONFIGURATION MANAGEMENT.....	39
13-1	The Central Role of Configuration Management.....	39
13-2	The Use of Configuration Baselines.....	39
13-3	The Roles of the Government and the Contractor in Configuration Management.....	40
14-0	SOFTWARE SYSTEMS ENGINEERING.....	41
15-0	SUPPORTABILITY ENGINEERING UNDER THE SE PROCESS.....	41
15-1	General.....	41
15-2	Support Environment.....	41
15-3	Integrate Support Concepts into Systems Engineering Guidelines.....	41
16-0	EFFECTIVE USE OF COMPUTER BASED SYSTEMS ENGINEERING TOOLS.....	43
17-0	CONCURRENT ENGINEERING.....	43
18-0	SYSTEMS ENGINEERING FOR MODULAR AVIONICS.....	44
18-1	General.....	44
18-2	JLAWG Modular Avionics.....	44

# Table of Contents (continued)

18-3	Integrated Modular Avionics.....	46
19-0	TQM AND THE SYSTEMS ENGINEERING PROCESS .....	47
20-0	PRODUCT IMPROVEMENT .....	48
20-1	General.....	48
20-2	The Role of Systems Engineering in Product Improvement Programs.....	48
20-3	Avionics Experience in Product Improvement Programs. ....	49
20-2.1	MIL-STD-1553 (Standard Avionics Data Bus).....	49
20-2.2	Standard High Speed Data Bus.....	49
20-2.3	Modular Avionics.....	50
20-2.4	Global Positioning System Retrofits.....	50
20-3	Recommendations for Product Improvement Systems Engineering.....	50
21-0	SYSTEMS ENGINEERING GUIDANCE .....	52
21-1	General.....	52
21-2	Custom tailoring of MIL-STD-499B for Avionics/Aircraft Development.....	52
21-3	Customizing SE for Advanced Integrated Avionics.....	53
21-3.1	Accommodate Open Systems Trends.....	53
21-3.2	Linking of COTS and Open Systems Architecture.....	53
21-3.3	Modular Avionics for Growth and Ability to Upgrade .....	54
21-3.4	Process Integration of Hardware and Software Development.....	54
21-3.5	Employ Multi-Disciplinary Product Teams .....	54
22-0	SYSTEMS ENGINEERING STANDARDIZATION .....	56
22-1	General.....	56
22-2	Defense Systems Management College.....	56
22-3	National Council of Systems Engineering .....	56
22-4	Standards for Systems Engineering Education .....	57
22-5	The Role of the Society of Automotive Engineers (SAE) Avionics Systems Division .....	58
23-0	IMPLEMENTATION STRATEGIES AND PLANS .....	59
	General Approach.....	59
23-1	Apply Open System Standards through the SE Process.....	59
23-2	Promote a Rational Use of COTS/NDI Under SE Guidelines.....	60
24-0	SYSTEMS ENGINEERING SUMMARY.....	61

# Table of Contents (continued)

## APPENDIX A:

SYSTEMS ENGINEERING GLOSSARY .....	A1
1.0 Definition of Terms Commonly Encountered in Systems Engineering .....	A1
2.0 Acronyms and Abbreviations Encountered in the Engineering of Military Avionics.....	A16

**Advanced Avionics Architecture  
and Technology Review**

**Final Report**

**>> Volume 2 - Avionics Systems Engineering <<**

Throughout this Review, industry respondents emphasized the important role played by Systems Engineering (SE) as a major contributor to effective avionics systems development. Although most of the major aerospace firms that contributed to the review place a great deal of importance on "the SE process", there is a great deal of variability in how SE is defined and actually applied by these firms. This volume provides a summary overview of the SE Process (SEP) as taught at the Defense Systems Management College and as currently applied by several major programs in the Army, Air Force and Navy. An introductory discussion of today's current role and tomorrow's future potential of utilizing SE to develop affordable, high performance avionics systems is also provided to stimulate interest.

This volume outlines the basic elements of SE with focused emphasis on its application to the specific development of avionics systems. It is intended to highlight the highly important, expanding role of SE to more efficiently and effectively develop modern avionics systems.

## 1-0 BACKGROUND

The development of large and complex systems has given rise to an increased awareness of, and need for, Systems Engineering (SE). This has been especially important in the development of DoD weapon systems because of the need to inject new technology into both new and existing systems, in a manner that controls the inherent risks. The difficulties experienced in evolving new systems has led to the development of specific methods and techniques within SE in an attempt to provide better insight and control into the development and management process.

### Importance of Systems Engineering (SE)

A major finding of this review is the critical requirement for a resurgence in emphasis on SE. In one sense SE is considered to be nothing more than "good, common sense" engineering practice. In a more formal sense, SE describes the complete process of engineering as applied to system development. SE is the primary technical methodology by which customer needs are translated into a balanced set of product and process descriptions. A majority of the corporations that participated in the review placed considerable emphasis on the need for a rigorous SE approach as a basis for effective systems development. SE is undergoing a revolution today, primarily because it is recognized that many earlier approaches to the engineering of systems development are inadequate for the complex, software intensive systems of today. An approach is needed that considers, in addition to product operability, all down stream processes such as producibility, testability, deployability, supportability and disposability in its early development efforts. In addition, the revolution is enhanced by the availability of a large number of computer based tools. Tools are available and are continuing to evolve that support virtually all activities within Systems Engineering.

One stated purpose for this review is to predict the avionics architectures that the Navy will deploy 10 to 20 years hence. We are unable to make these predictions explicitly because of uncertainty over the exact impact of new and emerging technologies. *No one can make these predictions explicitly!* However, during the course of this review, it has become clear that the process by which future architecture alternatives are conceived, synthesized, analyzed and selected is just as important a factor as the specific technologies employed. Designating this approach "Systems Engineering" we are safe in saying that if a well defined avionics SE approach is in place, and if the available avionics architectural alternatives are objectively considered through this process, the Navy will obtain avionics architecture solutions that optimally balance emerging technologies and the use of Commercial Off The Shelf (COTS) products against the practical constraints imposed by fleet supportability and affordability requirements.

### Role of MIL-STD-4993

The defining document for the application of SE to the development of military systems is MIL-STD-499B entitled "Systems Engineering". A renewed emphasis on SE is reflected in the title of this Standard, changed from "Engineering Management" of the predecessor MIL-STD-499A. This standard provides a comprehensive guide to the SE Process to be applied to the development of defense systems. The current revision (draft dated 6, May 92,) is waiting final release but has been fully reviewed and is being referenced extensively. Although not officially released, this document has achieved de facto recognition as the military SE guide. This revision is considerably updated and is consistent with recent SE thought and current military practice.

MIL-STD-499B must be considered a general standard or "framework" for the application of SE to the acquisition of military systems.

Application of SE to military avionics requires background information and guidance specifically tailored to the application. Systems Engineering, as delineated in MIL-STD-499B is designed to function in accordance within event-based milestones imposed by the acquisition life cycle. The process requires performance of certain specific tasks for successive development levels (concept, system, subsystem, ...) and the decision events related to progressing from one level to the next. Technically, application of the SE process, as defined in MIL-STD-499B, must be customized to deal specifically with avionics system needs and with current technical trends in military avionics.

### **Focus on Open Systems**

Although it is recognized that an avionics system is but a subsystem of the total aircraft system; it is a costly subsystem and therefore it is essential to take advantage of the trend toward Open System Architectures (OSA). Avionics application of SE appropriate for the present day must accommodate this trend. The architectures chosen for avionics systems must be well defined and suitable for the growth and evolution required by increasingly complex system needs for advanced avionics. The aircraft systems envisioned should place considerable emphasis on the use of modular packaged avionics. The use of a modular OSA provides a physical framework for the development of advanced avionics systems while providing a convenient basis for technology insertion and evolutionary growth (at the module level). A modular Avionics framework utilizing OSA provides a means of mitigating technology obsolescence while providing the military an opportunity to leverage commercial technology trends and enhance "economy of scale" potentials through "dual use" technology application.

*An up to date tailoring of MIL-STD-499B for Avionics should establish the basis for the application of a family of "architecture defining" Open Systems Architecture standards.*

## **1-1 The Role and Scope of Systems Engineering**

The essential role of SE in the systems acquisition process is to ensure that the right things get done during the development of new defense systems, or during any upgrade of already fielded systems. By employing SE methods a total system design solution is developed that incorporates all down-stream life-cycle needs (e.g., manufacturing, verification, deployment, operations, support, training, and disposal) at an affordable balance of performance, risk, cost and schedule. SE is the primary technical methodology by which customer needs are translated into a balanced set of product and process descriptions. These descriptions should enable a quality system to be produced and deployed that satisfies the operational need and is affordable, operationally suitable and operationally effective.

### 1-1.1 Systems Engineering (SE) Definition

There are many general definitions offered in the literature to cover the scope of Systems Engineering. The definition in MIL-STD-499B is comprehensive, linking together the many elements of Systems Engineering.

*Systems Engineering: "An interdisciplinary approach to evolve and verify an integrated and life-cycle balanced set of system product and process solutions that satisfy customer needs. Systems Engineering: (a) encompasses the scientific and engineering efforts related to the development, manufacturing, verification, deployment, operations, support, and disposal of system products and processes., (b) develops needed user training equipment, procedures, and data (c) establishes and maintains configuration management of the system, (d) develops work breakdown structures and statements of work, and (e) provides information for management decision making."*

SE is required during each level of development and during each acquisition and support phase of the life cycle. The principles of SE apply to all system developments whether large or small, major or non-major, new development, modifications or upgrades, and for single or multiple procurements. Properly performed, SE requires full consideration of all the technical disciplines involved in the system development. This mandates participation by interdisciplinary teams of engineers, concerned with design, development, production, testing, logistics and training, and by appropriate representatives from contracts, budgeting, finance, legal, marketing and the using community.

Although the need for SE is universal for any engineering development, its application is crucial for the effective development of large and complex systems characteristic of modern integrated avionics.

### 1-2 Avionics Application of Systems Engineering

Beyond the above definition, Systems Engineering applied to advanced integrated avionics systems development must have certain properties.

It must be kept in mind that the avionics system is a subsystem of a larger aircraft system. The needed functions of the avionics subsystem should be defined by a "Top Down" process. Once the avionics functions are defined, then the avionics system planning must begin at the top and flow down to the various levels. This requires that there be a top level concept development which is the architectural model ("vision") selected for the system to be developed. This systems vision may change due to constraints found during the development process, however without the initial top level model/vision, any chance of producing a balanced avionics system design is virtually non existent. A "Top Down" approach is generally advocated for Systems Engineering. The complexity and interactive nature of modern avionics systems, within a larger aircraft system context, makes "Top Down" application a necessity.

The scope of modern avionics systems (large scale and complex) leads to the use of multi-disciplinary teaming for system development. The concept of multi-disciplinary development teams is relatively new, but has now become inherent in modern Systems Engineering practice. The interactive nature and complexity of integrated avionics requires teaming to establish needed information flow among the various engineering disciplines and their practitioners and is necessary for effective coordination and proper implementation of Systems Engineering.

## 2-0 SYSTEMS ENGINEERING POLICY

The importance of Systems Engineering is emphasized throughout the Defense Department acquisition life cycle. DoD instruction DODI 5000.2 "Defense Acquisition Management Policies and Procedures", establishes the need for an integrated framework for translating mission needs into a stable, affordable program that meets operational needs. It calls for a rigorous event-driven management process for acquiring products, emphasizing certain key Systems Engineering tasks.

### 2-1 Specific Systems Engineering Responsibilities

DODI 5000.2 In Part 6A, spells out specific Systems Engineering responsibilities to conduct in accordance with Mil-Std-499. These include: applying a comprehensive, iterative technical process throughout the system life cycle; assessing progress with periodic technical reviews and audits, transitioning applicable technologies into the development; managing risks; verifying requirements; integrating specialty requirements; maintaining configuration and data management; and utilizing a work breakdown structure. The specific Systems Engineering tasks identified include the following:

**(1) Translating Identified Needs into Design**

*This task is extremely important in assuring that identified needs, through the application of the Systems Engineering process, are translated into design requirements. Generally known as Requirements Analysis, this task is extremely important in assuring that the systems development is designed to the right technical requirements. In the past, failure to perform this task properly has led to failed system developments.*

**(2) Transitioning Technology from the Technology Base to Product and Process Applications.**

*This is an important task to ensure that emerging technologies are effectively applied to Naval avionics. Failure to effectively transition technology can result in premature obsolescence as avionics requirements continue to stress the technology base. This task is further complicated by the present emphasis on the use of Commercial Off The Shelf or COTS technology. By following Systems Engineering guidelines for avionics, a proper evaluation can be made that ensures an intelligent balance between the use of COTS and new technologies.*

**(3) Establishing A Technical Risk Management Program.**

*This task emphasizes that a major part of the Systems Engineering effort is that of managing technical risk. It is required that technical risks be identified, quantified, and impacts assessed and dealt with throughout the acquisition cycle. The Systems Engineering process includes risk management as an integral element of product and process development and requires the management of risk to acceptable levels. Technical, cost, schedule and other program risks are addressed at each technical and program review. Because advanced avionics require new and emerging technologies to satisfy technical requirements, the Avionics Program Manager is faced with a very difficult task in skillfully trading off performance gains versus technical risk.*

**(4) Verifying that Item Design Meets Established Requirements.**

*This task requires that a product (item design) undergo a comprehensive verification process that is established, implemented, and controlled as an integral part of the systems engineering effort. This task requires a progressive verification of the system development from parts, materials and subprocesses up through total system products and processes. This is an extremely important part of the process and is made especially difficult for advanced avionics systems development because of the extensive reliance on software. Satisfactory Software metrics and test routines comprise a major part of the avionics verification process. This is often made more difficult because for major systems the Hardware and Software is often developed in parallel, with the result that the system Hardware is usually not complete and therefore not available to demonstrate the software during the development process.*

**2-2 Specified Tools**

The two specific tools that are spelled out in DODI 5000.2 to assist management of the Systems Engineering effort are a Systems Engineering Management Plan (SEMP) and Technical Performance Measurement (TPM). Effective use of these tools is important to the Systems Engineering Process and will be described in a later section.

(see also: Volume 1, section 9-3.5)

### 3-0 THE SYSTEMS ENGINEERING FRAMEWORK

Systems Engineering involves developing system descriptions one level at a time; first as a concept, next as a system, and then as a set of subsystems. More and more descriptive detail is provided in successive levels of development. After each level of development, technical reviews are conducted to determine the maturity of the development, the affordability of the approach, and the readiness to proceed to the next level of development.

Systems Engineering is fundamentally a problem solving activity. The problem that is solved is a military need for a new or improved capability by a new or modified system. Systems Engineering defines this problem by determining the functional and performance requirements, constraints, measures of effectiveness, utilization environments, and external interfaces. This problem definition is then translated into solution criteria by decomposing problem level functions to lower level functions commensurate with the level of development and then allocating problem level performance requirements, constraints, etc. to the lower level functions.

Next, the Systems Engineering approach determines alternative solutions for each defined function and by means of analysis selects the best solution to satisfy total system needs. After each solution determination, a verification is made to demonstrate that the problem requirements are indeed satisfied by the solution selected. The solutions are described in appropriate documents (specifications and drawings) which provide details for the next level of development or for manufacturing of the products described.

It is important to emphasize that Software development is part of the total Systems Engineering development and should not be considered as a separate area. This is an important point since treating the Hardware development and the Software development as two distinctly different isolated efforts has led to major cost overruns and program failures. Systems Engineering is an umbrella for all the engineering efforts including the engineering specialties grouped under Supportability. A recommended approach to ensure that proper coordination and balance between the many engineering tasks is established and maintained, is to use multi-disciplinary Integrated Product Teams (IPTs) throughout development.

In essence, Systems Engineering provides a framework for development that is balanced against the primary functions that a system must satisfy. The Systems Engineering framework is shown in Figure 3-0 (from MIL-STD-499B).

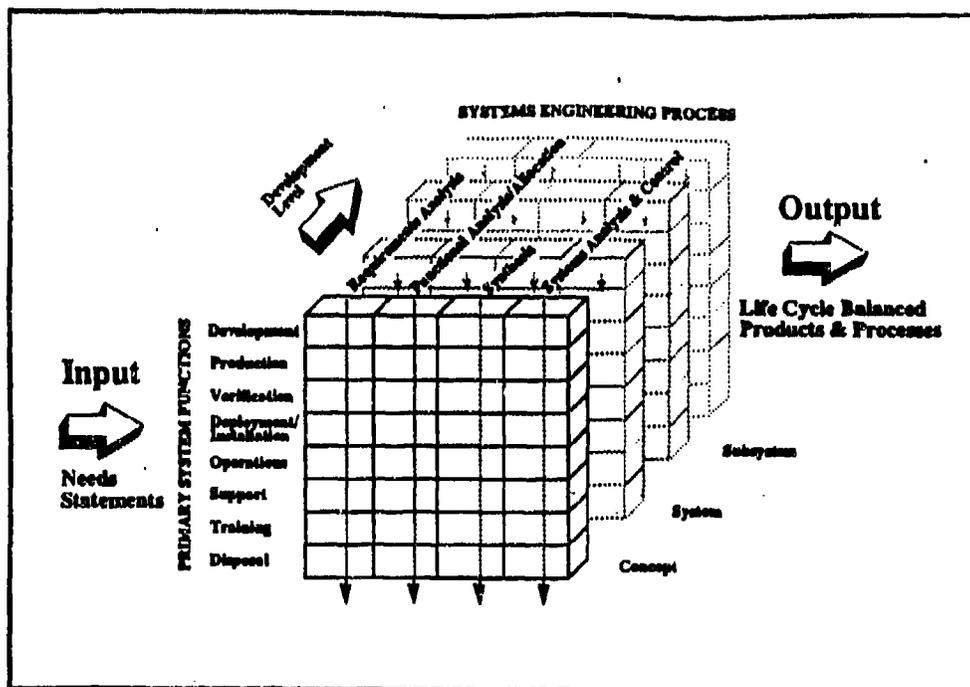


Figure 3-0 Systems Engineering process framework. (Source: MIL-STD-499B)

As shown in Figure 3-0, the original input to the Systems Engineering Process is the customer defined needs statements for the system. Any system can be described as a set of components working together with a common objective of performing a function in response to a designated need. The components of a system may consist of any combination of people, processes and products that satisfies the designated function. As indicated in the figure, the customer needs statements must be considered against the eight primary systems functions. The implication of this depiction is to emphasize that all functions are to be considered, not just development, so that a functional balance is achieved as the system requirements are evolved. Each layer of the Systems Engineering Process Framework represents a phase of the acquisition life cycle. As each layer is traversed the elements of the Systems Engineering Process are performed considering the eight primary functions throughout. As each acquisition phase is completed, the Systems Engineering outputs become the inputs to the next phase. As each successive phase is completed the products of the Systems Engineering process become more completely defined and matured.

### 3-1 Systems Engineering Process

The definition for the Systems Engineering process (from Mil-Std-499B) is as follows:

**Systems Engineering Process:** *"A comprehensive, iterative problem solving process that is used to (a) transform validated customer needs and requirements into a life-cycle balanced solution set of system product and process designs, (b) generate information for decision makers, and (c) provide information for the next acquisition phase. The problem and success criteria are defined through requirements analysis, functional analysis/allocation, and systems analysis and control. Alternative solutions, evaluation of those alternatives, selection of the best life-cycle balanced solution, and the description of the solution through the design package are accomplished through synthesis and systems analysis and control".*

The Systems Engineering Process is applied for each level of development of the system (concept definition, system definition, and subsystem definition) to provide a low risk transition from development to production. These levels of development are timed to coincide with the DODD 5000.1 and DODI 5000.2 acquisition management system controlled by the key milestone events of the program and the Program, Planning and Budgeting System (PPBS).

The Systems Engineering Process provides a comprehensive, simultaneous (concurrent) engineering approach for the development of the products and processes of the total system (design in all down stream processes needed to produce, verify, distribute, operate and support the desired operational system, including personnel training and disposal of process by-products and spent products). The Systems Engineering Process is applied recursively, and in an iterative manner, throughout system development. It is iterative in that the activities of the process are not linear but highly interactive to develop the best balanced set of process and process descriptions (Technical Data Package) appropriate for that level of development. In addition to being comprehensive and iterative for each application, the Systems Engineering Process (SEP) is also recursive when applied to successive levels of development. Because of its recursive nature, value is added by each application in that the output of one level is used as an input to the next level. Thus the definition of the system and its subsystems mature with successive applications of the SEP into a stable, affordable design.

The Systems Engineering process (as depicted in MIL-STD-499B) is presented in Figure 3-1, listing the major activities included in each of the four principal tasks. The figure indicates the iterative nature of the process as shown by the feedback paths around certain tasks. As the design details are solidified, feedback to the input of each process step is used to verify consistency of the design to requirements.

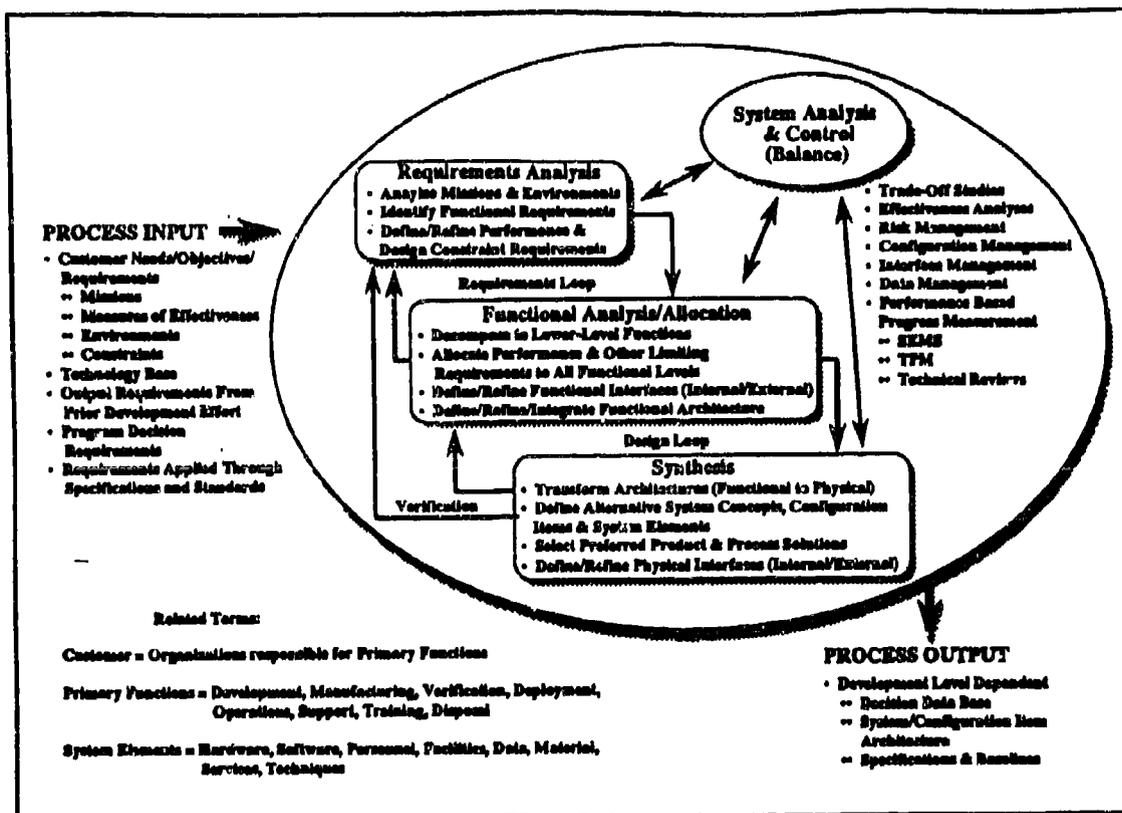


Figure 3-1 The Systems Engineering Process (Source: MIL-STD-499B)

### 3-2 The Principal Systems Engineering Activities of the Systems Engineering Process.

As described in MIL-STD-499B, there are four major activities performed as part of the Systems Engineering Process. These activities are the following: (1) Requirements Analysis, an activity which translates customer requirements into discrete functions; (2) Functional Analysis, an activity which decomposes all functions and allocates them to domains within a functional architecture of the system, (3) Synthesis, an activity which defines hardware and software items and assigns the required functions to those items; and (4) Systems Analysis and Control, an activity which includes the engineering management of system development, the balance of the other Systems Engineering Process Outputs and selection of the "best" candidate from physical and system architectures proposed as alternatives.

#### 3-2.1 The Primary Tasks of the Systems Engineering Process.

The Systems Engineering Process as defined in MIL-STD-499B involves the following four tasks:

- (1) Requirements Analysis,
- (2) Functional Analysis,
- (3) System Synthesis,
- (4) System Analysis and Control (Balance)

The activities within these four tasks, take customer inputs to the process and translate them into the desired process outputs.

### 3-2.2 Inputs To The Process : Requirements

Customer Requirements are broad in scope. They include mission needs, operational environments, and various constraints placed upon the development (such as costs, development time, the mandated use of legacy equipment, etc.). Other parameters which can be considered as requirements (or design constraints) include various Measures Of Effectiveness (MOEs), and the need to satisfy specific support environment requirements. Specifically, the Supportability requirements for Carrier operation distinguish the needs of Navy aircraft from those of the other Services. Inputs may include the output from a prior level of development (e.g., draft or approved specifications), the Requirements provided from Technical Reviews or from Program Decision Memoranda, as well as new or revised customer requirements. In early phases of the program, requirements are often very general and are easily and often modified. As the program proceeds through the acquisition process these requirements become more firmly established and become the basis for the preparation of contracts for the actual product development.

### 3-2.3 Requirements Analysis

Requirements Analysis generates information on what the system must do and how well it must do it. Process inputs provide the source from which these top level technical, functional and performance requirements are generated. In essence, Requirements Analysis defines the problem. Through Requirements Analysis, we establish utilization environments, and design constraints (e.g., system level functional and performance requirements and external interfaces) in quantifiable characteristics and tasks for the eight primary functions (e.g., development, manufacturing, verification, deployment, operations, support, training and disposal).

### 3-2.4 Functional Analysis

Through Functional Analysis, the functional architecture of the system is defined. This is done by identifying successive levels or sub functions necessary to accomplish upper level functions. The sub functions are arrayed in a functional architecture which shows relationships and interfaces. Upper-level performance requirements are flowed down and allocated to lower-level sub functions. With each level of development one or more sub functional levels are added to the functional architecture based on the physical solutions generated from the prior application of the SE Process.

### 3-2.5 Synthesis

Through Synthesis, the functional architecture is translated into a physical architecture. Groups of functions from the functional architecture form Configuration Items (CIs). Synthesis activities generate alternative physical solutions for each CI. A CI can be made up of one or more of the following system elements: hardware, software, data, personnel, facilities, material, services or techniques. These system elements provide physical building blocks that, in combination, make up the physical architecture of the system. Done rigorously, the system Synthesis process is supported by the extensive use of computer based trade studies and system effectiveness analyses.

*Recommendations from this Review strongly support the use of Open Systems Standards as the "building block" elements and interfaces employed in synthesizing systems in order to expand the industrial supplier base. The complexity of modern avionics systems requires a major utilization of computer models and simulation tools to accomplish the Synthesis function.*

### **3-2.6 System Analysis and Control (Balance)**

This task provides the assessment and control necessary to manage the development process. This task is essentially one of achieving an effective balance between cost, schedule, performance and risk suitable for the specific development. To accomplish this task, the development team makes use of Trade Studies, Effectiveness Analysis, Risk Management, Configuration Management, Interface Management, Data Management, Performance Based Progress Measurements (through use of SEMS, TPM, Technical Reviews) and other necessary activities to ensure development balance and control. This task covers the activities necessary to manage system development and the conduct of the systems engineering process. The performance of this task is determined by the use of performance based progress measurements that provide indicators of how well system development is progressing.

The Systems Engineering activities included in the Systems Analysis and Control task of the Systems Engineering Process are described below. Successful application of these activities lead to a balanced set of product and process solutions to the development problem.

**Trade Studies** - A trade study (also called trade-off analysis) is a process whereby viable alternatives are examined to determine which is preferred. The results of this iterative process are used to support technical decisions concerning system concepts, requirements or design solutions. The trade-off methodology provides an approach to decision making that is rational, objective, and repeatable.

**Effectiveness Analyses** - Effectiveness analyses are used to: support identification of mission and performance objectives and requirements; support allocation of performance to functions; provide criteria for the selection of alternative solutions; provide analytic confirmation that design solutions satisfy customer requirements/needs; and support product and process verification. The results of these analyses are used in trade studies to determine the best alternative solution set.

**Risk Management** - Technical risk management is an organized, analytical process to identify what can go wrong with a technical development effort, the consequences if it does go wrong, and a method of either preventing or handling the resulting problem(s).

**Configuration Management** - Configuration management is a disciplined approach to applying technical and administrative direction and surveillance over the life cycle of Configuration Items (CIs) to: identify and document the functional and physical characteristics of CIs; control changes to CIs and their related documentation; record and report information needed to manage CIs effectively; and audit CIs to verify conformance to specifications, drawings, interface control documents, and other contract requirements.

**Data Management** - Data management includes ensuring that a proper data call is accomplished, that only needed data are asked for in the contract, that data are properly transmitted, received, stored, and handled. It is recommended that data be delivered and stored in digital form to be Computer-Aided Logistics Support (CALs) compliant.

**Interface Management** - Interface management is essential to ensure that system elements are compatible in terms of form, fit, and function. Both internal and external interfaces must be managed. Interface controls must be established to ensure all internal and external engineering interface requirement changes are properly recorded and communicated to all affected CIs.

**Systems Engineering Master Schedule (SEMS)** - The SEMS is an event based scheduling system to plan all technical program events (e.g., a design review or audit) that need to be accomplished, the significant activities that must be accomplished to complete an event, and the criteria by which each activity within an event is judged to be completed.

**Technical Performance Measurement (TPM)** - TPM is the technical performance measurement equivalent to Cost/Schedule Control System Criteria (C/SCSC) used for cost and schedule progress measurement. Selected critical technical parameters are tracked to ensure that key performance requirements are progressing in development over time as planned. This is a management by exception performance based progress measurement.

**Technical Reviews** - A technical review is conducted between (and often during) each level of development to determine the maturity of the development effort, the amount of risk inherent in a continued effort, the affordability of the design solution, and whether the investment should be made to continue development.

### **3-2.7 Outputs of the Process : Decision Data Bases; Specifications, Baselines and a System Architecture**

Program Peculiar Specifications (hardware and software), Interface Control Documents, Technical Data Package, Decisions, etc. The overall end product of the Systems Engineering process is the technical data package: specifications, drawings, materials lists, parts lists as applicable for all product and support hardware and software, and processes/planning for manufacturing and support. This product matures through successive applications of the Systems Engineering process during each level of development.

## 4-0 SYSTEMS ENGINEERING THROUGHOUT THE LIFE CYCLE

The Government acquisition process for major defense systems, as defined in DODD 5000.1 and DODI 5000.2 consists of five major milestone decision points and five major acquisition phases as illustrated in Figure 4-0. The purpose and responsibilities for each phase are also identified in the figure. The acquisition life cycle is provided as a basis for the comprehensive management and progressive decision making required to successfully develop a modern weapon system. The Major Technical Reviews and Audits shown in the figure are not completely consistent with MIL-STD-1521B, *Technical Reviews and Audits for Systems, Equipment and Software.*; rather they refer to the Reviews and Audits that are consistent with MIL-STD-499B(5/6/92 Draft) as included in APPENDIX C. With formal approval of MIL-STD-499B, this new set of reviews and audits will be adopted and MIL-STD-1521B will be superseded.

Systems developments generally go through the following phases as defined in the DoD acquisition life cycle:

The Concept Exploration & Definition (CED) phase is generally conducted by a small group organized to form a program office within the contracting agency. This group may draw on outside contractor or consultant support to provide expertise in areas where it is lacking. It evaluates broad concepts that may be capable of satisfying the mission need. It attempts to quantify user requirements and assess the capability of technology to meet the conceptual requirements. If current technology is not adequately developed, contracts may be awarded to bring hardware and software from the laboratory stage to an operational state.

The Demonstration and Validation (Dem & Val) Phase involves several contractors in competition. The effort in this phase is to establish a firm set of system performance specifications and to allocate them to lower levels in order to begin to detail design features. Concurrently, a design is defined in enough detail that its performance can be analyzed to establish its capability to meet system level requirements. Trade studies are conducted to achieve balance in the design in regard to cost, performance, schedule and operational support goals. During this phase, requirements are allocated from the system level down to hardware and software elements.

The Engineering and Manufacturing Development (EMD) phase is usually performed by a single prime contractor, although several contractors may be selected to build prototypes and to engage in a demonstration before selection is made. During EMD, the system moves into the detailed design state, followed by fabrication, assembly and "first article" testing. Several major reviews are held during this phase. A Preliminary Design Review (PDR) and a Critical Design Review (CDR) are held to examine how well the design meets requirements based on simulation and analysis. Following the testing period, a formal qualification review is held to certify that the system has demonstrated a capability to meet operational requirements. The detailed software requirements specifications are developed and completed before PDR. The software design is completed before CDR and then proceeds into coding and checkout.

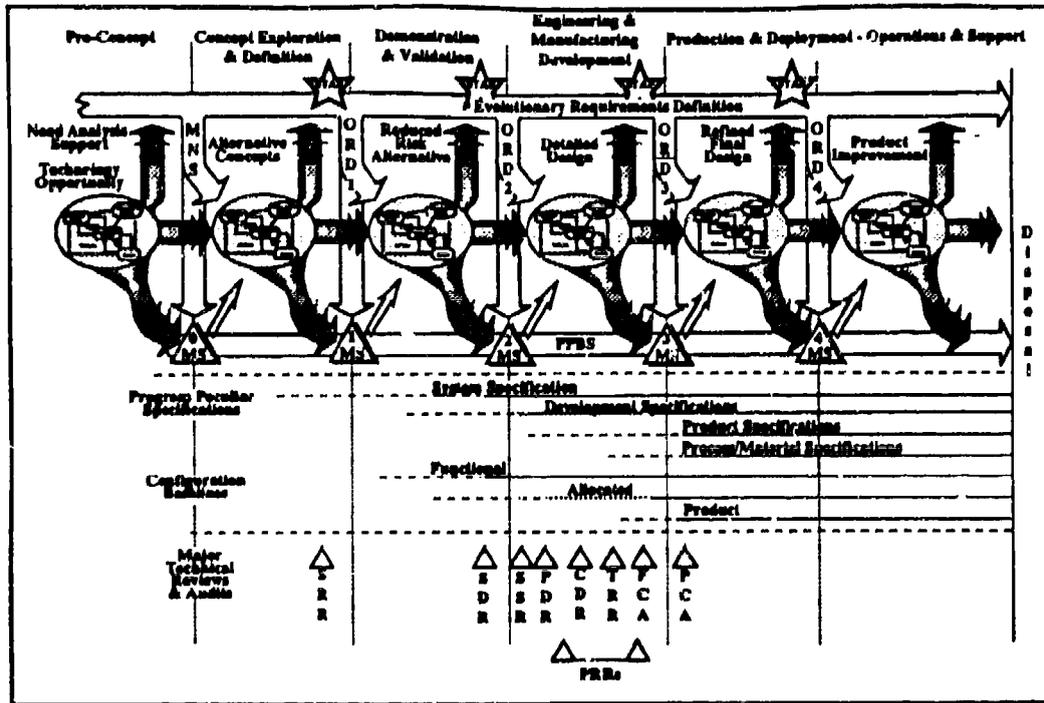


Figure 4-0 Iterative life-cycle application of the Systems Engineering Process (Source: MIL-STD-499B)

This description may take the form of a set of draft system specifications and corresponding sketches that are documented in an operational concept document and decision data base. The first level of development also determines the physical type of system to be developed (e.g., a tank, an airplane, a ship, a missile, a satellite, a circuit board, a radio, etc.) to satisfy the mission needs statement or modification need statement.

The second level of development determines the specific external characteristics and interfaces, and functional and performance requirements for the total system (considering all eight primary functions - development, manufacturing, verification, deployment, operations, support, training, and disposal). This description is a more detailed set of sketches and systems specifications for the hardware, software, personnel, data, materials, services, facilities and techniques that make up the physical architecture of the system. This system-level set of descriptions is called the functional baseline and includes the system specification (Type A Specification). Another output of this second level of development effort is the draft functional architecture and physical architecture that describes the next level below the system. These architectures make up a draft functional and performance requirements description (Draft Type B Specifications) of the major subsystem and/or configuration items that will make up the system. It is these descriptions, along with the system description, that the government uses to define development contracts.

The third level of development effort results in a more detailed set of functional and performance requirements, with both internal and external interface descriptions and constraint descriptions, for the subsystems/configuration items below the system level. This functional architecture is then translated into a physical architecture (hardware, software, etc.) providing physical solutions to the functional architecture requirements.

Successive applications of the Systems Engineering process are made to each subsystem/configuration item to develop the functional and physical architectures of the components, elements and parts of the subsystem/configuration item to arrive at the development baseline (including the Type B Specifications) and the product baseline (including the Type C and, if requested, D and E Specifications).

For each level of development the realization of a complete functional and physical architecture is dependent on the technical inputs of a variety of engineering specialties, as well as by the business requirements from contracts, funds management, budget, personnel, and marketing. These specialists help define requirements, and apply knowledge from their specific discipline into the development/design efforts.

To ensure that conflicting requirements and/or interests from these technical and business specialties are resolved efficiently and effectively, so that the architectures are integrated and include all needed down-stream life cycle requirements, an integrated multi-disciplinary effort is needed. The integration approach currently used by competitive industries and some government program offices is that of integrated product teams or multi-disciplinary teams. These teams are composed of the appropriate permanently assigned specialists needed for the level of the development effort. Team members are added and/or teams are added as the architecture levels increase and more detailed development/design results are required.

## **5-0 PLANNING FOR SYSTEMS ENGINEERING**

Successful implementation of any development program requires early and comprehensive planning. Planning for systems engineering commences at the inception of the program. Planning begins with the definition of program requirements. The various System Engineering functions and tasks are identified and a detailed work breakdown structure is prepared.

### **5-1 Management Plans**

Systems Engineering Planning is documented in the Systems Engineering Management Plan (SEMP). Individual SEMPs are utilized by both the government and the contractor to govern the Systems Engineering activities of each.

#### **5-1.1 Systems Engineering Management Plan; Background**

The Systems Engineering Management Plan (SEMP) is the tool used to document the technical processes of the development program in order to ensure well coordinated system development. The SEMP outlines what steps are planned to accomplish the Systems Engineering effort. MIL-STD-499B recommends that a SEMP be prepared and followed by both the Government (customer) and by the contractor.

#### **5-1.2 Government SEMP**

The government SEMP is a planning document that outlines the Systems Engineering activities/tasks that must be carried out by either the government or contractors throughout the life-cycle of the program. It is an essential part of the Program Management Plan. The government systems engineer writes the SEMP, tailoring it to the specific system and organizational environment.

The government SEMP provides the engineering community and the program manager with an overview of the key Systems Engineering Process tasks required to be executed. It also describes the technical organizational structure to enable development tasks to be accomplished by a multi-disciplinary effort (e.g., integrated product teams). In addition, this plan provides plans and criteria for transitioning critical product and process technologies into the development effort at the appropriate time, or for the evolutionary development or pre-planned product improvement of technologies by IR&D or government laboratories in parallel to the system development effort by the program office. Another section of the SEMP describes the key trade studies, system effectiveness analyses, and other system analyses and controls planned (e.g., Risk Management, Configuration Management, Data Management, Interface Management, SEMS, TPM, and Technical Reviews). Engineering specialty plans (e.g., configuration management, ILSP, TEMP, manufacturing management, and other specialties such as reliability, maintainability, etc.) may be attached to the SEMP or a short summary provided to demonstrate the integration of the engineering specialty areas.

### 5-1.3 Contractor SEMP

The SEMP defines the contractor's, or performing government agency's, plan for the conduct and management of the fully integrated engineering effort necessary to satisfy the general and detailed requirements of Mil-Std-499 as implemented by the Request for Proposal (RFP) or contract schedule, statement of work, and specifications. The contractor, or performing government agency, uses the SEMP to: (1) document the decisions and resulting technical implementation necessary to satisfy contract requirements and (2) communicate that approach both internally and externally. This SEMP will be used by the government program office to understand and evaluate the contractor's proposed Systems Engineering work during proposal evaluation and/or as part of the contract monitoring process. The contractor systems engineer writes the SEMP, tailoring to the specific system and organizational environment.

A new SEMP Data Item Description (DID) is provided with MIL-STD-499B. This DID specifies the information that the contractor should provide in each of four sections of the SEMP. Section 1 focuses on the planned Systems Engineering Process application for the contractual effort. Section 2 describes the planned approach for transitioning critical technologies. Section 3 describes the integration of the Systems Engineering effort to include how the multi-disciplinary effort will be implemented. Section 4 provides a description of the additional Systems Engineering activities required to complete the contractual efforts (e.g., engineering tools; systems integration plans; compatibility with production, test and support activities; long lead items; and other plans and controls). The purpose of the DID is to provide sufficient information to the contractor for preparing the SEMP to allow him to tailor his Systems Engineering approach to the specific system acquisition for maximum effectiveness. It is desired to place the tailored SEMP on contract, not MIL-STD-499B itself.

## 5-2 The Use of System Effectiveness Metrics and Overall System Cost Criteria in Planning

The Relationships among key system effectiveness metrics and Life-Cycle Cost (LCC) and Design to Cost (DTC) requirements and their relationships to operational suitability, effectiveness and affordability are described.

Four facets of system effectiveness are:

**Cost Effectiveness** - a measure of affordability and life-cycle cost as a function of design-to-cost, suitability, dependability, and capability.

**Suitability** - a measure of the degree to which an item is appropriate for its intended use. (Is it the right system for the need?) Includes consideration of compatibility, inter operability, transportability, man-machine interfaces, training, safety, security, and documentation.

**Dependability** - a measure of the degree to which an item is operable and capable of performing its required function at any (random) time, given its suitability for the mission. (Will it be available and operate when and as long as needed?) Includes availability, reliability, usage rates, durability, survivability, penetrability, vulnerability, mobility, flexibility, and repairability.

**Capability** - a measure of the system ability to achieve the mission objectives, given the system is dependable and suitable. (Will it get the job done when engagement takes place?) Includes accuracy, payload, range, lethality, ability to destroy, number of engagements, and information rates.

Key definitions and relationships of these system effectiveness facets are:

**DT&E Metrics** - The four above elements of systems effectiveness are Development Test & Evaluation (DT&E) metrics that must be satisfied by the system/CIs during development. These are the measures (DT&E metrics), in the form of program peculiar specifications, that the development testers (government and contractor) must verify and the configuration management personnel must audit to ensure that the product and processes of/from the development effort do in fact satisfy.

**OT&E Metrics** - The system provided by the government developers to the users must be operationally effective and operationally suitable. These measures of effectiveness are translated into measures of performance called Operational Test & Evaluation (OT&E) metrics during requirements analysis (an element of the Systems Engineering Process) to provide performance requirements for the functions that must be accomplished by the system. The operational test community performs independent testing to ensure that the system delivered does in fact meet the OT&E metrics.

**Affordability** - Characteristics of the product with a price approaching its functional worth and within the limits of what the government is able and willing to pay/invest.

**Design to Cost (DTC)** - A method of evaluating cost and technical performance so that they are relatively equal. The essence of the DTC effort is making the design converge on cost instead of letting design determine the cost.

**Life-Cycle Cost** - The total cost of acquiring and utilizing a system over its entire life span. The LCC includes all RDT&E, production, operations and support, and disposal costs. It is the role of Systems Engineering to minimize this for affordability while meeting operational suitability and effectiveness requirements.

## 6-0 SYSTEMS ENGINEERING OUTPUTS

### 6-1 SE Baselines Used in Acquisition Management

The technical baselines (*functional, allocated and product*) that identify and define an item's functional and physical characteristics are used in Systems Engineering to document the outputs of the Systems Engineering process. These baselines are referred to as configuration baselines and are guided by MIL-STD-973, Configuration Management.

Three other baselines that are managed by the program manager are called program baselines. These baselines (*concept, development, and production*) embody the cost, schedule and performance objectives of the program. These baselines are required on all programs for measuring and reporting status of program implementation and guided by DODI 5000.2, Part 11A and DOD 5000.2M, Part 14.

### 6-2 Specifications; General and Program Peculiar

The initial definition of system requirements is projected through a combination of formal specifications and planning documentation. Specifications basically cover the technical requirements for system design.

A specification is a document prepared specifically to support acquisition which clearly and accurately (hopefully) describes essential technical requirements for purchasing materiel. Procedures necessary to determine that the requirements for the materiel covered by the specification have been met are also included as specifications (Source MIL-STD-961C - *Military Specifications and Associated Documents, Preparation of*). There is no definition for a specification given in MIL-STD-490A or B (draft) - *Program-Unique Specifications, Preparation of*. MIL-STD-973, *Configuration Management*, references MIL-STD-961 for the definition.

MIL-STD-490 is closely related to MIL-STD-961 in that they both address the preparation of specifications used by DoD. However, there are significant differences between the types of specifications prepared under each standard and they serve different purposes. Specifications prepared under MIL-STD-490 are program-unique documents necessary to control item configuration and establish baselines. Program-unique specifications are unique to a particular weapon system or program, and little or no potential exists for the application of these documents to other systems or programs. Furthermore, program-unique system, development, product, material and process type specifications are used when it is the Government's intent to control, procure, and support the exact design of the item; or if the specifications are intended to control the process or material during development phases.

In contrast, specifications prepared under MIL-STD-961 have or could have multiple applications and are needed to support the goals of the Defense Standardization Program to limit the variety of equipment and supplies in military supply systems [ref: DOD 4130.3-M].

### 6-3 Specifications used in Systems Engineering

In dealing with large systems, there are numerous elements that must be covered by specification (dictating technical requirements) and by planning documentation (providing the "HOW", "WHEN", and "WHERE" information pertaining to program implementation and control). Some components of the system may require an extensive amount of research and development effort, while other components are obtained directly from existing supplier inventories. Depending on the degree of development required, and the uniqueness of any processes applied to development, there may be a large variety of specifications necessary to provide the guidance and controls necessary for the development of the system and its components.

Both types of specifications are needed for the SE effort. For product developments, system level specifications under MIL-STD-490 are needed to baseline system functional performance. The systems specification is normally developed during the DEM/VAL phase prior to EMD. Development specifications are prepared for individual configuration items (CI) during EMD in accordance with MIL-STD-490. Then for each CI a product specification is developed along with needed material and process specifications under MIL-STD-490. In addition, military General Specifications that have been developed under MIL-STD-961 and tested in the Acquisition Management Systems and Data Requirements Control List (AMSDL) are also used to guide product developments and manufacturing requirements.

Like other aspects of Systems Engineering there is a hierarchy to the specifications which begins with top level systems specifications and tends to flow down to those at a lower level which address details of fabrication and process.

Figure 6-3a presents a hierarchy of technical specifications showing where each type resides in an overall set of specifications. The figure also gives some indication as to how the level of detail increases as you move from top level (type "A" to lower level specifications. The classification and description of specifications are the following:

- (1) System Specification (Type "A") Includes the technical, performance, operation and support characteristics for the system as an entity. It includes the allocation of requirements to functional areas, and it defines the various functional area interfaces. Information derived from the feasibility analysis, operational requirements, maintenance concep., and the functional analysis is covered by Type "A".
- (2) Development Specification (Type "B") This type includes the technical requirements for any item below the system level where research, design, and development are accomplished. This may cover an equipment item, assembly, computer program, facility, critical item of support, and other like items. Each specification must include the performance, effectiveness, and support characteristics that are required in the evolving of design from the system level and down.
- (3) Product Specification (Type "C") Included are the technical requirements for any item below the top system level that is currently in the inventory and can be procured "off the shelf". This may cover standard system components (equipment, assemblies, cables, etc.), or a specific computer program, a spare part, a tool, and like classes of components.

(4) **Process Specification (Type "D")** This type of specification includes the technical requirements that cover a service that is performed in the preparation/fabrication of any component of the system. Examples may include machining, bending, welding, plating, forming, heat treating, marking, packing and any other necessary process that requires definition.

(5) **Material Specification (Type "E")** This specification type describes the technical requirements that apply to raw materials, mixtures (e.g., paints, chemical compounds, etc.), and/or semi-fabricated materials (e.g., electrical cable, piping, etc.), that are used in the fabrication of a product used in the system.

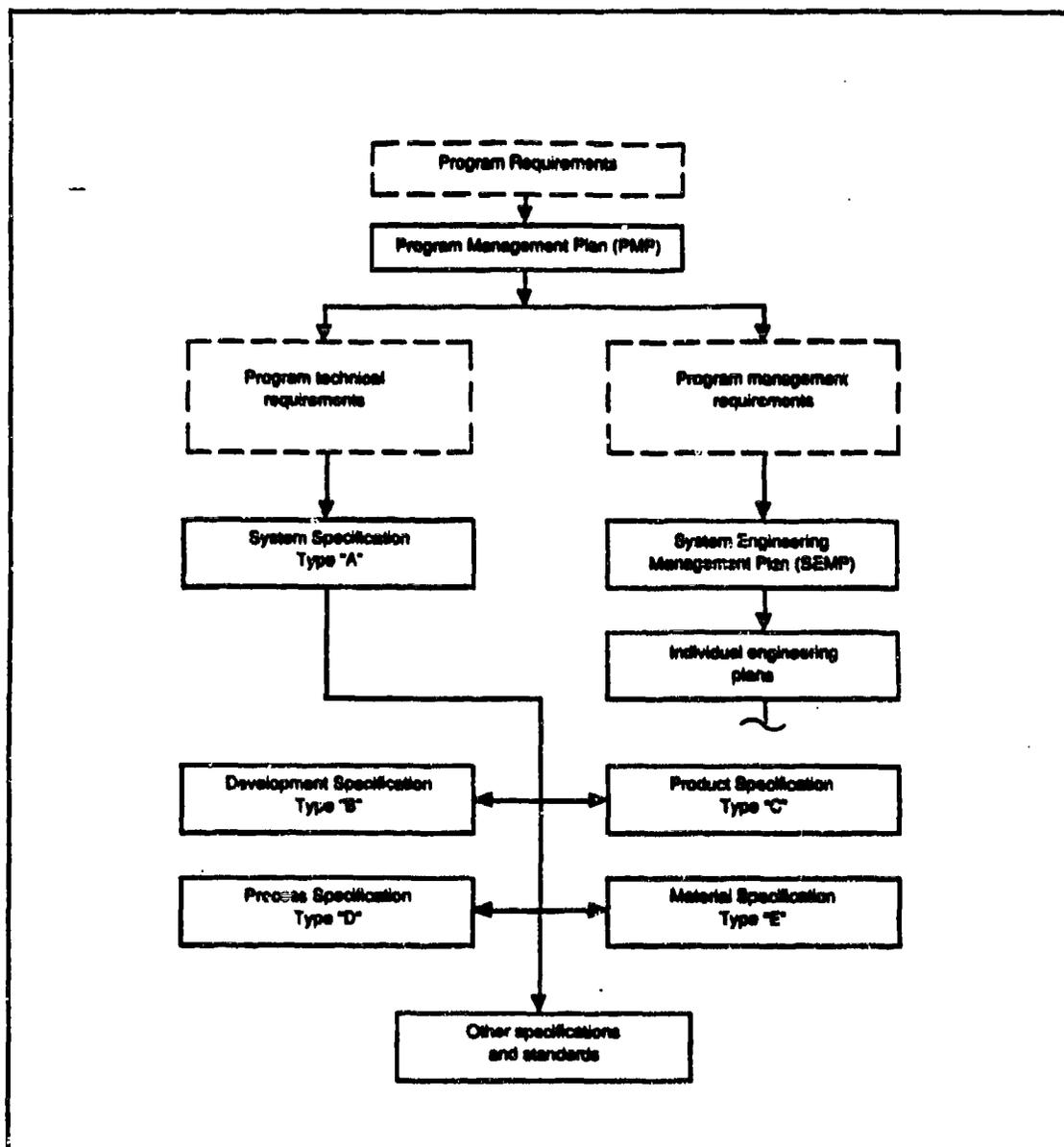


Figure 6-3a Hierarchy of technical specifications  
 (from, B.S. Blanchard, *System Engineering Management*, with permission)

Example specifications used in Systems Engineering include the following:

*Draft System Specification - Output of Concept Level of Development (Concept Exploration & Definition Phase)*

*System Specification - Output of System Level of Development (Demonstration & Validation Phase)*

*Draft Development Specifications - Output of System Level of Development (Demonstration & Validation Phase)*

*Development Specifications - Output of the Engineering & Manufacturing Development (EMD) Phase, as early as preliminary designs or as late as the Functional Configuration Audit (FCA)*

*Draft Product Specifications (Material & Process) - Output of EMD Phase*

*Product Specifications (Materials & Process) - Output of Initial Production Lot*

Preparation of specifications is an important engineering activity. The system specification is prepared at program inception during the conceptual design phase. Development and product specifications are often based on the results of trade studies, "make-or-buy" decisions and the like. They are generally prepared during the preliminary design phase. Process and material specifications are primarily oriented to production activities and are normally prepared during the full scale design development phase. Specifications are documents utilized primarily for acquiring items, including the procurement of off-the-shelf components, contracting for the design and development of a new item, testing and verification of a product, and the like. Specifications may be applied on contracts and imposed on major contracts, subcontractors and suppliers for goods and services. Specifications are of necessity requirements oriented and must be written in a clear and concise manner. Vague, redundant, nebulous, and ambiguous language should be eliminated. Requirements should be quantifiable and verifiable, and should not require judgment in interpretation. Phrases such as "best design practices" or "good workmanship" should be avoided. In applying specifications, care must be exercised to ensure that they are prepared to the proper depth of detail and applied at the appropriate level in the system hierarchy. Such documents must be detailed to the extent required to establish the basis for appropriate design and the application of suitable materials and processes. For complex specifications, best practice is to extend the concept of document inspections first applied in software development to a review of requirements documents. A document inspection team consisting of a trained experience moderator (leader) together with a team of reviewers (multi-disciplined as required) and the documents author is convened to verify and validate the requirements included in the specification document. Through a disciplined, structured approach the document is reviewed to determine whether the requirements are complete, are necessary and clearly and unambiguously presented.

Figure 6-3b provides a sample specification tree (simplified) for a typical system showing the relationship of each of the specification types within a typical system family tree.

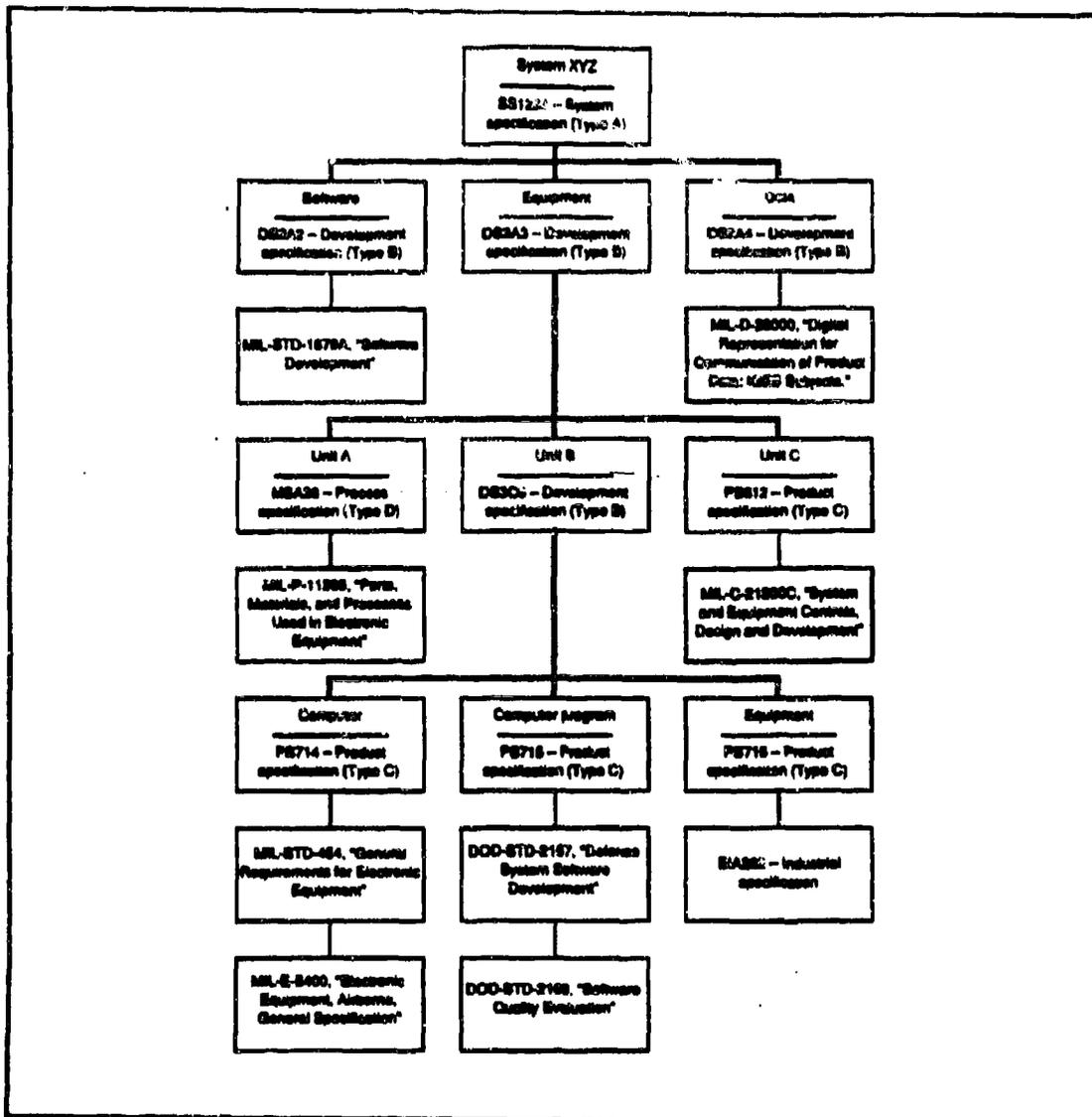


Figure 6-3b Sample Specification/Documentation Tree  
(from, B.S. Blanchard, *System Engineering Management*, with permission)

The figure shows the hierarchy of a documentation tree. The tree is developed from the top down starting with the preparation of the top level system specification. Subsequently additional specifications are added, with additional detail added at each level. The tree is a combination of specifications and reference standards.

The critical task is in the efficient tailoring of the specifications developed to the particular system to which they are applied. Even though design needs may dictate the use of an off the shelf component, specific requirements of the application may be quite different from those in which the component has been previously applied. In some instances the application may dictate a re-engineering or "militarization" of the commercial component to satisfy needs of the specific application. A balance must be struck between "over specification" (a complaint of many contractors who participated) and "under specification" resulting in designs that do not meet operation needs.

## 7-0 REQUIREMENTS ANALYSIS

### 7-1 The Requirements Analysis Process

Requirements analysis helps both the government program office and the contractor in problem definition. Requirements are essentially a statement of the problem to be solved by the development agencies. It is essential that all involved organizations understand what the system is suppose to do, where the products of the system will be used, under what conditions the products will be used, and who will use the products of the system. The results of this analysis provides developers with an understanding of WHY the system is needed. From this understanding the program-unique specifications (the WHAT) are produced by developers that explain to designers the product and process characteristics required to be satisfied by the design (the HOW).

### 7-2 Inputs to and outputs of the Requirements Analysis portion of the Systems Engineering process.

Three essential outputs are produced by the requirements analysis.

- (1) An operational description that explains the operational need; system missions; operational sequences; operational environments; conditions and/or events to which the system must respond; constraints on the system; user roles; organizations that will operate, support and maintain the system; and operational interfaces with other systems.
- (2) A functional description that documents in a decision data base the tasks that must be performed by the system; the qualitative, quantitative and timeliness performance requirements/constraints of the system ; interface requirements; and verification requirements.
- (3) A physical description, also documented in a decision data base, that provides physical interface data and characteristics of information displays and operator controls; relationships of operators to system equipment; characteristics of the users such as special operational environments and movement or visual limitations; and system performance characteristics such as physical capacity, power, size or weight limitations, technology limitations or GFE, NDI, COTS or reusability requirements.

There are three types of inputs to the requirements analysis used to produce the necessary outputs.

- (1) Those inputs that are converted through the analysis process into the desired outputs (mission needs statement (MNS), operational requirements document (ORD), measures of effectiveness (MOEs), and program-unique/peculiar specifications developed during the prior development effort).
- (2) Those inputs that control the requirements analysis process such as organizational policies and procedures, military (general) specifications and standards, constraints (technology, funds, schedule, skills, etc.), and utilization environments.

- (3) Those inputs needed to enable the requirements analysis to be effective such as multi-disciplinary product team efforts, an effective decision data base that tracks prior efforts, changes and decisions; and tools, usually computer based, that make the analyses more efficient.

### **7-3. The Role of Trade Studies in the Development of Functional and Performance Requirements.**

Trade studies are required because requirements from different customers (developers, manufacturers, verifiers, deployers, operators, supporters, trainers, and those responsible for disposal) often conflict, imposed constraints limit options and resources are not unlimited. Trade studies are used for the following reasons:

- (1) To resolve conflicts among mission objectives, input constraints, human factor considerations and utilization environments;
- (2) To ensure that a cost effective balance of functional and performance requirements are developed; and
- (3) To reduce technical risks so that there is a higher probability that all customer problems will be solved.

The role of trade studies therefore ranges from helping to identify the proper set of system requirements, supporting the control function to assure that a balance is achieved among the various functional needs of the system, and as a risk assessment tool to establish the maturity of candidate technologies and to compare the efficacy of alternate system concepts. The emphasis areas for trade studies change throughout the various acquisition phases as described below:

*Concept: Exploration and Definition - During this phase, trade studies focus on comparing various technologies and approaches. Various concepts for meeting mission needs are compared and traded off during this phase. Trade studies are used to select alternate system configurations to explore in later phases.*

*Demonstration/Validation - Trade studies are used during this phase to select preferred technologies and to reduce system alternatives to a testable number.*

*Engineering and Manufacturing Development - The trade study emphasis during this phase is directed to selecting among components/part designs, selecting among testing methods and selecting suitable support processes.*

*Production and Deployment - Trade study activity is directed toward comparing the advantages of proposed design changes, evaluating new mission requirements and assess the merits of incorporating new technologies.*

### **7-4 The Importance of Requirements Analysis to Systems Engineering.**

Requirements Analysis is critical. Many programs are doomed to failure largely because the requirements are not fully or correctly refined. The most efficient system development process is of little importance if you build the wrong system.

Application of the Systems Engineering process to military avionics is complex in application because of the complexity of high performance avionics systems. The process is often unduly compromised when avionics system development is constrained by other

design choices imposed on the aircraft as a total weapon system. As a Mission Needs Statement is first translated into an aircraft design, avionics is only a secondary consideration. The aircraft structure, aerodynamic properties, engines and control systems are usually the first design elements considered in the development of the aircraft as a weapon system. Often the avionics Systems Engineering team is not part of the decision process at this level, resulting in sometimes unnecessarily limiting the avionics systems choices to costly, stringently complex designs. For the most efficient, cost effective process, the avionics Systems Engineering team (especially the avionics architect) must participate in the early definition process where high level aircraft systems choices are made.

In detailing (specifying) the avionics it is important to start with a "clean sheet of paper" when possible. Too often, the preliminary avionics specification is derived from the most recent aircraft of a similar type. We then attempt to "stretch" the specification by increasing the performance specified in certain areas where we feel the technology has matured to allow a performance increase without undue risk. In contrast, the "clean sheet" of paper concept allows the design to be focused on the newly specified avionics needs for the weapon system rather than to carry over specifications from the predecessor aircraft that may no longer be necessary. This approach can help to avoid overly complex avionics, often characteristic of the conventional specification method. Reviewing the specification of the predecessor aircraft is useful, but each requirement retained should be thoroughly validated before it is retained for the new aircraft development. Because of the significance of requirements analysis the term "Requirements Engineering" has been coined and used to describe this aspect of engineering. Specific computer aided tools have also been developed specifically to support the several aspects of Requirements Analysis.

## **8-0 FUNCTIONAL ANALYSIS, SYNTHESIS AND WORK BREAKDOWN STRUCTURES**

### **8-1 Evolution of the Functional Architecture to a Physical Architecture**

Functional Analysis consists of decomposing the higher level functions that came out of requirements analysis into lower level functions. The operational, performance and other requirements are then allocated to these functions at the appropriate level. Interfaces are defined and from this process a functional architecture of the system is developed. It is important to emphasize that the process includes a looping back up to the requirements analysis activity to constantly evaluate whether or not the requirements are met as the functional architecture evolves.

### **8-2 The Evolution and Uses of the Engineering WBS**

As each level of the system's architecture evolves, a product and process oriented Work Breakdown Structure (WBS) can be developed. There is a product element of the WBS assigned for each product specification derived for a physical architecture. As the levels of physical architecture increase, so do the levels of the WBS. This type of WBS can be referred to as the engineering part of the program or contract WBS because it depicts the product structure for which engineering efforts must be applied to satisfy customer requirements.

The engineering WBS can also be effectively used as an organizational model to organize the development and to assign integrated product or multi-disciplinary teams. This approach, for example, is used in several Air Force aircraft and missile programs. Each team is given one product of the WBS, the resources needed for its development, and is held responsible for its development. The team is given:

- (1) The responsibility to meet allocated requirements (including integration with interfacing products),
- (2) The authority to accomplish the tasks needed, and
- (3) The accountability for the funds associated with that development effort.

Appropriate specialists from the product teams are assigned to the process/functional elements of the WBS related to the prime item equipment of which their product team is a part. For an example, the support specialists from each lower-level product team working to support a radar (prime item equipment) development would be assigned to a support functional/process team to ensure that the system elements (hardware, software, personnel, etc.) needed to support the radar are developed along with the radar products. This approach ensures that specialized support needs of all radar product elements can be fully supported. Likewise, the engineering WBS can be used to assign management teams. For example, the radar management team can be composed of team leaders of each of the product teams for the products that make up the radar. In like manner, if the radar is the prime item equipment in a fire control subsystem, the fire control subsystem team can be made up of team leaders of each of the functional/process teams and the radar team. This "team of teams" approach helps to ensure the integration of all products that make up a subsystem by having team members that are responsible for the success of their individual products and the subsystem. As with the product teams, appropriate business specialists need to be assigned to appropriate teams. Each team (subsystem, prime equipment and

functional) may also have an assigned leader, staff and specialty engineers assigned to ensure the integrated development of their related product and process system elements. The IPT approach supports the use of incremental technical reviews and audits. Much of the review activity at the lower level WBS elements is accomplished through team reviews and regular meetings. Major reviews can be managed by having a series of incremental reviews at subsystem and functional levels. As reviews at the lowest reviewable level are completed, the incremental reviews can proceed to higher and higher levels in the Engineering WBS. When the major reviews such as the PDR and the CDR are conducted, they consist largely of demonstrations of compliance reflecting the earlier reviews at lower levels. In this manner it hoped to avoid the large scale reviews witnessed in major system programs when hordes of reviewers descend upon a contractor to conduct simultaneous reviews of all key program elements usually breaking into "splinter groups of specialists reviewing each significant program element. Reviews of this type are not usually very efficient, suffer from poor communications at a full program level and usually take a considerable time period to identify and resolve the most important issues. The incremental review process promises to make such reviews considerably more efficient and effective.

### 8-3 Roles of the WBS in the Program Office

The engineering part of the program or contract WBS is an imperative for getting the engineering effort accomplished efficiently and effectively and for assigning integrated product teams. It is also useful for deriving statement of work tasks. A statement of work (SOW) tells a contractor what work the contractor must accomplish. The work breakdown structure dictionary for each element of the engineering WBS provides a task description useful for structuring requirements for section three of the SOW.

The WBS also provides an outline for:

- (1) Determining Contract Line Item Numbers (CLINs) that specify the deliverables of a contract,
- (2) Assessing and identifying technical risks,
- (3) Establishing key interface identification and control requirements,
- (4) Evaluating and managing Engineering Change Proposals (ECPs),
- (5) Determining the number and type of technical reviews and audits required.

In addition, when the appropriate program level elements (e.g., Data, Program Management, Site Activation and Spares) are added to the engineering part of the WBS to meet MIL-STD-881B requirements, a structure is provided for forming a budget and making cost estimates. The cost data, collected on the elements of the work breakdown structures specified in MIL-STD-881B, provides cost information for making independent cost estimates (ICE) and the COEA early in the program. The WBS is also used for collecting and analyzing costs for Cost Performance Reporting (CPR) and/or C/SCSC reporting.

The WBS, has historically not been widely used by the engineering community and has been primarily used by the business side of the program office for cost and budget purposes. The MIL-STD-881 has had this role as its central focus. The proposed standard revision, MIL-STD-881B, places equal emphasis on the engineering roles and cost roles, and a WBS developed to support the Systems Engineering Process is expected.

## 9-0 TECHNICAL RISK MANAGEMENT

### 9-1 Defining Risk

Acquisition risk is a measure of the potential inability to achieve program objectives, including technical performance, cost or schedule. The level of risk has two components: the probability of failing to achieve a particular outcome, and the consequences of failing to achieve that outcome. Defense system acquisition programs are often aimed at providing significant increases in capability over existing systems, making risk inherent to many aspects of the program.

The technical risks can encompass any of the technical disciplines/specialties that impact the ability to achieve any of the eight primary functions of the system (development, manufacturing, verification, deployment, operations, support, services, training, & disposal). Requirements definition, operating environment, material properties, system complexity, maturity of technology, software, support, reliability & maintainability, inter operability, and interfaces are among the sources of technical risk to a program. Failure to appropriately address such areas can have an adverse effect on the performance, cost, or schedule objectives of the program.

In general, high system complexity (hardware and/or software) or technical immaturity tends towards increasing probability of failure to meet objectives. This probability of failure, coupled with the ensuing consequences (performance, cost, or schedule), will provide an indication of the overall level of program technical risk.

- (1) Low risk is characterized by mature technology, demonstrated capabilities, and few changes required to develop system. There is little potential to impact cost, schedule, or performance objectives and normal contractor effort and government monitoring generally will be all that is required.
- (2) Moderate risk is characterized by breadboard/brassboard demonstrations, capabilities not fully demonstrated, and design iterations required to achieve objectives. Some disruption of schedule, increase in cost, or degradation in performance requiring special contractor effort and close government monitoring is associated with moderate risk items.
- (3) High risk is characterized by technology not having been demonstrated, capabilities which do not currently exist, and expectation that a significant number of design iterations will be needed to meet performance requirements. High risk also infers serious disruption of schedule, increase in cost, or degradation in performance and requires immediate attention. Risk management is part of the systems analysis and control element of the Systems Engineering process. Early risk identification and assessment allows one to plan appropriate risk mitigation measures into the program, plan for tracking (e.g., technical performance measures) and reassessment, and implement corrective actions.

## 9-2 The Technical Risk Management Process

DODI 5000.2, Part 5B formalizes the requirement for a risk management program within each acquisition program. At each milestone, the program manager must provide an assessment of risks along with plans on how each of the identified risks will be managed. To accomplish this, the program manager needs a process which may be applied within each phase of the development program. The risk management process consists of Planning, Assessment, Analysis, and Handling. Planning is needed to establish program objectives; acceptable risk levels; and identify risk management resources, responsibilities, and techniques. Assessment includes identification of the risks and their relative importance. Expert interviews, detailed review of program and technical plans (e.g. SEMP, ILSP, TEMP), and technical assessments are some of the methods used to assess risks. The work breakdown structure can be a useful tool in identifying and focusing attention on the risk areas.

Risk Analysis is the quantification of risk areas and assignment of priorities. Risk analysis is supported by various risk analysis/simulation tools. In addition, the use of experts in specific subject matters and use of the templates (DOD 4245.7-M) are effective ways to assess and analyze risk. Outputs from risk analysis include watch lists and a list of those parameters (technical performance measurements) which should be tracked during the development program. Risk Handling is taking the appropriate actions to mitigate risk. These actions generally fall into one of five categories: Avoidance, Control (probably the most prevalent), Assumption and Transfer.

The process outlined here should provide the program manager a systematic method to assess risks and define appropriate actions to assure decision makers that the program risks are at an appropriate level to proceed to the next phase of development. It should also be pointed out that risk management is not a stand-alone or add-on activity in system acquisition. It is an integral part of the program management process which includes: Systems Engineering, logistics support, manufacturing management, software development, acquisition strategy, and cost estimating. Conversely, the activities conducted within each of these areas are in essence risk management activities and techniques.

## **10-0 TRADE STUDIES AND EFFECTIVENESS ANALYSIS**

### **10-1 Trade-Off Analysis**

Trade-Off Analysis is a formal, technically based decision analysis method used for resolving conflicts and making choices among viable alternative solutions associated with:

- (1) competing requirements and constraints;
- (2) system concept functions and allocations; and
- (3) configuration management and design synthesis issues associated with systems, subsystems and configuration items and interfaces.

During the earlier stages of the program, trade-offs will primarily be made at relatively high levels of abstraction. For example, trade-offs among competing system concepts that might be employed to accomplish the mission need. During later stages of the program the trades will be made at much lower levels, for example trade-offs among competing subsystems or even trade-offs among specific components that can accomplish some specific function within a system or subsystem. In all cases these trade-offs involve decisions that consider the performance capability of the item involved in the context of the cost and effectiveness of that particular item.

### **10-2 Elements of the Trade-Off Process.**

A Trade-Off process requires a rational, objective and repeatable methodology to support decision making. A typical Trade-Off Analysis (TOA) Methodology involves:

- (1) problem definition;
- (2) review and understanding of user requirements;
- (3) selection of the TOA methodology and evaluation criteria;
- (4) identification and selection of the problem solving alternatives;
- (5) development of models and measurement of the performance of the various alternatives;
- (6) analysis of results, including selection of the preferred alternative and the conduct of a sensitivity check (sensitivity analysis); and
- (7) documentation of the process and results for future examination and use.

### 10-3 Effectiveness Analysis

Effectiveness analyses are required, in key areas, to ensure a system meets customer needs and provides a balanced set of products and processes. Effectiveness analyses evolve from the primary life cycle functions mentioned in MIL-STD-499. Effectiveness analyses may involve any of the following:

- Manufacturing
- Verification
- Deployment
- Operations
- Support
- Training
- Disposal
- Environment
- Life cycle cost

Effectiveness analyses are used to:

- (1) support identification of mission and performance objectives and requirements,
- (2) support the allocation of performance to functions,
- (3) provide criteria for the selection of solution alternatives,
- (4) provide analytic confirmation that designs satisfy customer requirements, and
- (5) support product and process verification.

Various effectiveness factors can be used to characterize the performance of a system. These include various figures-of-merit for cost-system effectiveness, operational availability (A<sub>0</sub>), logistic support effectiveness, mean time between maintenance actions (MTBM), mean time between failures (MTBF), mean time to repair (MTTR), facility use (in percent), required maintenance skill levels and tasks, facility use (in %) and other like factors. The particular set of effectiveness factors that are selected to characterize a system depends on the critical needs for the operation and support of that particular system. Note that the effectiveness factors listed above fit into various system functional categories listed below and are identified by conducting effectiveness analyses against the primary system functions. System functional effectiveness can usually relate to various operational requirements originally established through examining various mission scenarios. Many of the effectiveness factors selected are candidate technical performance measures to be tracked during the development program to assess progress.

## 11-0 TECHNICAL PERFORMANCE MEASUREMENT

### 11-1 General

The need for performance measurement is established by DODI 5000.2 and MIL-STD-499. Technical Performance Measurement (TPM) is one of the techniques for conducting systems analysis and establishing technical management control, which are among the major activities that comprise the Systems Engineering process. TPM is complementary to the C/SCSC system for determining the technical status of development. If measurements are properly scheduled, TPM provides the Program Manager with the technical status of critical technical parameters in a timely manner so that corrective actions can be taken.

TPM evaluation is usually conducted during a design review. TPM evaluation may be accomplished at other milestones such as significant test events or as directed by the Program Manager.

TPM is concerned with evaluating the adequacy of a Configuration Item design to satisfy the system performance requirement specification. As such, the parameters selected for technical performance measurement must (1) be important indicators of the eventual technical success of the entire system, (2) be measurable, and (3) have an associated performance profile indicating anticipated future performance over the time period of interest.

In essence, TPM is employed to identify and flag a Configuration Item (CI) design deficiency that might seriously endanger meeting a critical system performance requirement. If a technical performance deficiency is flagged, an analysis must be performed to determine the cause and to assess the impact on higher level parameters. Alternate recovery plans must then be developed and cost, schedule and performance impacts must be fully explored. If performance is in excess of requirements, opportunities for reallocation of requirements and resources are assessed.

Technical performance measurement is fundamentally rooted in the concept that management control is established by selecting key parameters to be tracked, estimating the future behavior of those parameters over a period of time, then tracking actual performance against estimated values and using the degree of variation from estimated values to identify the need for management attention. In this sense TPM is no different than most of the other performance based management techniques familiar to DoD managers.

### 11-2 Technical Performance Measurement (TPM) Profile

A TPM Profile is a plot that depicts the performance of a specific TPM parameter throughout the program.

Relevant terms and relationships used in a TPM profile are described below:

**Achievement to Date.** Measured progress or estimate of progress plotted and compared with planned progress at designated milestone dates.

**Current Estimate.** The value of a technical parameter that is predicted to be achieved with existing resources by the end of the contract.

**Milestone.** Point in time when a TPM evaluation is accomplished. Evaluations are typically scheduled to support technical reviews, significant test events, and may also support cost reporting intervals.

**Planned Value.** Predicted value at the time of measurement based on the planned profile.

**Planned Profile.** Profile representing the projected performance of the selected parameter.

**Tolerance Band.** Management alert limits placed on either side of the planned profile indicating the degree of variation allowed. Represents projected estimating error.

**Threshold.** The limiting acceptable value of a technical parameter, usually a contractual requirement.

**Variation.** Difference between the planned value of the selected parameter and the achievement-to-date derived from analysis, test, or demonstration.

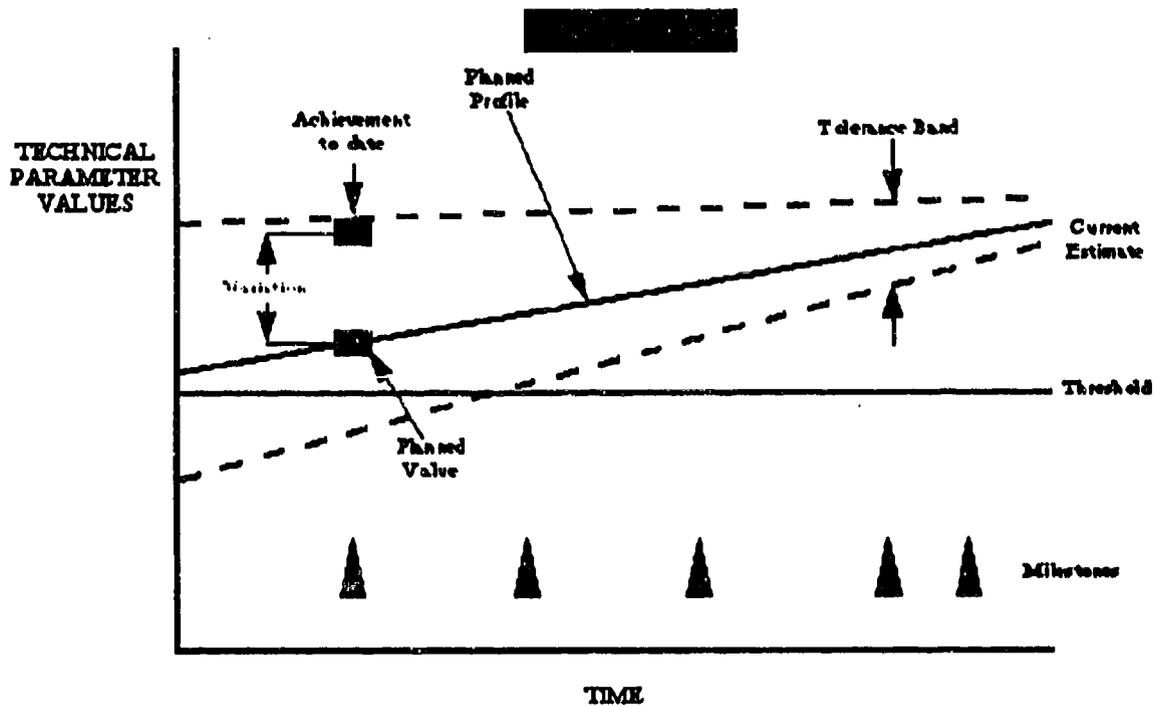


Figure 11-2 Typical Technical Performance Measurement Profile

Figure 11-2 presents a typical performance profile illustrating the use of the terms defined above. TPMs comprise an important element of the Risk Management process. Only key parameters are selected as TPMs to be tracked throughout the program. These are

usually broad and critical parameters that provide insight into important aspects of the health of the program. Candidate TPMs include overall weight of the aircraft or the avionics suite (always a critical parameter), reliability expressed as mean time between failures (MTBF), or repairability expressed as mean time to repair (MTTR). To explain the use of the TPM, consider Figure 6. representing for example the MTBF of the avionics suite. The Threshold value shown is the specified minimum acceptable value of MTBF for the avionics suite established by the contract. The area above the Threshold is the favorable region and that below the Threshold is designated as the unfavorable region. At the start of the program the estimate for MTBF is shown to be in the favorable region just above the Threshold. The Current Estimate is the estimate of the system MTBF to be achieved at the conclusion of the program based on the current rate of progress. The Planned Profile is shown as a linear plot from the initial value to the Current Estimate. The Profile may have any shape depending on the parameter and actions taken during a particular program phase that affect the TPM in question. A Tolerance Band is established on both sides of the Planned Profile to establish a practical tolerance level for the TPM. Achievement to date is a current measure of the TPM usually taken against a program milestone (such as a technical review or audit) or as part of a periodic reporting process. Variation is the difference between the planned value for the TPM and the actual value (achievement to date). TPMs are used as part of a management by exception technique which is invoked when a TPM value falls outside the Tolerance Band. When the value falls in the unfavorable region, the risk is assessed and previous contingency plans are acted upon (or a contingency plan is prepared if none previously existed). If the value falls outside the profile on the favorable side it may be possible to reallocate program resources from this TPM to others that reflect an unfavorable prognosis. Used in this way the TPM process becomes an important part of program Risk Management.

## 12-0 TECHNICAL REVIEWS AND AUDITS

### 12-1 General

Technical reviews are an integral and essential part of the acquisition process. Reviews range from very formal technical reviews by government and contractor engineers to informal reviews concerned with product and task element of the work breakdown structure (WBS) that involve only a few directly concerned personnel.

MIL-STD-499B advocates a revised approach to technical reviews and audits. The use of incremental reviews is proposed to provide increased efficiency to the entire review process. The proposed approach depends heavily on the development and utilization of an Engineering WBS that is structured to efficiently support the Systems Engineering Process. The WBS is then used as a basis for the formation of integrated product teams for each of the elements of the WBS. Teams are assigned to each of the levels of the WBS that are appropriate. The IPT approach supports the use of incremental technical reviews and audits. Much of the review activity at the lower level WBS elements is accomplished through team reviews and regular meetings. Major reviews can be managed by having a series of incremental reviews at subsystem and functional levels. As reviews at the lowest reviewable level are completed, the incremental reviews can proceed to higher and higher levels in the Engineering WBS. When the major reviews such as the PDR and the CDR are conducted, they consist largely of demonstrations of compliance reflecting the earlier reviews at lower levels. In this manner it hoped to avoid the large scale reviews witnessed in major system programs when hordes of reviewers descend upon a contractor to conduct simultaneous reviews of all key program elements usually breaking into "splinter groups of specialists reviewing each significant program element. Reviews of this type are not usually very efficient, suffer from poor communications at a full program level and usually take a considerable time period to identify and resolve the most important issues. The incremental review process promises to make such reviews considerably more efficient and effective.

### 12-2 Reviews and Audits Under MIL-STD-499B

MIL-STD-499B includes a description of the Technical Review and Audit procedures to be followed in support of the System Engineering Process. Appendix C of Draft MIL-STD-499B (5/6/92) is entitled *General Guidance On The Conduct of Technical Reviews*. In this appendix, MIL-STD-1521 is referenced but complete guidance for reviews to be utilized and the criteria for each are provided.

#### Reviews under MIL-STD-499B:

- (1) Alternate System Review (ASR)
- (2) System Requirements Review (SRR)
- (3) System Functional Review (SFR)
- (4) Preliminary Design Review (PDR)
- (5) Critical Design Review (CDR)
- (5) System Verification Review (SVR)
- (6) Functional Configuration Audits (FCA)
- (7) Physical Configuration Audits (PCA)

Criteria is provided for each of the review types listing the System Engineering Master Schedule entry accomplishment appropriate for the review and its specific acquisition phase. Also listed are the exit accomplishments that are necessary for the successful completion of the review or audit. Emphasis on the Incremental review process is added by describing additional review types that support it. These reviews include Subsystem Reviews, Functional Reviews and Interim System Reviews. The process encourages extensive use of IPTs mapped into the elements of the Engineering WBS as a means of utilizing incremental reviews to improve the performance of the entire Technical Reviews and Audits process.

## 13-0 CONFIGURATION MANAGEMENT

### 13-1 The Central Role of Configuration Management

A primary output of the Systems Engineering process is a technical data package containing engineering drawings, associated lists, process descriptions, specifications and interface documents which together define the physical geometry, material composition, performance characteristics, manufacture, assembly, and acceptance test procedures. Configuration management includes the function of identification of products and processes to be controlled (through specifications, drawings, etc.), change control (ECPs, etc.), status accounting, and audits (verification).

Interface management is an integral part of configuration management. Interface management includes internal and external interface definition, interface control, compatibility assessment, and interface coordination through an Interface Working Group (IWG) or an Interface Control Working Group (ICWG). Effective Interface management is a major factor in the effective integration of large scale systems.

Another aspect of configuration management is Technical data management. Technical data management administers the development, control, release, and delivery of required technical data.

### 13-2 The Use of Configuration Baselines

A configuration identification document or a set of such documents formally designated by the Government at a specific time during the life cycle of a Configuration Item (CI) is called a configuration baseline. Baselines, plus approved changes, constitute the current approved configuration documentation. Configuration baselines are used to ensure an orderly and controlled transition from one major commitment point (milestone decision) to the next. For configuration management purposes there are three baselines, which are established sequentially:

**Functional Baseline:** The initially approved documentation describing the functional characteristics of a system or CI and the verification means required to demonstrate the achievement of those characteristics. It is initially drafted near the end of the Concept Exploration & Demonstration (CED) phase with the System/Segment Specification (Type A) and established by Government approval and contract implementation early in the Dem & Val phase, but no later than the System Design Review (SDR). This baseline is the foundation for configuration management by the Government during subsequent phases of the program.

**Allocated Baseline:** The initially approved documentation describing a CI's functional and interface characteristics that are allocated from those of a higher level CI, with verification required to demonstrate the achievement of the specified functional and interface characteristics. This baseline is documented in Type B Specifications and related documents. For hardware configuration items (HWCI), the timing of the establishment of this baseline will be agreed upon between the contractor and the contracting agency and will take place during the Engineering & Manufacturing Development (EMD) phase. There are significant variations on policy regarding the timing of the establishment of the allocated baseline among services. For computer software configuration items (CSCI), the allocated

baseline is typically established upon completion of the Software Specification Review (SSR), although timing may again vary by service. This baseline is used to govern the development of the selected CI's that are allocated from system requirements, or that are a part of a higher level CI.

**Product Baseline:** The initially approved documentation describing all the necessary functional and physical characteristics of the CI, any required joint/combined operations inter operability characteristics, and the selected characteristics designated for production acceptance testing. This baseline, documented by a Type C Specification with accompanying Type D (Process) and Type E (Material) Specifications plus engineering drawings and detailed design documentation, evolves for each CI from the corresponding Type B Specification. The Product Baseline is established by the Government upon completion of the Physical Configuration Audit (PCA). The product configuration documentation is used to prescribe the necessary "build to" or "form, fit, and function" requirements and the acceptance tests for those requirements. The degree of detail required is dependent upon anticipated methods of extended procurement and for logistics support of potentially reparable components of the item.

### **13-3 The Roles of the Government and the Contractor in Configuration Management.**

In the Statement of Work, the Government tasks the contractor to perform the following four functions of configuration management:

- (1) Identify and document the functional and physical characteristics of the configuration items.
- (2) Control changes to configuration items and their related documentation.
- (3) Record and report information needed to manage configurations effectively, including the status of proposed changes and the implementation status of approved changes.
- (4) Audit configuration items to verify conformance to specifications, drawings, interface control documents, and other contract requirements.

The Government performs this tasking by selectively tailoring the specific requirements of MIL-STD-973. While most configuration management tasks are actually performed by the prime contractor, many of the tasks involve making recommendations which the Government may or may not accept, e.g., designation of CI's, configuration changes, etc. Some other tasks are performed jointly by the prime contractor and the Government, e.g., audits and assignment of document numbers and nomenclature.

## **14-0 SOFTWARE SYSTEMS ENGINEERING**

Software engineering must be integrated into the overall Systems Engineering effort and not be considered separately. See Volume 1, section 9, for additional details.

## **15-0 SUPPORTABILITY ENGINEERING UNDER THE SE PROCESS**

### **15-1 General**

Supportability engineering is an important, multi-faceted engineering discipline that must be integrated under the Systems Engineering umbrella, not considered separately. This requires that the multi-disciplinary Integrated Product Development Teams (IPDT) defined in both Systems and Concurrent Engineering must include engineering representation from the many elements that comprise supportability. Supportability management is embodied in the Integrated Logistics Support Plan (ILSP) which is usually initiated during conceptual design and is updated in preliminary design. Logistics support activities that are performed throughout the system life cycle as part of Systems Engineering is shown in Figure 15-3.

### **15-2 Support Environment**

It is important to define the support environment for the system under development. The system support environment must be compatible and consistent with fleet operations. For avionics systems this usually means compatibility with the Aircraft Carrier environment and Carrier operations. In general, Carriers have little excess space for storage of spare parts. The sparing concept (as an example) must be a major consideration in many of the design decisions made. Because of the Carrier environment, configuration management is of major importance. With the many aircraft types that must be supported aboard an Aircraft Carrier, many avionics systems configurations for each type can severely exacerbate the problem leading to a logistics nightmare.

Supportability concerns must consider the levels of maintenance provided for the avionics system. Modern avionics design has attempted to move towards a two level maintenance concept eliminating the Intermediate or "I" level. In the Joint Integrated Avionics Working Group (JIAWG) standardization efforts, two level maintenance has been the goal with strong Air Force support. Two level maintenance is difficult to implement fully on an Aircraft Carrier with limited availability of storage space for spare parts.

### **15-3 Integrate Support Concepts into Systems Engineering Guidelines**

Ensure that the systems engineering process defined is consistent with supportability needs. Recognize that the logistics Supportability Analysis (LSA) procedure is part of the SE process and that the entire Integrated Logistics Support (ILS) approach is in reality part of (or at least closely coupled to) the SE process.

Require that supportability/logistics engineers and all supportability specialties be represented as part of the multi-disciplinary integrated product teams recommended under systems engineering guidelines.

Figure 15-3 lists the logistic support activities that must be considered in the system life cycle as part of the Systems Engineering process.

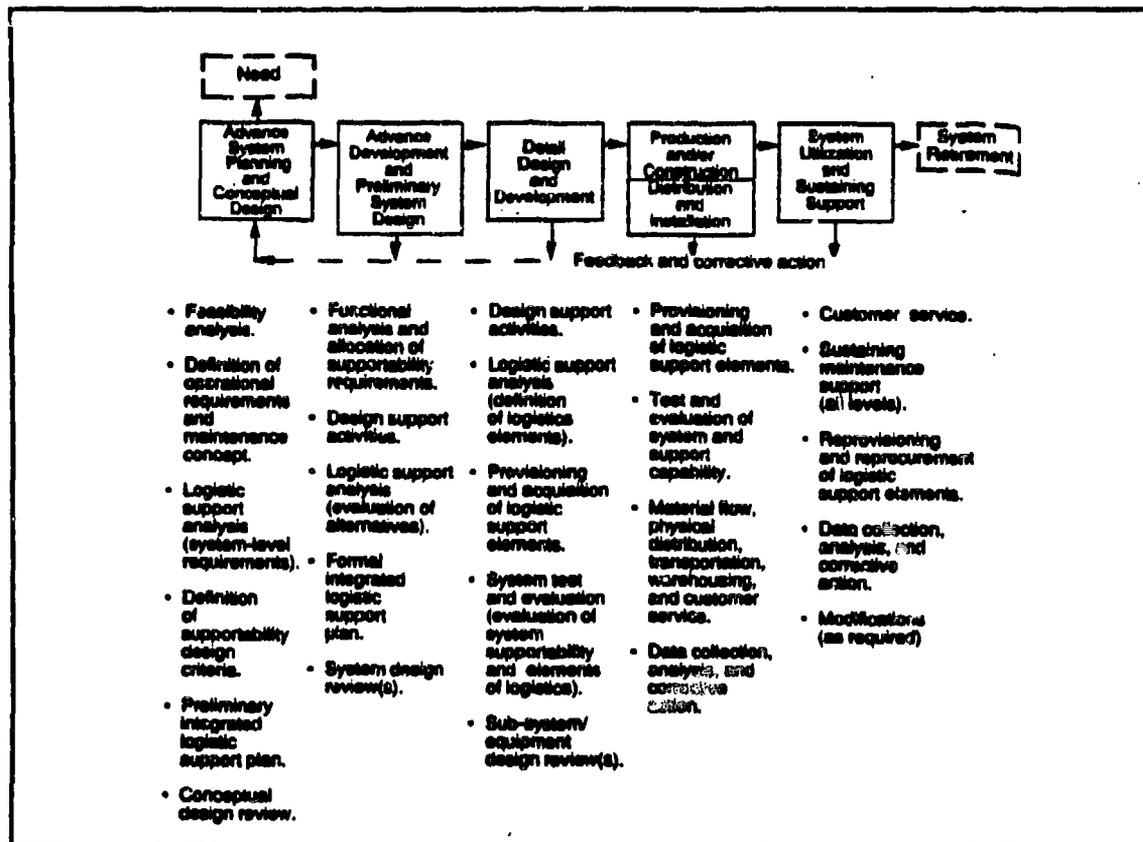


Figure 15-3 Logistic support activities in the system life cycle.  
(from, B.S. Blanchard, *System Engineering Management*, with permission)

## **16-0 EFFECTIVE USE OF COMPUTER BASED SYSTEMS ENGINEERING TOOLS**

It is recognized that large scale (many complex and interleaved requirements) systems employing the latest generations of advanced electronic/electro-optic technologies require the use of modern computer tools in order to achieve efficient effective designs and to intelligently choose among the myriad design choices that are available. The avionics suite for a modern tactical aircraft employing advanced integrated avionics technologies is a prime example of this class of large scale systems.

Tools are used in each aspect of the acquisition process from simulation programs used in the conceptual phases of development to make broad stroke weapon systems decisions, to the life cycle cost models used in later phases to compare the overall costs through the expected service life of the system and evaluate the projected cost effectiveness of various alternate supportability/logistic concepts.

One of the most significant advances in the application of Systems Engineering is the increasing availability and use of computer based tools for all phases of the process. Analysis and design tools for Systems Engineering are available for elements of all of the acquisition phases. Many of the tools were originally developed to support the Software development task as Computer Aided Software Engineering (CASE) tools, and have been extended to Systems Engineering. A need to establish a complete, well integrated systems engineering tool set is very important to the effectiveness and efficiency of the process. (see also Volume 1, Section 9-3.5).

## **17-0 CONCURRENT ENGINEERING**

Concurrent Engineering is closely related to Systems Engineering in the sense that both disciplines require that all elements of the product life cycle be considered simultaneously. Systems Engineering combined with Engineering Management and Automated Tools makes up Concurrent Engineering (CE). The SEP embodies both SE activities/tasks and Engineering Management to control the development effort. Integrated Product Teams (IPTs) also enable the CE effort.

## 18-0 SYSTEMS ENGINEERING FOR MODULAR AVIONICS

### 18-1 General

The concept of Modular Avionics packaging has in effect created the basis for a de facto architecture. Although a module based system isn't in and of itself an architecture, the choice of packaging concept provides a certain physical definition to the architecture. At present, the evolution of Standard Electronic Module (SEM) packaging programs has brought us to the era of the SEM "E" module profile; a module that is approximately 6 inches by 6 inches by 0.6 inch thick.

*The size of a module is usually a compromise based on the current state of technology. Choice of a module that is too small, allows for implementation of low level functionality for each module providing a great deal of functional commonality at the module level but require a great number of modules and a large number of external connections. Choice of a module size too large for the technology will allow a great deal of functionality per module but will provide little commonality, e. g. each module will tend to be unique. A large module size will tend to minimize external connections at the expense of commonality.*

Modular Avionics provides significant impacts to Systems Engineering. As a design approach, Modular Avionics defines a tailoring to the Systems Engineering process. Modules imply and enforce a system partitioning at the module level. With significant experience with a particular module format (size and configuration) comes an applications data base that can provide an enhanced degree of confidence in design predictability. For example, reliability of an electronics module can be related to power dissipation (total thermal loading), thermal and mechanical environment and other operational conditions. For a given power dissipation per module and cooling technique employed, data can be gathered from operational experience and used to predict very consistently such factors as reliability, total avionics weight, and backplane data bus performance when the same module/bus configurations are used for new designs. An established Modular Avionics architecture approach makes future applications more effective and predictable without severely limiting technological enhancement (at the module level). Avionics modular architectural activities are described below.

### 18-2 JIAWG Modular Avionics

JIAWG Avionics provides for a major transition of Avionics systems from a Federated "Black Box" architectural approach to a module based integrated architecture. Figure 18-2 provides a pictorial depiction of this transition. The architectural approach is based on a great investment of avionics focused R&D funds by both the Navy and Air Force over several decades.

Military use of the module concept was first developed and demonstrated by the Navy in the Poseidon Program and transitioned into various Standard Electronic Module (SEM) and Standard Avionics Module (SAM) implementations through the years. Research performed under the Air Force "PAVE PILLAR" Program and the Navy Advanced Avionics Technology Demonstration (AATD) Program, established the basis for the integrated avionics systems and the fiber optics bus developments employed in the JIAWG aircraft.

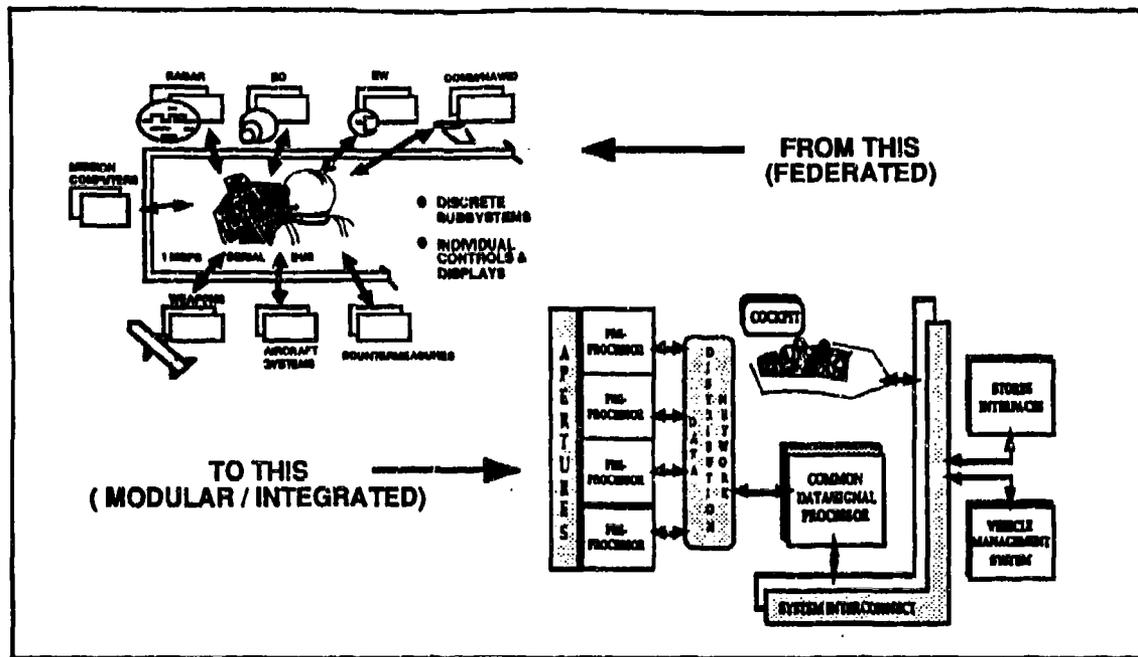


Figure 18-2 Architectural Transition from Federated to Modular under JIAWG

The SEM "E" profile has been adopted for the bulk of the "common avionics" digital processing performed on JIAWG aircraft (F-22 and RAH-66 currently). Recent advances in microcircuit densities (sub-micron feature size) has resulted in power densities of 80-90 watts for certain F-22 modules. Since forced air cooling can accommodate only about 40 watts per module reliably in the SEM "E" size, F-22 has been forced to adopt liquid flow through cooling techniques for higher power modules. Technology advances towards microcircuits with 0.3 micron feature sizes and lower will continue to drive thermal management for advanced avionics to liquid flow through cooling. RAH-66 has managed to retain air cooling by limiting circuit density on SEM "E" modules to control power dissipation. It appears certain that both conventional air and conventional liquid cooled modular systems will be used in aircraft. Two SEM "E" module variations will undoubtedly become standard in future generation aircraft. The first of these types is a conventional SEM "E" in which cooling is by conduction to a thermal manifold (rail) cooled by air or liquid flow through the rail structure, limited to approximately 40 watts maximum per module. The second type provides liquid flow through the modules in which the cooling liquid (or air) is forced to flow through passages in the central module core and providing cooling for modules dissipating upwards of 100 watts. Both techniques can be used where applicable to satisfy a wide diversity of avionics requirements.

### 18-3 Integrated Modular Avionics

Hailed as the most important innovation in airline avionics in 20 years is the development of a standard modular avionics concept. Designated as Integrated Modular Avionics (IMA), the concept introduces a standard avionics cabinet enclosure with the use of Line Replaceable Modules (LRMs). This integrated approach has the cabinet replacing many separate black boxes and enables the integrated avionics to share processing, memory, I/O functions and power supply generation. This concept will be deployed on a full scale basis in the future Boeing 777. Many of the concepts were originated in the military and have been deployed on a limited scale in military aircraft for many years. The IMA concept is unique in that all of the standards that support the concept are being developed together in the same relative time phasing so that a Systems Engineering approach can be taken to the overall standardization effort. The standardization process is managed by the Airlines Electronic Engineering Committee (AEEC) as a replacement for the current generation ARINC 700 avionics. Development of the IMA is a further move towards digital avionics. Employing a high degree of software control, the IMA will employ latest versions of high performance microprocessors, and advanced data bus technology. The overall architecture is defined in ARINC Report 651. Other reports that comprise the standards include, ARINC 629 data bus, ARINC 659 backplane bus and the ARINC 429 databus. The airline industry has fully adopted Ada as their standard Higher Order Language because of the rigor it provides the software development process. The AEEC standards organizations are participants in the Ada 9X revision process and look forward to the release of the revised standard. Other standards that comprise the IMA architecture include:

- ARINC 609, Design Guidance for Aircraft Electrical Power System;
- ARINC 613, Guidance for Using the Ada Programming Language in Avionics Systems;
- ARINC 652, Guidance for Avionics Software Management;
- ARINC 653, Standard Application Software Environment.

ARINC 653, is of particular interest since it is an attempt to describe the functionality of the IMA Operating System, as well as describe the interface between the Operating System and each of the applications that use the Operating System services. Many of these standards are still under development with full standards development scheduled for completion in the 1995-1996 time frame.

## 19-0 TQM AND THE SYSTEMS ENGINEERING PROCESS

Total Quality Management (TQM) initiatives and other ongoing initiatives for process improvement should be incorporated with the Systems Engineering Process. Because of ongoing emphasis throughout the federal government on productivity enhancement, there has been a "general conditioning" to focus on quality, productivity enhancement and effective teaming of all participants. Emphasis on the needs of the customer, and the total involvement of all employees as "stakeholders" in a team process are TQM principles that are very much in concert with modern Systems Engineering practice. Viewing Systems Engineering as the technical element in an overall TQM process should ease the acceptance of systems engineering as part of an overall process improvement initiative. Because industry which serves the military avionics market has been exposed to TQM principles; invoking Systems Engineering concepts through TQM programs should be equally well supported by the commercial industry that serves the military avionics market. One contractor (Harris Corporation) has used the TQM process with employee participation to develop and achieve consensus for a Systems Engineering Process to be employed throughout the corporation.

## 20-0 PRODUCT IMPROVEMENT

### 20-1 General

Systems Engineering must specifically address needs of product improvement developments. Product improvement efforts include retrofits, upgrades and Pre planned Product Improvement (P<sup>3</sup>I) programs. The discussion below addresses the application of Systems Engineering in product improvement.

### 20-2 The Role of Systems Engineering in Product Improvement Programs

Systems Engineering plays a major role in product improvements, both during development and during post-production operations and support phases.

There are several strategies by which a system may be improved during its life cycle; all require a disciplined application of the Systems Engineering process if the improvement is to be made in the most effective and efficient manner. A need for future improvements may be foreseen early in the initial stages of the

program, and in such cases the acquisition strategy should be formulated to include product improvement as an integral aspect of the system acquisition. Product improvements of this type typically fall into one of the following categories:

**Pre-Planned Product Improvement.** A relatively low risk development is planned based upon the technology available at the time, with the understanding that more advanced technology will be integrated into the system when it becomes available during the latter development stages.

**Evolutionary Acquisition.** Often employed in C<sup>3</sup>I systems and other highly software intensive systems where requirements are impossible to define with certainty in advance of user employment in the fielded environment. A core capability is initially fielded, then improvements are added as they become feasible, (often through block upgrades) and operational requirements become more completely understood. This type of acquisition has application to advanced avionics systems as well. Core Avionics capability is necessary with the first operational deployment. Improvements in Mission Avionics can evolve over time as, for example, Improved algorithms for Sensor Fusion are developed and applied. In this example the bulk of the improvements are implemented in software, requiring little change to the physical configuration (hardware) of the avionics suite.

Changes and improvements are also made to systems as they progress through the normal stages of development prior to fielding the system. When these changes have not been explicitly anticipated and planned for in the acquisition strategy as discussed above, they generally take the form of Engineering Change Proposals (ECPs) and block changes

After the system has been deployed, product improvement typically takes the form of Modifications and Block Changes. The decision authority overseeing the effort may elect to have the system being modified enter into either Phase II or III of the Systems Acquisition Process.

Regardless of the type of product improvement undertaken, the role of Systems Engineering remains the same as during the normal product development cycle. That is, given a set of input states, conditions, and constraints, all activities of the Systems Engineering process (Requirements Analysis, Functional Analysis and Allocation, followed by Synthesis, managed through Systems Analysis and Control) are employed to achieve a balanced system as the eventual output. The process is an orderly problem solving approach applied iteratively and recursively, with the level of detail involved dependent upon the level of development undertaken. If anything, it is even more important that a well developed Systems Engineering approach be taken to post-production modification, because many of the problems typical of modifications to deployed systems are often more complex than the original development itself.

### 20-3 Avionics Experience in Product Improvement Programs.

#### General

Upgrades offer significantly different problems than do so called "clean sheet" new designs. It is recommended that any SE Avionics revisions based on the findings of this review make provision for upgrades (generic and as specific as can be projected) as part of the initial design process. Often specific retrofits have not been considered as part of the original avionics development. As a result, "interface" provisions have not been put into place and have to be added. This is very costly and must be avoided when possible.

#### 20-2.1 MIL-STD-1553 (Standard Avionics Data Bus)

Special requirements exist for aircraft developed prior to the almost universal use of MIL-STD-1553 A/B as the standard data bus. These aircraft employ no standard data transfer interface and suffer from a proliferation of arbitrary single purpose point-design interfaces. Retrofit to such aircraft may be made more difficult because of the lack of a standard digital data bus. Adoption of MIL-STD-1553 as the standard military avionics data bus made a remarkable change in avionics integration. This bus provides a standardized control interface and is an effective integration tool for avionics systems. Many aircraft developed prior to the use of "1553" have later incorporated this bus in painfully expensive stages (The F-14 aircraft is one such example).

#### 20-2.2 Standard High Speed Data Bus

With MIL-STD-1553B as an example, the military focused on the development of a higher performance data bus commensurate with the data flow needs of newer generations of signal and data processors. Work in the SAE Avionics Systems Division (ASD), the organization that prepared and reviews MIL-STD-1553B, initiated development of a next generation high performance data bus as a follow-on to MIL-STD-1553B. This effort brought the aerospace industry together with military engineers (from the three Services) to develop a new standard. This effort started with a requirements (military avionics) definition phase and evaluated many data formats, bus acquisition techniques and protocol types. The result was the preparation of two standards, both operating at 50-100 Mb/s rates (compared to a 1 Mb/s rate for MIL-STD-1553B) and both employing a "Token Passing" protocol but different distribution topologies. The SAE issued two high speed data bus (HSDB) standards, one for a "linear" topology and the other for a "ring" topology. The JIAWG (F-22, RAH-66 and A/FX) aircraft both introduce new high performance data buses based on the SAE Linear HSDB standard while retaining MIL-STD-1553B for basic command/control functions and low data rate information transfer. It is unlikely that any new aircraft will ever be developed without use of MIL-STD-1553B and/or a combination

of standard networks and buses that provide a flexible interface for signal/data connectivity. Future retrofits will take advantage of the standard networks/bus infrastructure now being established to ease engineering and integration costs.

### 20-2.3 Modular Avionics

New retrofits of federated "black box" avionics systems are looking toward the use of advanced modular packaging. The Air Force Modular Avionics System Architecture (MASA) program has investigated this type of upgrade and has established certain guidelines. Any avionics tailoring of MIL-STD-499B should include guidelines for these generic upgrade cases. The JIAWG aircraft in development and the Navy SHARP program also make a great deal of modular packaging R&D available for avionics applications. Standardized modular avionics design can ease the costs of retrofit. In future aircraft employing a standard modular avionics packaging scheme with standard connector pin-outs and standard backplane buses, retrofit electronics may consist of adding a module (or a few modules) in an existing "integrated rack or equipment enclosure. Where aircraft apertures (usually customized to the aircraft physical configuration) are involved in the retrofit the design can't be completely standardized, but overall costs should be better predicted and controlled.

### 20-2.4 Global Positioning System Retrofits

One of the most significant recent experiences in product improvement is concerned with retrofit of the Global Positioning System (GPS) into several Navy platforms. Costs of the engineering design development necessary for the retrofit in each of the selected aircraft were (in each instance) larger than the total procurement costs of the equipment installed.

Although, the "Up Front" engineering costs were derived from complex and diverse factors, in large measure they can be attributed to a lack of consistency in the interface standards used among the aircraft. This lack of standard interfaces forced each retrofit installation into a unique "Point Design" with no further use. Lack of consistent interfaces were apparent in both hardware and software and much of the incurred costs were attributed to new Software development.

## 20-3 Recommendations for Product Improvement Systems Engineering

Based on the findings of this Technology Review, avionics product improvement efforts for future systems will be greatly enhanced by instituting the following System Engineering process procedures:

- (1) Adopt a family of avionics architecture standard elements and appropriate interface standards. Integrate these standards into the Avionics Systems Engineering process as preferred standards. (Applies to both Hardware and Software).

*The use of common avionics elements and interfaces will simplify retrofits into multiple platforms since the interfaces will be the same for each.*

- (2) When possible, adopt Open System standards and support the maintenance of the standards by full government participation in the Standards Groups that develop and maintain them.

*Consistent with the use of a common set of interface standards, this will ensure that the standards satisfy military avionics needs and are supported by a broad industry base.*

- (3) Support the concepts of modularity for both Hardware and Software. Reliance on Modular Avionics provides a Systems Engineering basis that allows for technology insertion at the Line Replaceable Module (LRM) level that mitigates technology obsolescence and provides for affordable module level system upgrading.

*Software modularity promotes the capability for software reusability, that in turn will help to control the inordinate software costs associated with system retrofits and upgrades.*

- (4) Emphasize the need to plan for retrofits and upgrades as part of the Program Managers Systems Engineering responsibilities. This planning should be instituted in very early phases of the Avionics system conceptual development to integrate the concept of upgrades and P<sup>3</sup>I initiatives into the early Functional Architecture definition.

*System Upgradeability should routinely be considered as part of the evaluation criteria when comparing alternate architectural system solutions. In the present period of fewer platforms, each performing multiple roles, planning for product improvements and future reconfiguration must become embedded in the Systems Engineering process.*

## 21-0 SYSTEMS ENGINEERING GUIDANCE

### 21-1 General

Because of the complexity of the Systems Engineering efforts described above, it is necessary that Systems Engineering application guidance be provided. Whether in the form of application guidelines or a full fledged user's handbook, the guidance should not duplicate guidance available from other sources such as MIL-STD-499B but should interpret these documents for avionics and extend the guidance where necessary.

Effective guidance for avionics must recognize the severe operational environment for high performance military aircraft and the added difficulties imposed by designing for and qualifying to this environment. The following sections provide specifics as to the guidance to be provided to the Program Manager and his staff.

### 21-2 Custom tailoring of MIL-STD-499B for Avionics/Aircraft Development

As the defining document for the application of systems engineering to the development of military systems, MIL-STD-499B provides a comprehensive guide to the systems engineering process applied to the development of defense systems. As stated in MIL-STD-499B, it is important that the document be tailored to accommodate the specific nature and system peculiar needs for each separate product class. A tailoring is necessary because of specific needs and constraints necessary to satisfy the operating environments and restrictive supportability requirements imposed by carrier operation. Examples of the special requirements imposed upon Naval aircraft include the following:

- Severe vibration, thermal and thermal shock (rapid thermal cycling), characteristic of avionics environments on high performance aircraft.
- Stringent Electro-Magnetic Interference (EMI) shielding needs imposed by proximity to high radiation fields aboard aircraft carriers (shipboard Radar, etc.).
- A need for "Real Time" processing that is much more stringent (faster processing times) than the "Real Time" needs for commercial applications. This is especially demanding on software performance.
- The need for multi-level security in an avionics suite predicated on shared buses and networks.
- Severely restricted and oddly configured areas/bays available to install avionics equipment.
- Supportability within an aircraft carrier environment which imposes constraints on configuration management (minimizing the number of different configurations), requires minimizing spares requirements and requires conformance to established maintenance procedures.

These (and other additional) factors can be dealt with, but do require special consideration as part of a customized Systems Engineering Process for Naval aircraft.

Tailoring of MIL-STD-499B for avionics will provide more direct guidance to the avionics acquisition/development process in terms of relevance to systems peculiar aspects of avionics. Administered by the NAVAIR Avionics division for all Naval Avionics development programs, such a tailored document can provide for higher level standardization without unduly limiting the capability of the individual program manager to develop for high performance. Because of the wide diversity of Naval avionics aircraft platforms, additional guidance beyond the tailored standard is recommended. Additional guidance should be provided in two forms. The first recommendation is that a series of templates be prepared for each of the classes of Navy aircraft or aircraft missions. For example, one template could be prepared for advanced high performance tactical aircraft with complex missions, and a major requirement for high speed processing and interconnection networks. A separate template could be prepared for each additional class of aircraft or each functional set of mission requirements to provide specific guidance.

## 21-3 Customizing SE for Advanced Integrated Avionics

### 21-3.1 Accommodate Open Systems Trends

Avionics application of systems engineering appropriate for the present day must accommodate trends towards the use of Open Systems Architectures (OSA). Effective accommodation of the OSA approach requires that the Navy must become knowledgeable on all the various systems building blocks that constitute advanced integrated avionics architectures. These building block elements consisting of busses/networks, processors of various types, memory devices, etc. usually represent state-of-the-art technologies that are continually changing. It is important that Navy technologists are brought into the avionics development process in the earliest phases of system development so that advanced technologies can be effectively evaluated for maturity, value added and realistic risk assessment.

Important features of the OSA concept must be accommodated by the Systems Engineering process employed. When the choice is available, emphasis should be placed on the use of non-proprietary interface standards or at least attempt to make the standards chosen available as "open" or non-proprietary standards. The use of COTS should be accommodated so that leverage of the commercial market place can be achieved when feasible. Agreement reached during the review was to use the SE process to consider COTS standards and products along with products designed specifically for the military market in a strictly objective frame of reference. This approach allows the use of COTS to be considered on a case by case basis to achieve the best "value" for the system development. To use this approach effectively, guidelines for the evaluation of COTS must be established to assure that the cost and development time to add special features to COTS to sustain performance in a more severe environment (technically this is now called militarized COTS), and the cost of adding additional configuration data (if necessary), etc., are traded off against cost advantages afforded by the COTS.

### 21-3.2 Linking of COTS and Open Systems Architecture.

COTS and Open Systems are often thought of together. It is not necessarily true that COTS products are based on Open System standards. Many of the available COTS products are as often point designs as are military products. In selecting COTS for a pervasive role, it is important that the necessary interface standards are "open", non-proprietary and well defined/disclosed. Certainly the most effective way to ensure effective use of COTS standards and products by the military is for joint standardization, and full

participation (by commercial industry and the military) in the development and review of the underlying standards.

### **21-3.3 Modular Avionics for Growth and Ability to Upgrade**

The architectures chosen for advanced high performance integrated avionics systems must be well defined and suitable for the growth and evolution necessary to match the growth needs required of increasingly complex upgradable systems for advanced avionics. The Systems Engineering process employed must require that provision for growth and ability to upgrade are considered to be fundamental parameters. Sizing of processors, memory and busses/networks is a key to this planning. The avionics systems envisioned place considerable emphasis on the use of modular packaged avionics. Modular packaging has become the preferred approach, where practical, because it provides for effective use of high density microelectronics, employs effective cooling techniques and provides for high speed data transfer on the backplane by means of data paths and data buses. The Navy SHARP program has dealt with advanced modular packaging for both air and ship platforms. The NAVAIR directed Advanced Avionics Technology Demonstration (AATD) program has funded advanced avionics packaging research often jointly funding efforts with the SHARP program. It is recommended that close coordination be maintained between avionics systems engineering decisions and the focus of SHARP/AATD avionics packaging research so that the research is well focused and coordinated with the specific development guidelines utilized by the Program Manager.

### **21-3.4 Process Integration of Hardware and Software Development**

Of necessity, the development of software has closely focused on process. The software development process has initiated many computer aided tool developments with the result that an entire category of computer based tools are called Computer Aided Software Engineering (CASE) tools. The SE process requires that the development of Software be integrated effectively with Hardware development for efficiency. The Software Development process should be considered only a component of the entire systems engineering process. Many case tool developers now focus on the entire systems engineering process with the result that the use of CASE often is used to mean Computer Aided Systems Engineering as well. Navy tailoring of the SE process for avionics should have as a goal more effective integration of the hardware and software development processes. It is important to think of Software engineering as part of an integrated Systems Engineering process, not as a separate discipline.

### **21-3.5 Employ Multi-Disciplinary Product Teams**

*The Systems Engineering Process management approach that best meets the DODI 5000.2 requirement for organizing for efficiency and effectiveness utilizes the integrated product team, or a multi-disciplinary effort to ensure a balanced approach to development.*

Organize functionally within the Navy (specifically within NAVAIR) to encourage Multi-Disciplinary product teams for avionics acquisition. Organizing government acquisition for avionics along the lines of the technical disciplines employed in modern avionics systems provides the basis for a multi-disciplinary product team organization for avionics acquisition that parallels the teams used by industry for development. Providing training to Navy avionics acquisition personnel in team dynamics and methods will develop skills that will make the working of government IPTs effective and provide an a better understanding of the workings of industry IPTs.

Encourage contractors to use multi-disciplinary Integrated Product Teams (IPTs) for development. Establish modifications to FAR/DAR, as necessary, that allow for effective government representation to contractor IPTs when appropriate.

## 22-0 SYSTEMS ENGINEERING STANDARDIZATION

### 22-1 General

Characteristic of the revitalization of Systems Engineering is a parallel focus on definition and standardization of methodology, processes and tools used in its application. Generic Systems Engineering is practiced in both commercial and military sectors, often with different constraints but with essentially the same process elements. A certain lack of generally agreed upon definitions of Systems Engineering and the technical boundaries within which it is applied is most likely due to the fact that it has only become a generally recognized engineering discipline in recent years. With recognition comes standardization activities to formally define Systems Engineering standards and practice. In this section, some standardization activities useful to Navy avionics are discussed.

### 22-2 Defense Systems Management College

The Defense Systems Management College (DSMC) serves as the national center of excellence for defense acquisition management education, research, consulting and generation of supporting publications. The DSMC Systems Engineering Department has long led the way in defining Systems Engineering as applied to the Systems Life Cycle. DSMC initially published a "*Systems Engineering Management Guide*" in October 1983. This Guide succeeded by revisions in October 1986 and most recently December 1989 serve as the definitive DoD reference to Systems Engineering application. More recently the DSMC Systems Engineering faculty have taken the lead in the preparation of MIL-STD-499B, "*Systems Engineering*" (draft dated May 1992), the defining standard for the application of Systems Engineering to DoD Systems acquisition. The faculty of the Systems Engineering Department is very much in the forefront of modern Systems Engineering practice and can serve as a prime consultant to the Naval Air Systems Command for implementation of a rigorous Systems Engineering basis for avionics acquisition.

### 22-3 National Council of Systems Engineering

The National Council of Systems Engineering (NCOSE), was founded in recognition of the increasingly important role of good Systems Engineering practice to industrial productivity in the United States. Its founders recognized the need for a forum to exchange information regarding Systems Engineering and to establish standards for its practice. Established in late 1990, NCOSE has experienced very rapid growth with over 1500 members registered by mid 1993.

The rapid growth of NCOSE reflects an emphasis within US industry on methods to increase productivity and the role that Systems Engineering plays in productivity enhancement. NCOSE serves both the commercial and military arenas. With Dr. Jerome (Jerry) Lake of the Systems Engineering Department at DSMC serving as the first national president, NCOSE activities initially centered on the review and revision of MIL-STD-499B. With NCOSE involved in its review, the DoD is assured that the standard is consistent with generic systems engineering practice, and DoD needs for the System Engineering Process are carried over to commercial practice through the membership. Interest in NCOSE has extended to the international community, with a recent recommendation to change the name from "National" to "International" Council on Systems Engineering. There are now twelve local chapters including Boston MA, Houston TX,

Huntsville AL, Melbourne FL, San Bernardino, CA, San Francisco, CA, St. Louis MO, Whippany NJ, Seattle WA and Washington, DC., with at least eight other localities currently applying for official chapter status.

Corporate membership includes, Aerospace Corporation, Ascent Logic, Grumman, GTE, Hughes, IBM, Lockheed, Martin Marietta, Mitre, Motorola, and Northrop.

Presently, the Systems Engineering Practices Committee has a number of active Subcommittees serving as technical working groups. Current Subcommittees include:

- (1) Concurrent Engineering
- (2) Metrics
- (3) Risk Management
- (4) Requirements Management
- (5) Principles of SE
- (6) Policy Review
- (7) Tools
- (8) Process Description, and
- (9) Best Practices

Each of these groups is working towards agreement on standards for practice within each specific Systems Engineering specialty area identified.

## **22-4 Standards for Systems Engineering Education**

Develop focused training programs for all applicable management and technical levels. The entire area of training is complex and should be integrated into an overall training plan. SE training should provide a customized curriculum emphasizing avionics systems and avionics issues as examples and case studies.

SE training should also emphasize the integration of all engineering disciplines, such as software engineering and supportability engineering under the SE umbrella. Training in the use of Open Systems standards and COTS should be provided, emphasizing selection guidelines.

Coordinate training with specifics of MIL-STD-499B, the Standard for Systems Engineering. Coordinate with the Systems Engineering Department at DSMC. Utilize the National Council on Systems Engineering (NCOSE) as a systems engineering forum. Since the NCOSE has a wide diversity in membership and application point of view, it provides a forum for reviewing systems engineering concept techniques and tools that are introduced and used throughout the commercial sector of industry as well as in government. As a result, systems engineering concepts and supporting tools developed for the Commercial Off The Shelf (COTS) community are offered as well. Participation in NCOSE will therefore afford the military an opportunity to evaluate and leverage commercial systems engineering approaches for application to military applications.

Additionally, the NCOSE is currently reviewing the preparation of accredited systems engineering curricula. The curricula is being developed by recognized academic leaders in the field of systems engineering in coordination with experienced practitioners in both industry and in the government. As a recognized and accredited engineering specialty,

the future Navy can expect to acquire engineers holding degrees and expert in the technical disciplines that are embodied in systems engineering.

To leverage this effort, the Navy should consider developing an accredited systems engineering curriculum as an avionics/aerospace specialty. This could be done in coordination with the NCOSE, together with the Systems Engineering Department at DSMC, if desired. An available, accredited professional avionics systems engineering degree program at the Masters Degree level could be integrated into the Navy training program (at NAVAIR and at supporting NAWC facilities) through supporting local universities. This approach could provide a knowledgeable corps of systems engineers experienced in the application of SE principles to the development and acquisition of advanced avionics systems.

## **22-5 The Role of the Society of Automotive Engineers (SAE) Avionics Systems Division**

The SAE has long served the needs of the Aerospace community through activities of its Aircraft Engineering Divisions under the SAE Aerospace Council. Particularly appropriate to DoD avionics standardization is the SAE Avionics Systems Division (ASD). ASD along with its predecessor SAE organization has logged more than twenty five years of service to DoD in providing avionics standardization. Beginning with the standard avionics data bus, which became MIL-STD-1553, a prime integrating tool for modern avionics, the SAE ASD has oversight over MIL-STD-1760, The Stores Interface Standard, and MIL-STD-1750 for Standard Military Microprocessors. The SAE led the way in developing a concept for high speed data bus systems (HSDB) for military aircraft. The SAE Standard for a Linear Token Passing HSDB Protocol is the basis for 50 Mb/S buses employed in the F-22 and RAH-66 aircraft.

The SAE ASD has a broad international membership of leading aerospace and avionics systems developers and system integrators as well as component manufacturers. It has a broad representation from all U.S. military services and forms a broad standardization organization suitable for dealing with the overall issues of military avionics standardization.

## 23-0 IMPLEMENTATION STRATEGIES AND PLANS

### General Approach

In preparing the beginnings of an implementation plan to utilize the findings of this review, the following ideas become clear. The key finding or "backbone" for implementation is the rigorous use of Systems Engineering, customized as required for avionics systems. The principal focus of our avionics development of necessity is now Affordability, or more appropriately a quality, supportable product that is also affordable. A Systems Engineering basis becomes the "backbone" for a process that leads us to high performance high quality affordable avionics.

### 23-1 Apply Open System Standards through the SE Process

The other key findings then become modifiers to the SE process. Endorsing an Open System Architectural approach based on Open Interface Standards allows us to move into the mainstream of non-proprietary standardization. Much current standardization in the Open System arena is concerned with Computer/Network standards based on the International Standards Organization (ISO) model for computer communications called the Open System Interconnect (OSI) model. The advantage of using Open Standards where suitable, is that these standards will be well supported by the commercial industry sector as well as the DoD industry sector with suitable military endorsement. This should provide us with a broader industrial base to support our systems applications. Open systems should provide for portability of software and overall scalability; both features that will aid in obtaining affordable avionics products.

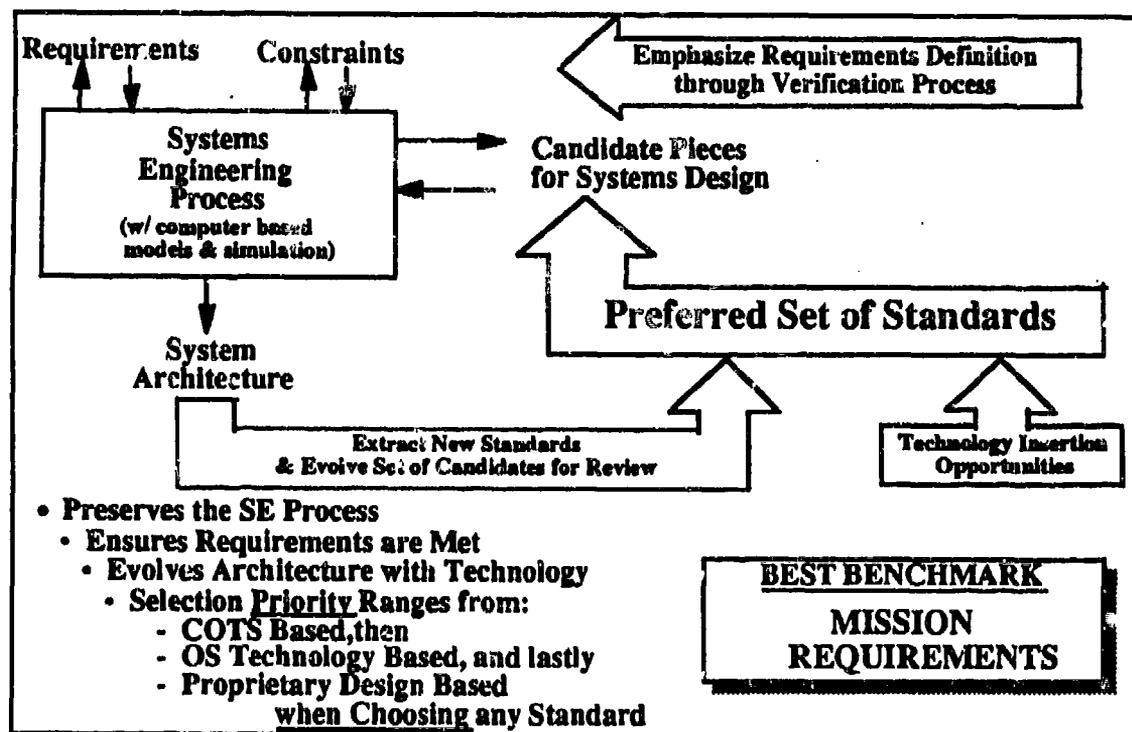


Figure 23-1 The Role of the Systems Engineering Process in Standards Selection

Figure 23-1 illustrates the central role that can be played by the Systems Engineering Process in establishing a complement of effective system standards.

Within the figure the SEP is represented as a functional Block with inputs received as system requirements and as constraints to the process. For avionics systems it is understood that the SEP Block is customized for the specific needs of avionics systems and is to be in harmony with the current avionics development process (e.g. customized to work with current avionics technologies, development tools and manufacturing processes). Through application of the process, standards that represent architectural elements (architectural components) and standard interfaces are utilized to conceive candidate architectural solutions. Suitable Open System standards can be utilized as elements or "candidate pieces" of architectural solutions. As standards are utilized in developmental avionics architectures they acquire an application history and, if successful, are candidates for the preferred avionics standard list. Although explicit avionics requirements vary with the specific needs and constraints of each program, it is expected that a core set of standards will emerge to provide some overall standardization and consistency to future Navy avionics. If the system requirements analysis is rigorously performed to ensure that the system development requirements are well defined and clearly match the system's performance needs, application of the SEP will ensure the availability of an optimal set of preferred standards. Additionally this process allows the architecture to evolve with technology advances. As standards that reflect new technologies are prepared they can be infused into the preferred standards set through the same SEP process as a need is demonstrated.

## 23-2 Promote a Rational Use of COTS/NDI Under SE Guidelines

Emphasis on COTS and NDI products is also a DoD endorsed initiative that should aid in overall affordability. In leveraging the commercial industry we are leveraging a broad sector of the industrial base for our applications. The same SEP approach shown in Figure 23-1. is applicable for consideration of COTS and NDI products as well. By undergoing the same SEP considerations as are applied to "designed for military use" products, COTS and NDI products can be considered for avionics architecture designs. The responsibility placed on the SEP is to ensure that the selection criteria for COTS/NDI is as rigorous as for military products so that it's use provides a "real" net gain in overall affordability without significant losses in performance. Where the COTS/NDI products are designed to Open Systems standards, those standards become candidates for the preferred standards set.

At another level it is important that we support the value of joint standardization efforts among the military services. Leveraging COTS and Open System standards for military avionics enhances the affordability for our avionics. Leveraging COTS and Open System standards on a joint service basis expands the potential scale for COTS/Open Systems based products application to the military avionics market. To make this approach work effectively, the Joint Services must adopt generally the same SE basis for avionics application and similar strategies for the use of Open System standards and COTS.

## 24-0 SYSTEMS ENGINEERING SUMMARY

Systems Engineering is precise and rigorous. Its elements have been well defined and its proper mode of application is well known. Policy defines the need and puts in place the requirement to fully utilize Systems Engineering in weapon systems engineering. The Systems Engineering Process is well integrated into the acquisition life cycle in all phases. If the process is in place and its value is unquestioned, then why do we have problems in efficiently and effectively developing complex avionics systems?

The simple answer is because we haven't yet become consistent and fully proficient in applying the Systems Engineering Process. There are a number of reasons for this situation. First and foremost, the complexity of the systems engineering process demands a great deal of experience and senior leadership. No one will be expert their first time through. Secondly from the top (the Program Manager), down through the ranks, the teams put together on the government side are often spread thin with respect to applied development experience. An additional factor is often a lack of a sophisticated knowledge of the advanced technologies applied to advanced avionics. To some extent, we have left our technological roots behind in pursuit of managerial excellence. A knowledge of the technology is still critical to the acquisition of sophisticated avionics, which remains heavily dependent on the application of advanced technologies.

The evolving trend toward the use of Integrated Product Teams is relatively new and is not fully in place in the Navy Aviation Team. A problem with the use of IPTs is that it represents a major culture change for acquisition. Significant training is necessary before integrated team approaches can be used effectively. The IPT concept also suffers from sparse staffing of programs. The Program Manager must make up for his lack of full headquarters staffing by ever increasing reliance on field activity and contract support personnel. The Field Activities to some extent suffer the same depletion of technical personnel from their ranks as does the PM Office.

In addition to the above, the Systems Engineering Process is being redefined. As a recently recognized engineering discipline, it is still undergoing definition and change. The structure for rigorous application is not yet consistently in place in either industry and DoD. The recent revision of the Systems Engineering Standard, MIL-STD-499B, provides a modern basis for practice, but is not yet formally approved. It is important to recognize the full impact of this revised standard. Since the release of the present draft standard in May 1992, it has served as a catalyst for renewed activity in the practice of Systems Engineering. The Air Force has started the development of a series of five handbooks to support the revised process. Industry has enthusiastically endorsed the document and urges that it be officially released as soon as possible. The National Council on Systems Engineering (NCOSE) participated in the preparation and review of the document and have used it as a structural reference for the organization of NCOSE and to develop focus areas for standardization. As a result many industrial concerns have been actively engaged in the review and revision of their own Systems Engineering Process to accommodate the revisions in the process. Techniques advocated in MIL-STD-499B, including the use of an engineering WBS to structure the process and provide a basis for, interdisciplinary product teams, and the use of incremental technical reviews for more effective process control are new and will take some time to fully integrate into the acquisition process.

Moreover, the computer tools that directly assist the Systems Engineering Process have not been readily available. These tools are now rapidly advancing in capability and are now being focused directly on specific Systems Engineering tasks. It is now possible to select a Systems Engineering tool set that supports the acquisition process from concept

through development and deployment. The problem with Systems Engineering tools is that they are costly and not yet integrated as a full set. However this situation is rapidly changing as the tools are maturing and interfaces are being constructed to effectively interface many of the most popular tools to each other.

In short, there is no single simple answer to a lack of consistent application of Systems Engineering to avionics development. Putting in place and emphasizing the use of a development process that makes effective use of Systems Engineering principles and tools for advanced avionics systems is an important step towards satisfying a demonstrated need.

## APPENDIX A

### SYSTEMS ENGINEERING GLOSSARY

#### 1.0 Definition of Terms Commonly Encountered in Systems Engineering

**Acquisition Decision Memorandum (ADM).** A memorandum signed by the milestone decision authority that documents decisions made and the exit criteria established as the result of a milestone decision review or in-process review.

**Acquisition Process.** The process of bringing a system into being. This includes the phases of conceptual design and advance planning, preliminary system design (demonstration and validation), detail design and development (full scale development), and production and/or construction. Given a defined need, the process includes those steps leading from the requirements definition stage to the delivery of the system for use.

**Allocation.** The top-down distribution, or apportionment, of system-level requirements to the subsystem, equipment, software, unit, or below, to the depth necessary for providing criteria as an input to design. This process tends to promote a top-down "systems approach" in helping to establish specific design requirements for all levels of the system hierarchy as appropriate.

**Allocated Baseline.** The initially approved documentation describing a Configuration Item's (CI) functional, performance, interoperability, and interface requirements that are allocated from those of the system or a higher level CI; interface requirements with interfacing CIs; design constraints; derived requirements (functional and performance); and verification requirements and methods to demonstrate the achievement of those requirements and constraints. The allocated baseline is typically placed under Government control during Engineering and Management Development. There is an allocated baseline for each CI.

**Assemble.** The physical act of putting the document or description together using the inputs of all appropriate supporting disciplines. To be the focal point of the activity and to be responsible for the quality and coherency of the output.

**Authentication.** An act by the Government that results in the Government approving and taking control of a configuration baseline.

Computer-aided Acquisition And Logistic Support (CALS) The application of computerized technology and available computer software to the entire spectrum of logistics. This includes the use of computer methods/tools in the design for system supportability (integrated into CASE/CAE/CAD activities), in the development of logistic support analysis data in determining logistic resource requirements, in the provisioning and acquisition of the identified elements of support (e.g., spare/repair parts, test and support equipment), and in the assessment of the system support capability in the user's environment. CALS also includes the development of technical manuals and the processing of design data automatically and using a digital data format.

Computer-Aided Design (CAD). The process of utilizing computer capabilities and available software to support detailed engineering design activities. CAD tends to deal primarily with three-dimensional graphics, circuit board layouts, the accomplishment of various categories of analyses, and the like.

Computer-Aided Engineering (CAE) . The process of utilizing computer capabilities and available software to support engineering design activities. CAE tends to deal primarily with engineering analyses and lower-level design activity, similar to the functions of CAD.

Computer-Aided Manufacturing (CAM). The process of utilizing computer capabilities, available software, numerical control equipment, robotics, and related resources to manufacture and produce products through automated means. CAM tends to deal with production process planning, materials handling, manufacturing, inventory control, and production management.

Computer-Aided Systems Engineering (CASE). The process of integrating system engineering concepts and constructs, computer capabilities, the use of analytical methods, and available software in such a manner as to complete system engineering functions in an effective and efficient manner. CASE represents a broader level of capability than either CAD or CAE.

Computer-Integrated Manufacturing (CIM). The process of utilizing computer capabilities and available software to manufacture products via automated means. CIM is used in a manner similar to CAM, except that CIM tends to emphasize the use of microcomputers and a common database.

Concurrent Engineering. "A systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product life cycle from conception through disposal, including quality, cost, schedule, and user requirements."

Coordinate. The act of harmonizing an approach or the activities associated with generating a document or performing a task.

Configuration Baselines. Designated points in the system design and development process where the system configuration is defined in detail. Common points include the (1) Functional Baseline, where the system configuration is described through the definition of operational requirements and the maintenance concept, the System Type "A" Specification, and feasibility study reports; (2) Allocated Baseline, where the system configuration is defined through a combination of Development, Process, Product, and Material Specifications (Types "B," "C," "D," and "E"), selected design tradeoff study reports, and system/subsystem design data; and (3) Product Baseline, where the system configuration is defined through a combination of Process, Product, and Material Specifications (Types "C," "D," and "E"), trade-off study reports, detailed design data (drawings, parts lists), supplier data, and the like.

Configuration Control. Deals with the categorization and control of proposed design changes, that is, Class 1 Changes that affect form, fit, and/or function, and Class 2 Changes that are relatively minor in nature. Given a designated configuration baseline, all changes applied to that baseline must be closely evaluated and controlled.

Configuration Control Board (CCB). A board, consisting of expertise representing different design disciplines, responsible for the reviewing and approving of changes to a given configuration baseline.

Configuration Item (CI). An aggregation of hardware, firmware, or computer software or any of their discrete portions, which satisfies an end use function and is designated by the Government for separate configuration management. Configuration items may vary widely in complexity, size and type, from an aircraft, electronic, or ship system. Any item required for logistic support and designated for separate procurement is a configuration item.

**Configuration Management (CM).** A management process used to identify the functional and physical characteristics of an item in the early phases of its life cycle, control changes to those characteristics, and record and report change processing and implementation status. CM involves four functions to include (1) configuration identification, (2) configuration control, (3) configuration status accounting, and (4) configuration audits. CM is the concept of "baseline" management.

**Contract Structure.** The type of contract negotiated between the customer and the contractor, and/or between the contractor and the supplier. Major categories of contracts include Firm-Fixed-Price (FFP), Fixed-Price-with-Escalation, Fixed-Price-Incentive, Cost-Plus-Fixed-Fee (CPFF), cost-plus-Incentive-Fee (CPIF), Cost-Sharing, Time and Materials, and Letter Agreement. The nature and type of contract negotiated are major considerations in the implementation of system engineering requirements.

**Control Hierarchy Analysis.** A functional analysis to determine if any constraints should be applied to the systems functional design and how those constraints will affect the design implementation.

**Cost Breakdown Structure (CBS).** A breakdown of cost in functional terms. All future costs associated with activities throughout all phases of the system life cycle must be included, and costs must be broken out to the depth required to provide the necessary visibility relative to different elements of the system and/or different program activities.

**Cost Effectiveness (CE).** The measure of a system in terms of its technical characteristics and life-cycle cost. Technical characteristics may include a combination of performance, capacity, range, weight and size, reliability, maintainability, supportability, producibility, quality, and related parameters. Life-cycle cost may include all future costs associated with research, design and development, production and/or construction, distribution, system utilization, sustaining maintenance and support, retirement, disposal, and the recycling of materials as necessary. These technical characteristics may be combined in some manner to provide a measure of "System Effectiveness." Cost Effectiveness can be expressed as the ratio of System Effectiveness to Life-Cycle Cost (LCC).

**Cost and Operational Effectiveness Analysis (COEA).** An analysis of the estimated costs and operational effectiveness of alternative materiel systems to meet a mission need and the associated program for acquiring each alternative.

**Define.** To be responsible for specifying an approach for accomplishing a designated task, or the design implementation to solve a given problem.

**Derived Requirements.** Those characteristics of a system that are not expressly specified by the Government, but are necessary to have system elements accomplish their intended functions.

**Design Review.** A formal review of the system configuration as it is defined at a specific point in time. Formal design reviews include the Conceptual Design Review, one or more System (Preliminary) Design Reviews, one or more Equipment/Software Design Reviews, and a Critical Design Review. Formal design reviews include the "checks and balances" necessary in the implementation of system engineering functions.

**Design-To-Cost (DTC).** A quantitative "design to" figure-of-merit specified as a system requirement during conceptual design and included in the System Type "A" Specification. The DTC figure-of-merit should be specified in terms of life-cycle cost. This can, in turn, be broken down into "Design to Unit Acquisition Cost" and "Design to Unit Operation and Support Cost" (or something equivalent). The basic categories of cost should be defined in terms of what is (or is not) included.

**Develop.** To be responsible for the progression of a concept or document into an end product.

**Earned Value.** The actual amount of budget expended and credited to the program to complete an authorized WBS task.

**Effectiveness.** A measure of the system in terms of its technical characteristics. This measure, or figure-of-merit, which will vary depending on the type of system and its mission, may be derived from a combination of performance factors, weight and size, capacity, reliability, maintainability, supportability, quality, and so on.

**Effectiveness Analysis.** An analytical approach used to determine how well a system performs in its intended utilization environment.

**Environment.** The natural environment (weather, climate, ocean conditions, terrain, vegetation, space conditions); combat environment (dust, fog, nuclear-chemical-biological); threat environment (effects of existing and potential threat systems to include electronic warfare and communications interception); operations environment (thermal, shock, vibration, power variations); transportation and storage environment; maintenance

environment; test environments; manufacturing environments (critical process conditions, clean room, stress) and other environments (e.g. software engineering environment, electromagnetic) related to system utilization.

Evolutionary Acquisition. An adaptive and incremental strategy applicable to high technology and software intensive systems when requirements beyond a core capability can generally, but not specifically, be defined.

Exit Criteria. Program specific accomplishments that must be satisfactorily demonstrated before an effort of program can progress further in the current acquisition phase or transition to the next acquisition phase. Exit criteria may include such factors as critical test issues, the attainment of projected growth curves and baseline parameters, and the results of risk reduction efforts deemed critical to the decision to proceed further. Exit criteria supplement minimum required accomplishments and are specific to each acquisition phase.

Facilitate. To help the process to be free from obstacles, to make easier. To act as the catalyst in the process.

Feasibility Analysis. The early investigation, study, and determination of possible technical design approaches in response to a defined need for a new system configuration. This includes the evaluation and comparison of new technologies, as well as the accomplishment of applied research in areas where additional knowledge is desired.

Function. A task, action or activity that must be performed to achieve a desired outcome.

Functional Analysis. The process of translating system-level requirements into detailed design criteria leading to the development of system components. Given the definition of system operational requirements and the maintenance concept, the next step is to define the system in functional terms, identifying the WHATs in terms of specific requirements. This can be accomplished through a series of functional block diagrams. These functions (to include both operational and maintenance functions) are broken down into sub-functions, trade-off studies are accomplished to determine the HOWs associated with the accomplishment of the sub-functions, and resources are identified in terms of human requirements, equipment requirements, software, data, facilities, and so on. Again, this is a top-down process stemming from system-level requirements and leading to the identification of specific design requirements.

Functional Analysis. [alternate] Examination of a defined function to determine all of the sub-functions necessary for the accomplishment of that function. The sub-functions are then analyzed to determine constraints (depicted in a Control Hierarchy Diagram), sequencing (shown in a Functional Flow Diagram), and subsequently architecture and interfaces (depicted in a Functional Architecture Diagram).

Functional Architecture. The arrangement of functions into logical functional domains, including both their internal and external functional interfaces, as well as their respective functional and performance requirements.

Functional Baseline. The initially approved documentation describing a system's or CI's functional, performance, interoperability, and interface requirements and the verification required to demonstrate the achievement of those specified requirements. This baseline is normally placed under Government control during Demonstration and Validation.

Functional Requirement. The necessary task, action, or activity that must be accomplished. Top-level functions are identified by requirements analysis and subdivided by functional analysis.

Human Factors. The characteristics of system design that relate to human element of the system. Considerations in design must include anthropometric factors, human sensory factors, physiological factors, and psychological factors.

Integrated Logistic Support (ILS). A management function that provides the initial planning, funding, and controls that help to ensure that the ultimate consumer (or user) will receive a system that will not only meet performance requirements, but one that can be expeditiously and economically supported throughout its programmed life cycle. The basic program elements include the initial planning for logistic support, the design for supportability, the analysis and acquisition of the various elements of support, and the assessment of the system's support capability in the field.

Interface Requirement. The functional, performance, electrical, environmental, human and physical requirements and constraints that exist at a common boundary between two or more functions, system elements, configuration-items, or systems.

**Interoperability.** The ability of systems, units, or forces to provide services to or accept services from other systems, units, or forces and to use the services so exchanged to operate effectively together.

**Item.** A non-specific term used to denote any product, including systems, subsystems, assemblies, subassemblies, units, sets, parts, accessories, computer programs or computer software.

**Level 1 (Weapon System Level) of the Systems Hierarchy.** The top level of the system hierarchy. Systems Engineering activities primarily focused on programmatic issues, planning and technical support to the Program Manager.

**Level 2 (System Level) of the Systems Hierarchy.** The second level of the system hierarchy. Systems engineering activities primarily focused on technical issues, including requirements allocation to the Integrated Product Teams, interface management between major segments of the weapon system (e.g. Air Vehicle, Support, Training, Production and Business Operations), and technical interface with the customer.

**Level 3 (Segment Level) of the Systems Hierarchy.** The third level of the system hierarchy. Typically the upper level of design activity for the various segments of the weapon system.

**Life Cycle.** The planned life cycle of the system from the initial identification of a need through the retirement and phase out of that system. It usually includes the phases of conceptual design and advance planning, preliminary system design (demonstration and validation), detail design and development (full scale development), production and/or construction, system operation (utilization), sustaining maintenance and support, and retirement. Although the life cycle (and its phases) may change because of budgetary limitations, the introduction of obsolescence, and so on, it is still essential that a life-cycle approach be assumed.

**Life-Cycle Cost (LCC).** The composite of all costs associated with the activities planned and/or accomplished throughout the system life cycle. This includes the costs of research and development, design, production/construction, operation use, maintenance and support, and system retirement.

**Life Cycle Cost (LCC).** [alternate] The total cost to the Government of acquisition and ownership of that system over its useful life. It includes the cost of development, acquisition, support and, where applicable, disposal.

**Life Cycle Resources.** All resources required for development, manufacturing, verification, deployment, operations, support, and disposal (including by-products) of an item throughout its life cycle. Also included are the resources required for training personnel in the operations and maintenance of an item throughout its life cycle. These resources are measured in terms of:

- a. Time (e.g., time required to develop and/or produce the item);
- b. Dollars (e.g., RDT&E, production, operations and support);
- c. Manpower (e.g., number of people required to develop, produce, support, and operate an item); and
- d. Strategic materials.

**Low-rate Initial Production (LRIP).** The production of a system in limited quantity to provide articles for operational test and evaluation, to establish an initial production base, and to permit an orderly increase in the production rate sufficient to lead to full-scale production upon successful completion of operational testing.

**Maintainability.** The characteristics of design that deal with the ease, accuracy, safety, and economy in the performance of maintenance actions. The measures most commonly associated with maintainability are maintenance elapsed times (*Mct*, *MTTR*, *Mpt*, *M*, *MDT*), maintenance frequency factors (*MTBR*, *MTBM*), maintenance personnel labor hour factors (*MMH/OH*, *MMH/MA*, *MMH/Month*, *MLH/OH*), and maintenance cost (*\$/MA*). When addressing maintenance elapsed time only, maintainability can be defined as the probability that a system can be retained in or restored to a satisfactory operating condition, when maintenance is performed by personnel with specified skills, using approved procedures and resources, at each designated level of maintenance.

**Management Information System (MIS).** A data collection, processing, and handling capability that supports management in the implementation of program requirements. A prime objective for system engineering is the establishment of a good communications network that will allow for a rapid assessment of program status, and for the initiation of corrective action in an expeditious manner.

**Measure of Effectiveness (MOE).** A metric used to quantify the performance of system products and processes in terms that describe the utility or value when executing customer missions. Systems Engineering uses MOEs in a variety of ways including decision metrics, performance requirements, and in assessments of expected performance. MOEs can include cost effectiveness metrics.

**Mission Need.** A statement of operational capability required to perform an assigned mission or to correct an efficiency in existing capability to perform a mission.

**Monitor.** To check and keep track of the progression of an activity in a non-intrusive manner. To make assessments as to the adequacy of the resulting approach.

**Operational Effectiveness.** An OT&E metric that measures the overall degree of mission accomplishment of a system when used by representative personnel in the environment planned or expected (e.g. natural, electronic, threat) for operational employment of the system considering organization, doctrine, tactics, survivability, vulnerability and threats. The operational system that is provided to users will be evaluated for operational effectiveness by a service OT&E agency.

**Perform.** To begin a process or task and carry it through to completion.

**Performance.** The characteristics of system design that relate to such measures as input-output requirements, throughput, capacity, size and weight, range, accuracy, power output, data transmitted per designated increment of time, etc.

**Producibility.** The characteristics of system design that relate to the ease, accuracy, and economy associated with the follow-on manufacture of system elements in multiple quantities as required. The objective is to design products that can be easily produced in multiple quantities, using conventional manufacturing processes.

**Requirements Allocation Document (RAD).** A document which contains all technical requirements allocated to a particular product definition team. It includes both direct, i.e., contract specifications, and derived, i.e., MDA self-imposed, requirements. Derived requirements fall into two categories: those allocated from other teams (such as Reliability/Maintainability requirements) and those derived by the respective Subsystem Manager for their equipment.

**Requirements Allocation Matrix (RAM).** A document (in matrix form) which relates contract specification paragraphs to a particular product definition team charged with the responsibility for ensuring design incorporation and/or compliance.

**Reliability.** The characteristics of design that relate to the ability of a system to perform for a designated period of time. More specifically, it can be defined as the probability that a system or product will perform in a satisfactory manner for a given period of time when used under specified operating conditions. The measures of reliability include MTBF, MTBM and MTTF.

**Risk.** A subjective assessment made regarding the likelihood or probability of not achieving a specific objective by the time established with the resources provided or requested. It also refers to overall program risk.

**Risk Management.** An organized, analytic process to identify what can go wrong, to quantify and assess associated risks, and to implement and control the appropriate approach for preventing or handling each risk identified.

**Risk Management [alternate].** An organized method for identifying and measuring risk, and for selecting and developing options for the handling of risk. Risk management must be addressed in the System Engineering Management Plan (SEMP), and it includes the functions of risk assessment, risk analysis, and risk abatement.

**Risk Management Plan.** Description of the risk management program that describes the approach and activities for risk management. The technical risk management plan is an essential part of the SEMP.

**Supportability.** The characteristics of system design that deal with the ability of a system to be supported in an effective and economic manner. These characteristics pertain not only to the prime elements of the system, but to the design of test and support equipment, the supply support capability, training equipment, facilities, maintenance software, and soon. Many of the principles of reliability, maintainability, and human factors are included.

**Specification.** A document prepared to support acquisition and life cycle management that clearly and accurately describes essential technical requirements and verification procedures for items, materials and services. When invoked by a contract it is legally enforceable and its requirements are contractually binding.

**Specification Tree.** The hierarchical depiction of all the specifications needed to control the development, manufacture and integration of items in the transition from customer needs to the complete set of system products and processes that satisfy these needs.

**Statement of Work (SOW).** The non-specification work tasks to be completed by the contractor.

**Subsystem.** A grouping of items satisfying a logical group of functions within a particular system.

**System.** A set of components working together with the common objective of performing a function in response to a designated need. A system constitutes a complex set of resources integrated so as to fulfill a defined mission scenario. Such resources may take the form of human beings, materials, equipment, software, facilities, data, and so on. The system must have purpose, it must be functional, be able to respond to some identified need, and it should be able to achieve its overall objective in a cost-effective manner.

**System Analysis.** An on-going iterative analytical process, included as part of the system engineering process, involving the evaluation of design approaches, the accomplishment of trade-off studies, and so on. System analysis is accomplished through the appropriate use of various operations research methods to assist in problem resolution (simulation, queuing theory, linear and dynamic programming, networking, etc.).

**Systems Engineering.** The effective application of scientific and engineering efforts to transform an operational need into a defined system configuration through the top-down iterative process of requirements definition, functional analysis, allocation, synthesis, design optimization, test, and evaluation. It involves the design engineering process of bringing a system into being, with emphasis on an integrated, top-down, life-cycle approach.

**Systems Engineering Detailed Schedule (SEDS).** The detailed, task oriented schedule of the work efforts required to support the events and tasks identified in the SEMS. The SEDS is used to track day-to-day progress and includes the continual assessment of the technical parameters required to support each SEMS task/event.

**System Engineering Management.** The management activities necessary for the implementation of system engineering requirements. This includes the initial planning, organization for system engineering, and the on-going program evaluation and control activities to ensure that system engineering objectives are met.

System Engineering Management Plan (SEMP). The principal management-oriented document covering the implementation of system engineering program requirements. This plan is developed during the conceptual design phase of a program, includes the results of some advanced planning, and leads into the requirements for the subsequent phases of system acquisition.

Systems Engineering Management Plan (SEMP). [alternate] A comprehensive document that describes how the fully integrated engineering effort will be managed and conducted.

Systems Engineering Master Schedule (SEMS). A compilation of key accomplishments requiring successful completion to pass identified events. Accomplishments include major and critical tasks, activities and demonstrations, with associated accomplishment criteria. Events include technical reviews and audits, demonstration milestones, and decision points. Successful completion is determined by the measurable criteria defined for each accomplishment. Examples of the criteria include completed work efforts and technical parameters used in TPM. Quantitative inputs into program decision points.

Systems Engineering Process. A comprehensive, iterative, problem solving process that is used to: a) transform validated customer needs and requirements into a life-cycle balanced solution set of system product and process designs; b) generate information for decision makers; and c) provide information for the next acquisition phase. The problem and success criteria are defined through requirements analysis, functional analysis, and system analysis and control. Alternative solutions, evaluation of those alternatives, selection of the best life-cycle balanced solution, and the description of the solution through the design package are accomplished through synthesis and system evaluation activities.

System Integration. Involves the technical integration of system elements as the design and development effort progresses, along with the integration of the various design and supporting disciplines into the overall design effort from a managerial perspective. Later during detail design and development, the integration process often involves a bottom-up approach relative to the combining of the various system components into subassemblies, subassemblies into assemblies, assemblies into units, until a totally integrated system is functioning in accordance with the initially specified requirements.

System Specification. The top-level technical document that defines the basic requirements for *system* design and development.

**System Synthesis.** The combining and structuring of components in such a way as to represent a feasible system configuration. This may be accomplished on a number of occasions throughout the system design and development process, and the particular configuration structured may not reflect the final design approach. In essence, one needs to define a configuration in such a way that it can be evaluated.

**Technical Performance Measurement (TPM).** The continuing verification of the degree of anticipated and actual achievement of technical parameters. TPM is used to identify and flag the importance of a design deficiency that might jeopardize meeting a system level requirement that has been determined to be critical. Measured values that fall outside of an established tolerance band require proper corrective actions to be taken by management.

**Technical Performance Measures (TPMs).** Those measures of the system, or of program activities, that are considered as being critical for the successful accomplishment of system engineering objectives. Specific quantitative parameters, reflecting the basic design characteristics of the system, are specified initially in the System Specification (Type "A"). Then, as design and development progresses, these factors need to be integrated into the periodically scheduled program/design review process for comparison against the initially specified requirements. Finally, an evaluation of the system, in terms of compliance with these requirements, is accomplished through the final system evaluation and test activity. The objective is to identify the critical factors that are performance related.

**Technical Reviews.** A series of systems engineering activities by which the technical progress of a program is assessed relative to its technical or contractual requirements. Conducted at logical transition points in the development effort to reduce risk by identifying and correcting problems or issues resulting from the work completed before the program is disrupted or delayed. Provides a method for the contractor and the Government to determine that the development of a system and/or configuration item and its documentation have met contract requirements. Includes incremental reviews (functional, subsystem and interim system) and major system level technical reviews.

**Test and Evaluation Master Plan (TEMP).** A key planning document, developed during the conceptual design and advance planning phase, covering the proposed integration and testing requirements for the system as an entity. A total integrated test approach is essential.

**Timeline Analysis.** Analytical task conducted to determine the time sequencing between two or more events. Examples of timelines include:

- a. a schedule line showing key dates and planned events;
- b. a mission flight path identifying when and where planned changes in course and velocity take place; and
- c. a portion of an engagement profile detailing time based position changes between a weapon and its target.

**Total quality management (TQM).** A total integrated management approach that addresses system/product quality during all phases of the system life cycle and at each level in the overall system hierarchy. It provides a "before-the-fact" orientation to quality, and it focuses on system design and development activities, as well as production, manufacturing, assembly, construction, and related functions. It includes activities associated with the "design for producibility," quality engineering, quality control, statistical process control (SPC), quality assurance, and supplier evaluation and control.

**Work Breakdown Structure (WBS).** A product-oriented family tree composed of hardware, software, services, data and facilities which result from systems engineering efforts during the development and production of a defense materiel item, and which completely defines the program. Displays and defines the product(s) to be developed or produced, and relates the elements of work to be accomplished to each other and to the end product. Provides structure for guiding multi-disciplinary team assignment and cost tracking and control.

**Work Breakdown Structure (WBS) [alternate]** A product-oriented family tree that leads to the identification of activities, functions, program tasks, sub-tasks, and work packages that must be performed for the successful completion of a given program. It displays and defines the system to be developed, and portrays all of the elements of work to be accomplished. There are "Summary Work Breakdown Structures" (SWBSs) used to show the top three levels of work broken out in a summary manner, and "Contract Work Breakdown Structures (CWBSs) used for the purposes of contract negotiation.

## 2.0 ACRONYMS AND ABBREVIATIONS

### Encountered in the Engineering of Military Avionics

ACAT	Acquisition Category (Navy)
ABL	Assembly Breakdown List
ACCM	Advanced Concepts Cost Model
ADM	Acquisition Decision Memorandum, also Advanced Development Model
ADPLS	Automated Drawing Parts List System
Ail	Aileron
ANOVA	Analysis of Variance
ASIRD	Avionics System Integration Requirements Document
ATP	Acceptance Test Procedure
ATF	Advanced Tactical Fighter
AZ	Azimuth
BIT	Built In-Test
BITE	Built In Test Equipment
BDA	Bomb Damage Assessment
CALS	Computer-aided Acquisition Logistics Support
CAP	Combat Air Patrol
CASE	Computer Aided Software Engineering
CCB	Configuration Control Board
CCI	Candidate Configuration Item
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CDSR	Contract Data Status Report
CE	Concept Exploration
CED	Concept Exploration & Definition
Cf	Consequences of Failure
CFE	Contractor Furnished Equipment
CHD	Control Hierarchy Diagram
CI	Configuration Item
CITIS	Contractor Integrated Technical Information Services
CLIN	Contract Line Item

CM	Configuration Management
CMRS	Calibration Measurements Requirements Summary
CNI	Communication, Navigation, Identification
C&RS	Control & Release System
COEA	Cost & Operational Effectiveness Analysis
COTS	Commercial Off-The-Shelf
CS	Configuration(s) Synthesis
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
C/SCS	Cost/Schedule Control System
C/SCSC	Cost/Schedule Control System Criteria
CSE	Chief Systems Engineer
CWBS	Contract Work Breakdown Structure
DCN	Drawing Change Notice
DAB	Defense Acquisition Board
DCP	Decision Coordinating Paper
DDM	Design Decision Memo
DEM/VAL	Demonstration & Validation
DI	Data Item
DID	Data Item Description
DoD	Department of Defense
DODD	Department of Defense Directive
DODI	Department of Defense Instruction
DTC	Design To Cost
D/V	Demonstration & Validation
ECP	Engineering Change Proposal
ECS	Environmental Control System
EDA	Electronic Design Automation
EDDP	Engineering Design Documentation Procedures
EDM	Engineering Development Model
EL	Elevation
EMC	Electromagnetic Compatibility
EMD	Engineering & Manufacturing Development
EM	Electromagnetic Interference

EO	Electro-Optics
ESP	Engineering Standard Practice
EW	Electronic Warfare
FA	Functional Analysis
FAD	Functional Architecture Diagram
FFD	Functional Flow Diagram
FFRR	First Flight Readiness Review
FOR	Functional Qualification Review
GFE	Government Furnished Equipment
HDBK	Handbook
H/W	Hardware
HWCI	Hardware Configuration Item
ICD	Interface Control Document
ICS	Interface Control Sheet
ICWG	Interface Control Working Group
IDP	Individual Training Plan
IFFC	Integrated Flight/Fire Control
ILS	Integrated Logistics Support
ILSP	Integrated Logistics Support Plan
IOC	Initial Operating Capability
IPD	Integrated Product Development
IPT	Integrated Product Teams
IR	Infrared
I&R	Interchangeability & Replaceability
IRS	Interface Requirements Specification
IWARS	Interactive Warfare Simulation
LCC	Life Cycle Cost
LH	Left Hand
LPI	Low Probability of Intercept
LRIP	Low Rate Initial Production
LSA	Logistics Support Analysis
LSAR	Logistics Support Analysis Records
LSC	Logistics Support Cost
MFD	Multifunction Display

MIL	Military
MIL-STD	Military Standard
MIMIC	Microwave and Milimeterwave Integrated Circuits
MNS	Mission Needs Statement
MOE	Measure of Effectiveness
MP	Maintainability Plan
MRF	Multirole Fighter
MS	Acquisition Milestone
MSRB	Management Safety Review Board
MTBF	Mean Time Between Failures
MTBM	Mean Time Between Maintenance
MTBO	Mean Time Between Overhaul
MTTR	Mean Time To Repair
NOR	Notice Of Revision
OJT	On-The-Job Training
O/S	Operations & Support
ORD	Operational Requirements Document
P/D	Production & Deployment
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
PERT	Performance Evaluation Review Technique
P <sup>3</sup> I	Pre-Planned Product Improvement
Pf	Probability of Failure
PM	Program Manager
PMO	Program Management Office
PPM	Program Performance Measure
PRR	Production Readiness Review
PSC	Preferred System Concept
PSCN	Proposed Specification Change Notice
PSP	Programmable Signal Processor
QFD	Quality Function Deployment
RA	Requirements Analysis
RAD	Requirements Allocation Document
RAM	Requirements Allocation Matrix

RAS	Requirements Allocation Sheet
RF	Radio Frequency
RH	Right Hand
R&D	Research and Deveioption
R&M	Reliability and Maintainability
RFP	Request for Proposal
RP	Reliability Plan
SAM	Surface-to-Air Missile
SBD	Schematic Block Diagram
SCN	Specification Change Notice
SDD	System Design Document
SDR	System Design Review
SE	Systems Analysis, Control & Evaluation
SECDEF	Secretary of Defense
SEDS	Systems Engineering Detailed Schedule
SEMP	Systems Engineering Management Plan
SEMS	Systems Engineering Master Schedule
SEP	Specialty Engineering Plans
SERD	Support Equipment Recommendation Data
SLDA	System Level Design Automation
SMS	Stores Management System
SOW	Statement Of Work
SPC	Statistical Process Control
SRR	System Requirements Review
SSDD	System/Segment Design Document
SSEMP	System Security Engineering Management Plan
SSPP	System Safety Program Plan
SSR	Software Specification Review
SSS	System/Segment Specification
Stab	Stabilizer
S/W	Software
TCM	Technical Coordination Meeting
TPM	Technical Performance Measurement
TQM	Total Quality Management

TRR	Test Readiness Review
UPC	Unit Production Cost
USAF	United States Air Force
USD(A)	Under Secretary of Defense for Acquisition
USN	United States Navy
VECP	Value Engineering Change Proposal
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuits
VLSI	Very Large Scale Integrated Circuits
VMS	Vehicle Management System—
VOC	Voice of the Customer
VR	Variability Reduction
WBS	Work Breakdown Structure