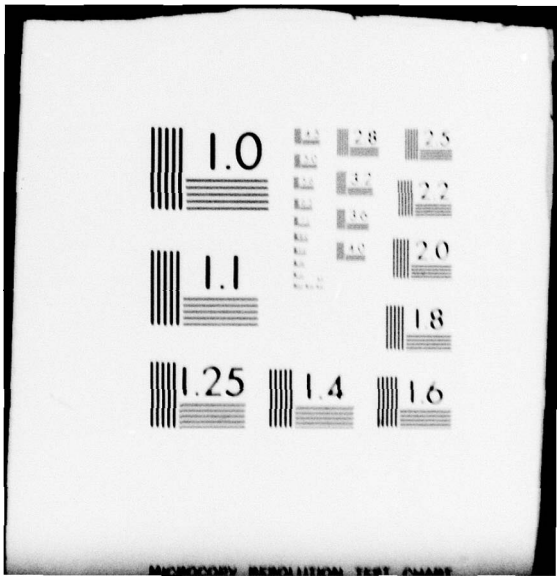
AD-A074 444 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2
A STUDY OF ALTERNATIVES FOR COMPUTER SYSTEMS ACQUISITION FOR TH--ETC(U)
JUN 79 V A NAOUM

UNCLASSIFIED NL

1 OF 3
AD A074444


1.0
1.1
1.25
1.4
1.6
2.0
2.2
2.5
MICROCOPY RESOLUTION TEST CHART

NAVAL POSTGRADUATE SCHOOL

Monterey, California

LEVEL II

(9) Master's THESIS,

(6) A STUDY OF ALTERNATIVES FOR COMPUTER SYSTEMS ACQUISITION FOR THE HELLENIC NAVY

by

(10) Vassilios A./ Naoum

(11) June 1979

(12) 224 p.

Thesis Advisor: U. R. Kodres

Approved for public release; distribution unlimited.

251 450

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Study of Alternatives for Computer Systems Acquisition for the Hellenic Navy		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; June 1979 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Vassilios A. Naoum		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE June 1979 13. NUMBER OF PAGES 223
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, CA 93940		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

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

The achievements of the advancing computer technology, coupled with a dramatic reduction in hardware costs have been acknowledged by smaller Navies such as the Hellenic Navy.

This thesis makes a survey of the existing Naval Tactical Data Systems in countries of the North Atlantic Alliance, the efforts undertaken to improve these systems through hardware and software standardization, the methodology required for the development and support of such systems, and the costs implied.

DD FORM 1 JAN 73 1473 (Page 1)

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Approved for public release; distribution unlimited.

A STUDY OF ALTERNATIVES FOR COMPUTER SYSTEMS ACQUISITION FOR THE HELLENIC NAVY

by

Vassilios A. Naoum
Commander, Hellenic Navy
Graduate of Hellenic Naval Academy, 1957

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1979

Author

Approved by:

Thesis Advisor

Second Reader

Chairman, Department of Computer Science

Dean of Information and Policy Sciences

ABSTRACT

The achievements of the advancing computer technology, coupled with a dramatic reduction in hardware costs have been acknowledged by smaller Navies such as the Hellenic Navy.

This thesis makes a survey of the existing Naval Tactical Data Systems in countries of the North Atlantic Alliance, the efforts undertaken to improve these systems through hardware and software standardization, the methodology required for the development and support of such systems, and the costs implied.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHORUS : And since when is then the city taken?

CLYTAEMNESTRA : I tell you: the night which has born this light?

CHORUS : And which messenger would arrive immediately?

CLYTAEMNESTRA : Hefaestus! sending bright fire from Ida.
And successive fires recruited farewell the
flame here; and Ida first to cape Hermes of
Lemnos, and from there mount Athos third the
glorious fire receives.....
.....
.....
and here arrives the shining to nearby peaks
of mount Arachnaeum, until it hits the roofs
of the Atreidae, the light that has the fire
of Ida as grandfather.....
.....
These then are the symbols and the signals
that my husband has sent me from Troy.

Aeschylus: AGAMEMNON

ACKNOWLEDGEMENT

I wish to thank Professor Uno Kodres, whose expert guidance, adequate help and personal interest, made the emergence of this thesis possible.

I. INTRODUCTION

Thirty five years have elapsed since ENIAC, the first of electronic digital computers was fully developed. During this relatively short period of time, the computer has evolved from a sophisticated laboratory equipment to a sine quanon tool in many areas of human activity, such as industry, commerce, space exploration and national defense.

This computer evolution has been marked by one distinguished trend: ever increasing processing capability in volumes of ever decreasing size. Today's so-called miracle chip has a calculating capability greater than that of the room sized ENIAC.

Micro-miniaturization, the art of making electronic devices many times smaller and lighter than ever before possible or envisaged, originated with the missile and space research, where space and weight requirements made it an absolute necessity. [1] The first technical breakthrough came with the development of the transistor. Since 1959 the number of components fabricated on a single chip (circuit) has doubled every year. The history of micro-miniaturization or Integrated Electronics (as otherwise is called) can be considered as made up of three distinct periods: [2]

* Small-Scale Integration (SSI) (1959-1965)

During this period, the number of components per chip was increased from one with single transistors to approximately 40 with diode-transistor logic (DTL) and transistor-transistor logic (TTL).

* Medium-Scale Integration (MSI) (1965-1969)

During this period the number of components per chip was increased to several hundred, conventional TTL, Schottky TTL and Complementary Metal Oxide Silicon (CMOS) logic families fall into these categories.

* Large-Scale Integration (LSI) (1969-Present day)

In 1969 the capability of the semiconductor manufacturers passed 1000 components per chip. The increase in complexity of integrated circuits has continued, so that in 1978 devices containing 65,000 components were manufactured. In fact it was inevitable, that at some point a computer on-a-chip was developed. Microprocessors, as they are now commonly known are widely used in industrial and commercial applications.

This vast reduction in size of electronic components in computer technology has been attended by:

* Increasing processing capability.

* Dramatic reduction in hardware cost.

* Reduction in electrical power requirements, resulting also in reduced power supply costs.

* Increased reliability (being one of the reasons why micro-miniaturization was undertaken originally).

The advantages of the LSI technology are so vastly significant, that it can be said that they have cleared the horizons to allow computer applications to "be seen" by smaller Navies, like the Hellenic Navy.

The objective of this thesis is to examine possible alternatives for the acquisition of computer systems, mainly in the area of Command Control and Communications (C^3), for the Hellenic Navy and to offer some recommendations in this regard.

Section II contains a consideration on potential application of computer systems for the Hellenic Navy.

Section III contains a rather lengthy examination of the C^3 management in the U.S.A. Certain of the systems considered are definitely beyond the acquisition scope of this thesis, but they have intentionally been included, in order to provide a more complete picture of the C^3 environment, as it exists today and its future trends.

Sections IV and V contains a similar but less extensive examination of the C^3 management in other NATO countries and in the North Atlantic Alliance itself.

Section VI covers the Systems Analysis technique, that should be followed in any computer system application.

Section VII considers the relevant costs to a computer system development.

Section VIII presents the alternatives for computer system acquisition for the Hellenic Navy.

Section IX makes a summary and presents certain conclusions of this thesis.

Section X provides a glossary of terms and definitions used in the text of this thesis.

Finally, Appendices A through G contain technical details of several computer systems and test as a complementary information to the main text.

This thesis is based solely on unclassified bibliography and it must be noted, that certain difficulties were encountered due to the fact that this bibliography is not readily available.

II. POTENTIAL APPLICATIONS OF COMPUTER SYSTEMS IN THE HELLENIC NAVY

Greece is one of the 15 sovereign nations comprising the North Atlantic Treaty Organization (NATO).

Furthermore, Greece will become the 10th member of the European Economic Community on January 1, 1981, when the treaty, signed in Athens on May 29, 1979, becomes active.

The Hellenic Navy is one of the three distinct Services of the Hellenic Armed Forces (the others are the Hellenic Army and the Hellenic Air Force). To carry out its mission, the Hellenic Navy is equipped with surface ships (including guided missile fast patrol boats), modern conventional submarines, maritime patrol aircraft and naval helicopters. A complex of headquarters and naval bases supports the operations of these units.

The application of digital electronic computer systems in the Hellenic Navy can enhance the effectiveness of the functioning of this Service.

The following are considered as potential computer system applications for the Hellenic Navy:

* Data Processing

It can be applied in support of administrative or management functions. The main advantage is the more timely availability of the required information.

Some reduction in the administrative costs can be foreseen but the order of this reduction may not be significant.

Economy in personnel would not be expected.

Generally this application is a step forward but not of tremendous significance.

* Weapon Systems Applications

These applications include reliable modern communications, tactical situation compilation and display, and control of sensors and weapons.

The importance of these applications cannot be overemphasized. They provide the means for the effective conduct of the modern naval warfare and the survival of the naval forces.

The environment where these applications might be exploited is foreseen to have two aspects:

- National environment

  In this environment independent operations would be supported.

- NATO environment

  In this environment joint operations with the Allied Navies would be supported.

III. C^3 MANAGEMENT IN THE UNITED STATES OF AMERICA

Currently, C^3 is the most used U.S. term for force management and control. Literally, it means communications, command and control, and includes data-processing systems that reduce and manipulate information; navigation systems that provide relative and absolute positions; a considerable variety of sensor systems for intelligence, reconnaissance and surveillance. These systems are centrally within the purview of the Office of the Assistant Secretary of Defense for Communications, Command, Control and Intelligence. [3]

C^3 emphasizes joint planning and operations rather than individual service effort and when completed, around 1985, it is expected to provide the interoperability capabilities required for the functioning of the individual service C^3 systems, when operating in a joint environment.

C^3 finds its expression in the so-called World Wide Military Command and Control System (WWMCCS).

Since the WWMCCS is a rather recent development, whereby the individual service C^3 (and for this study the USN) have a history of evolution of more than twenty years, the consideration of the systems will be done in a bottom-up manner, that is, from the individual service to the joint services. Consequently, the life cycle concepts for defense computer resources programs will be examined.

A. NAVY COMMAND AND CONTROL SYSTEMS

1. Shipborne Tactical Systems

a. Naval Tactical Data System (NTDS) [4]

NTDS is the designation given to the combination of digital computers, displays of various types and data links for the on-line collection, processing, storage and presentation of information from sensors such as radar, sonar, optical and aircraft or ship consorts via data link. NTDS is also engineered to interface with ATDS (Airborne Tactical Data System) and MTDS (Marine Tactical Data System).

The primary function of NTDS as well as of ATDS and MTDS is to provide for automated organization and display of information for command and control teams for such purposes as threat detection and assessment and weapon-target allocations.

NTDS is the offspring of studies started at MIT in 1955, while the first test service equipment underwent evaluation in 1961 on board USS MAHAN (DLG-11), USS ORISCANY (CVA-34) and USS KING (DLG-10), which formed the world's first NTDS Task Group.

There are four major components to the NTDS:

* Analog to digital (A/D) converters

Radar and sonar and other analog information is converted into digital form and entered directly into the digital computer.

* Computing equipment

Quick storage and processing of large volumes of information require large capacity, high-speed digital computers. The computer has to communicate to exchange data, perform necessary calculations in accordance with stored programs and provide a picture of the tactical situation for operator comprehension and action. Obviously, computer programming is a crucial consideration in the system. The necessary rules of engagement for threat evaluation and weapon assignment, as well as the other functions, which need to be performed, must all be carefully written down beforehand. Because these rules are constantly changing, the programs must be constructed so that they can be modified without disrupting the operation of the system.

* Communications equipment

In order to work together as a unit, all elements of the force, or even several forces, must be able to exchange information rapidly over high-speed data links. Since the computers themselves can communicate via these links, the result is a well-integrated, coordinated operation.

* Visual displays

The result of all the above must be presented to a human operator for comprehension or decision. Visual display consoles must present the tactical picture and allow the operator to enter new instruction or data into the system. Consoles are used for detection, tracking, identification, threat

evaluation and for weapons assignment and intercept control. In a tactical engagement console functions can be reassigned or switched to an alternate console.

The current NTDS computer program is a collection of large segments, called modules, consisting of instruction, data and control information, generally pertaining to a single operational function or capability, such as tracking. The modules are linked together to carry out the desired functions. Figure III-1 is a block diagram of the NTDS functions performed within the system. [4]

Although designed to a common concept, NTDS exists in a number of versions varying in size and equipment complement according to the age of the installation and the size of the vessel fitted. The overall system has also been subject to continuous process of updating since its introduction into service. By 1969, for instance, three generations of display devices had been employed.

The following computers have been used as basic computing equipment in NTDS installations: [5]

Computer	Civilian Nomenclature	Remarks
AN/USQ-17		Estimated as obsolete
CP642/USQ-20	UNIVAC 1200	Still in use
CP642A/USQ-20	" "	" " "
CP642B/USQ-20	" "	" " "
AN/UYK-7	UNIVAC 1100	Employed in new NTDS
AN/UYK-20	UNIVAC 1616	" " " "

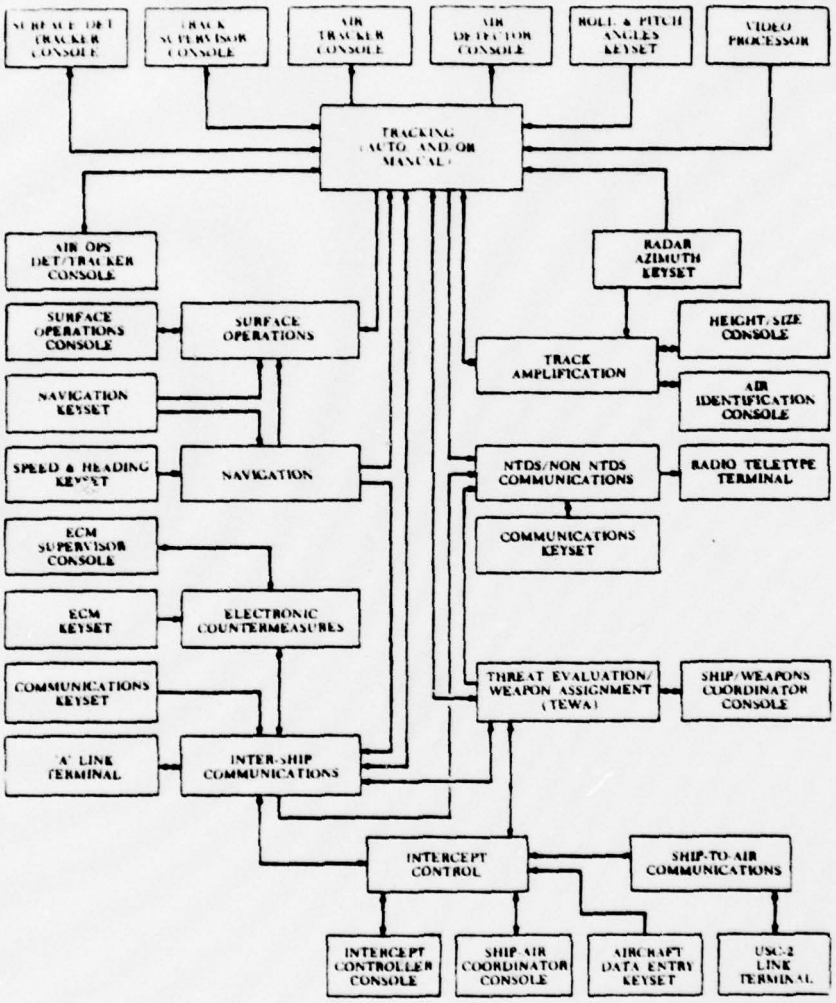


Figure III-1: NTDS Functions (1960)

Technical description of the CP642B and AN/UYK-7 computer systems is given in Appendices A and B respectively.

NTDS uses the CMS-2 high level language for the production of the operational programs.

Restructure of the NTDS [6]

The U.S. Navy is presently restructuring the NTDS software. Two reasons are calling for this restructure:

* In the current NTDS there is significant interdependency between modules with a large dependency on local data stores, thus generating almost intolerable level of Inter-Module/Inter-Computer (IMIC) message traffic. This modular architecture presently requires the modules to be large functionally orientated entities, resulting in rather heavy traffic, that extends the pre-1960 hardware (CP642) to its limits.

* There is a continuous requirement to reduce command decision time in order to insure readiness against evolving threats due to introduction of improved weapons. This requirement consequently demands an improved responsiveness in data processing and the achievement of a greater automation of information processing in the Navy's command and control system.

With the restructure, the identified system requirements are divided into functional areas, which are further subdivided into smaller independent modules called tasks. Each of the individual tasks is designed to be isolated, without interdependency on other tasks, communicating as necessary via the common data base or the executive as shown in Figure III-2.

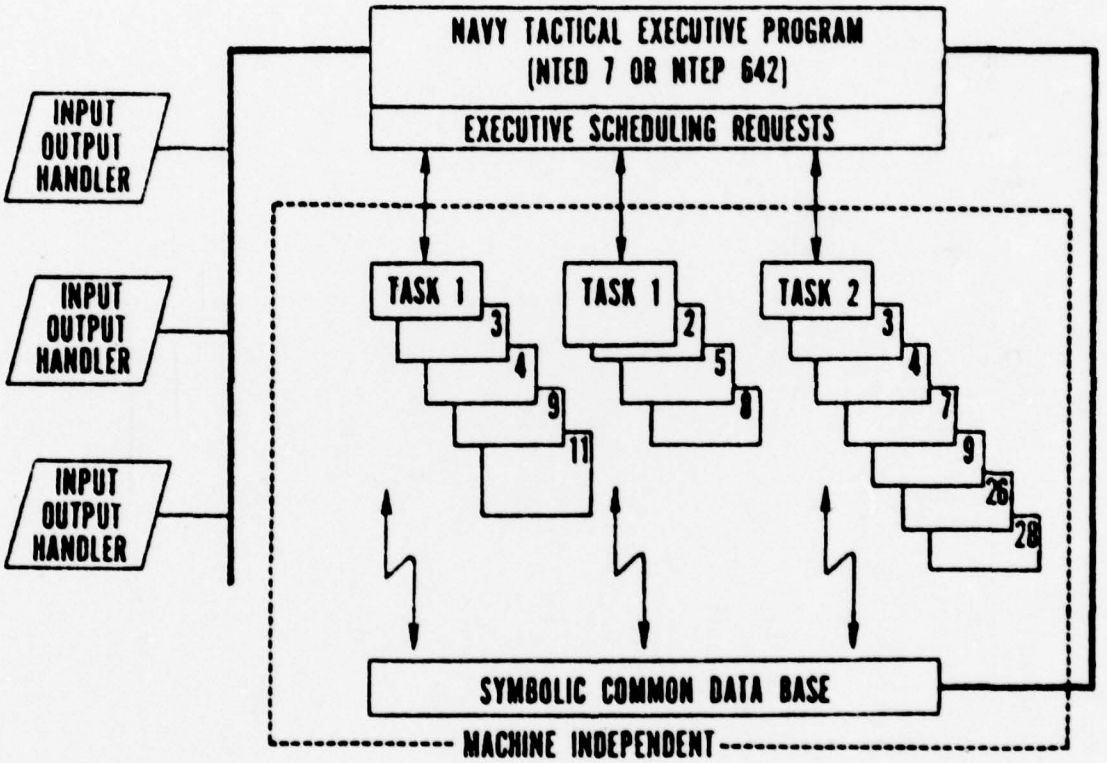


Figure III-2: Communication of tasks via common data base or the executive program in the restructured NTDS.

This will allow the modification or the replacement of the individual tasks, as operational requirements change, without affecting other tasks. Each functional area will be supported by a group of tasks, that meet the specified operational requirements for that function. The development of a capability to maintain a "single source library" of tasks written in a host transferable language and a support program, that will generate a system tape when an NTDS ship class configuration is defined, are key elements of the development of the universal program library for all NTDS ships.

It has been estimated that a minimum of 60% of the program library will be transferable (common) between all NTDS classes. The net effect of this will be to reduce procurement costs of software for the NTDS program to between 10% to 20% of new software costs, using existing programming techniques. It will not be necessary for the USN to continuously go to outside vendors for total NTDS program procurement, since only the new capabilities need to be designed, coded, tested and placed on the universal library.

b. DDG2-Tactical Data System (DDG2-TDS) [7]

This system will be installed in 23 ships of the DDG-2 class under a program known as the "DDG 2 Class Upgrade," during ship overhauls within the period 1979-1984.

The TDS is designed to provide data display, data correlation and processing, lower combat system reaction time, preselected threat response, coordinated control of ship's

weapons and the participation of the DDG-2 class in force actions over a real time link. The TDS will be comprised of the Navy standard AN/UYK-7 computer, AN/UYA-4 displays, conversion and input/output equipment common to many other ship classes.

The TDS capability, as envisioned for the DDG-2, is intended to upgrade the operational effectiveness of the conventional CIC and the combat systems. The operational functions scheduled have almost the same magnitude as those in larger Naval Tactical Data Systems. However due to DDG-2 TDS functions, coupled with the improved sensors and weapons being added during the upgrade, will provide this class of vessels with the capabilities required to fulfill their operational missions in the 1980's.

c. The AEGIS Command and Control System [8,9,10]

AEGIS is a totally integrated weapon system, primarily designed to defend against anti-ship missiles and generally to fulfill the mission of air defense for coming decades.

AEGIS is designed to provide exceptional system availability, fast reaction time, great firepower, immunity to electronic countermeasures and extended intercept range.

The system will be capable of automatically detecting, tracking and destroying airborne, seaborne and land-launched weapons of the future, launched against the defending fleet.

The quick-reaction and high firepower of AEGIS stems from the digital computer-managed operations, coupled with a multi-function phased array radar, capable of simultaneously performing search, fire control-quality tracking and missile midcourse command guidance functions, in the following sequence and as shown in Figure III-3: [8]

* Horizon Search

A highly adaptive normal mode search with an optimized burnthrough capability employing advanced ECM techniques; an almost immediate transition from search to track, minimizing reaction time.

* Track

A track capacity capable of handling high numbers of friendly and hostile targets at adaptive data rates, without interrupting search functions.

* Midcourse guidance

Simultaneously with search and track functions, the phased array radar supplies midcourse guidance commands to several missiles. This permits easier launches, multiple engagements, longer intercept ranges and higher firepower. Rapid cycling, multiple purpose high fire-power launchers sustain an unprecedented launching rate.

* Terminal guidance and illumination

CW illuminators employ increased power density on target and rapid frequency selection for the semi-active terminal phase of missile flight to minimize mutual interference

and counter spot jamming. Limiting illuminator use to the short terminal phase greatly increases firepower.

The major component of the AEGIS system are an electronically scanning radar, the computerized command and control of the system, guidance illuminator radars, missiles and missile launchers. These components are shown in Figure III-4 and are described in short below: [8]

* Multi-function array radar AN/SPY-1

It is capable of automatic surveillance and the simultaneous detection and tracking of multiple targets. Operating under computer control it can neutralize enemy jamming and track and transmit guidance command to the SM-2 missile. Targets detected by the AN/SPY-1 radar are automatically evaluated as to threat and engageability.

* Guidance illumination radars

The most threatening targets are assigned by the AN/SPY-1 radar to the MK 90 guidance illuminator radars. The illuminators transmit radar energy, which is reflected from airborne or surface targets, enabling AEGIS missiles to home-in on the radar reflections.

* SM-2 Standard missile

The AEGIS SM-2 modularized standard missile is a semi-active terminal homing weapon with mid-course command guidance. It can effectively attack both airborne and surface targets. The missile's lethality, coupled with AEGIS fast reaction and high availability provide the needed improvement in system effectiveness.

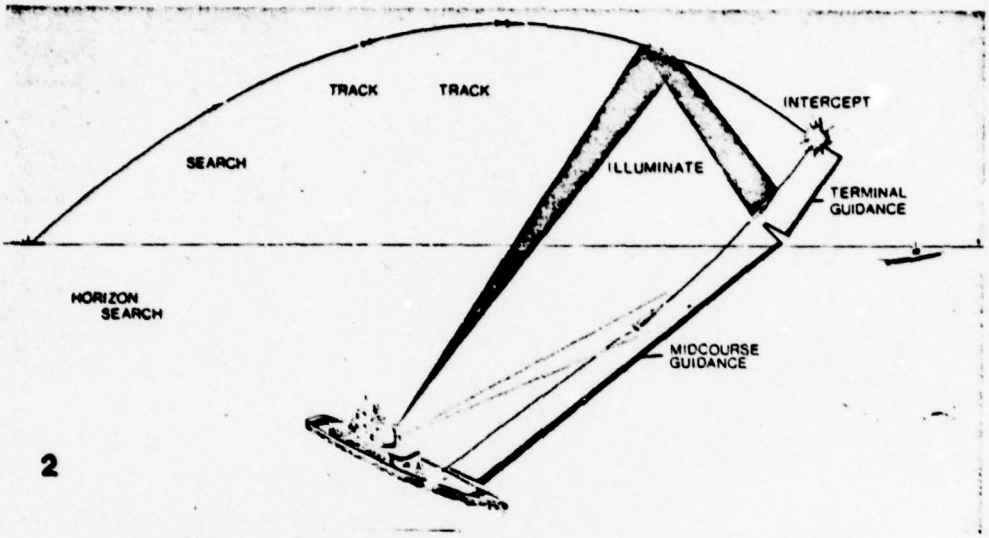


Figure III-3: Functions of the AEGIS multi-phased radar array.

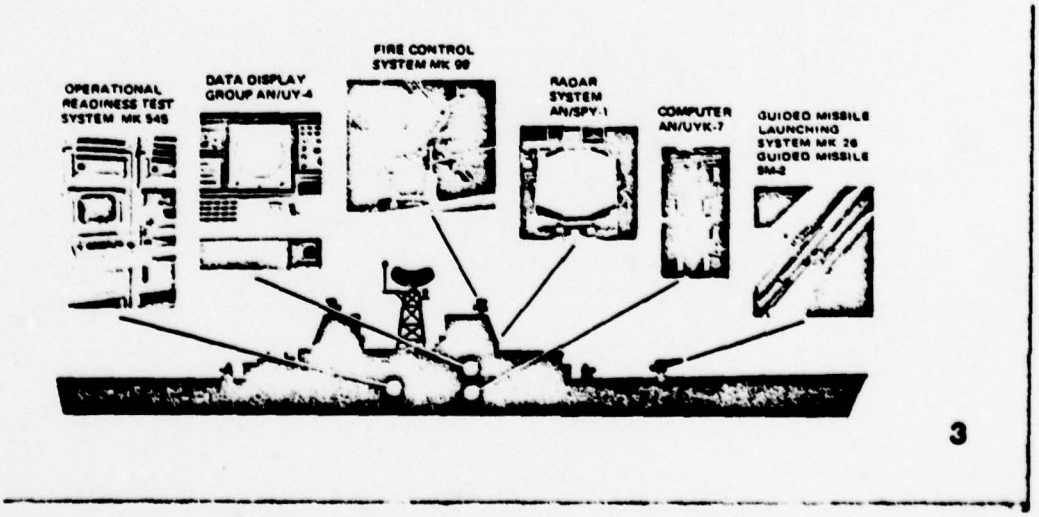


Figure III-4: Major component of the AEGIS system.

* MK 26 missile launcher

It is a fully automated and digitally controlled multipurpose quick reaction launcher. It can launch both anti-air (SM-2), anti-surface (SM-2, Harpoon) and anti-submarine (ASROC) missiles interchangeably.

* MK 542 operational readiness test system

It monitors the overall system, detecting and reporting malfunctions. Thus prompt repair is effected, or when repair must be deferred, data for operational reconfiguration to bypass the failure are provided.

* Computerized command and control of the system

The AEGIS is integrated and controlled through three multi-purpose computer groups, first the Radar Control (MK 10), second the Weapon Direction (MK 12) and third the Command and Control (MK 130). Operations in these groups are facilitated by a four bay AN/UYK-7 computer complex, AN/UYK-4 displays and associated peripheral equipment.

The component interrelationships and functional loops of the system are illustrated in Figure III-5 and are described briefly as follows: [8]

* Operational status

It is continually determined by the Operational Readiness System.

* Detection and decision

Targets enter the detection and decision loop from the AN/SPY-1 radar, other ship's sensors or data from

other vessels and aircraft. Depending on the operating mode, targets are evaluated and when threat criteria are met, assigned to weapon control for engagement. In the automatic special mode, targets meeting governing doctrine, barring over-ride, are automatically fired upon.

* Weapon control

In automatic, semi-automatic and casualty modes, weapon control inserts targets into the engagement queue and schedules equipment for launching and terminal illumination. Trial intercepts are computed and a time to fire is predicted. Positive action is required for firing. After missile launch, midcourse commands are generated using AN/SPY-1 missile and target position data. Weapon control reports kill results back to the detection and decision loop. AEGIS offers four operating mode options: automatic, special, semi-automatic and casualty.

* Manufacturers

The prime contractor for the system is RCA Government and Commercial Systems, Moorestown.

* Status

Candidate ships for installation of the AEGIS system are:

(1) Aircraft Carriers (CV)

(2) USS LONG BEACH (CGN-9) as part of her modernization.

(3) Ships of USS VIRGINIA Class (CGN-38)

(4) Nuclear-powered Cruiser Class (CGN-42)

(5) Destroyers of the DD 963 Class

(6) The new Destroyer Class (DDG-47), the leading ship of which is planned to meet the fleet in 1982.

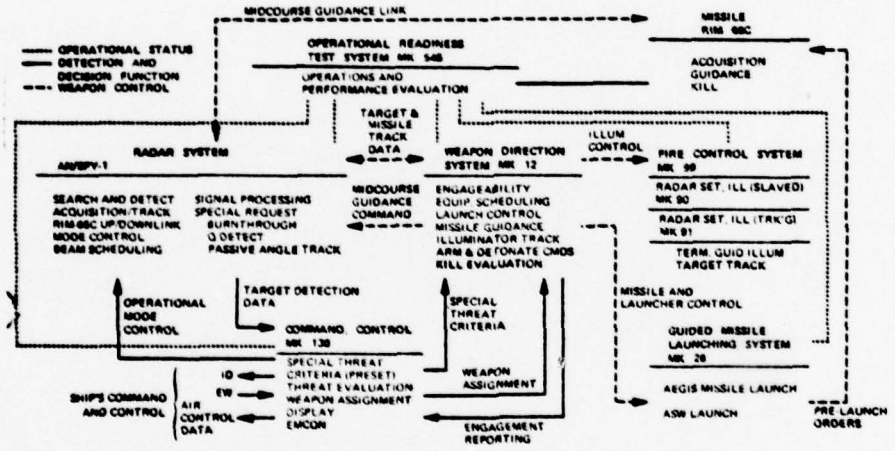


Figure III-5: Component interrelationships and functional loops of AEGIS system.

d. Shipboard Intermediate Range Combat System (SIRCS) [11,12]

SIRCS is a major defense program begun in 1975, in order to develop a total combat system, that will provide a detection-through-engagement capacity and that will be modularly adaptable to various vessels according to individual size, mission and capability constraints in the post-1985 time frame. Figure III-6 portrays the multimission, multi-platform characteristics of SIRCS.

SIRCS may be the primary combat system in lighter displacement and support vessels and will complement the longer range AAW and surface strike systems, such as AEGIS and Harpoon in major combatants. It will be fully responsive to the future anti-ship missile defense (ASMD) and surface warfare requirements of the Navy. System performance goals require a substantially greater overall capability, that presently exists in the areas of system reaction time, firepower, spatial coverage and simultaneous engagement capability.

SIRCS is one of the first programs planned for acquisition in accordance with the recently promulgated Office of Management and Budget policy on major acquisition systems (OMB Circular A-109). This approach is unique for a major weapon system in that the development plan calls for early competitive involvement on the part of the industry in the concept formulation and definition of the system, the operational requirement identified by the nature of the threat, and the desired performance, reliability and cost goals for the system. This concise statement of requirement was tailor-made to communicate a very broad, but bounded, problem, to which industry could respond with independently conceived concepts. Three major industry teams, McDonnell Douglas, Raytheon and RCA have been selected to develop independent conceptual designs for a total system, based on their own analysis of the requirements and available or emerging technology.

It was planned to select the two most promising of the three system concepts for competitive validation, commencing in late 1977. Long range plans call for initiating full-scale development with the single optimum system, beginning 1980. Fleet introduction is envisioned in mid-to-late 1980's.

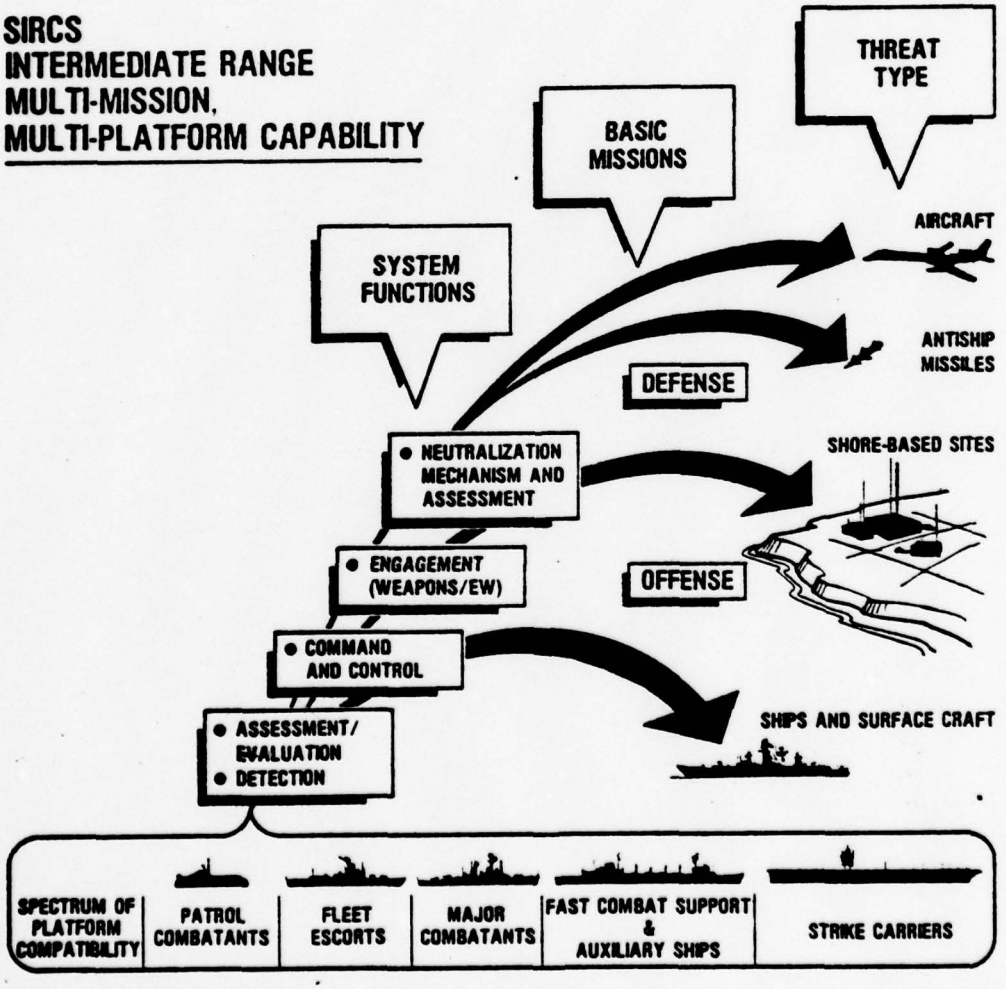


Figure III-6

2. Tactical Airborne Systems

a. Airborne Tactical Data System (ATDS) [4]

ATDS consists of the E-2C aircraft, the latest version of an aircraft to be designed initially for the airborne picket mission, a qualified crew and an extensive integrated system of various electronic equipment.

The ATDS electronic system is divided into four primary subsystems as illustrated in Figure III-7 and as described briefly below:

* Detection subsystem

Consists of the various radar raw video inputs of reflected energy and a computer detector unit. The computer detector determines when an actual target is detected, computes its location and prepares or stores the data for the data processing and display subsystems.

* Communication subsystem

It is the interface of the ATDS system with other TDS units and controlled aircraft. Its primary function is to transfer and receive binary coded data.

* Navigation subsystem

It is comprised of three independent but integrated navigation subsystems: an inertial set, a doppler set and an air data computer and compass subsystem. In addition to an accurate geographic reference for the TDS link the navigation subsystem enables the ATDS to establish an accurate data base for its computational data.

* Data Processing and Display subsystem

It consists basically of a computer and three cathode ray tube display units for the crew. Its function is to assist in accomplishing area surveillance, threat recognition and evaluation, control of aircraft and transfer of data to and from other TDS units.

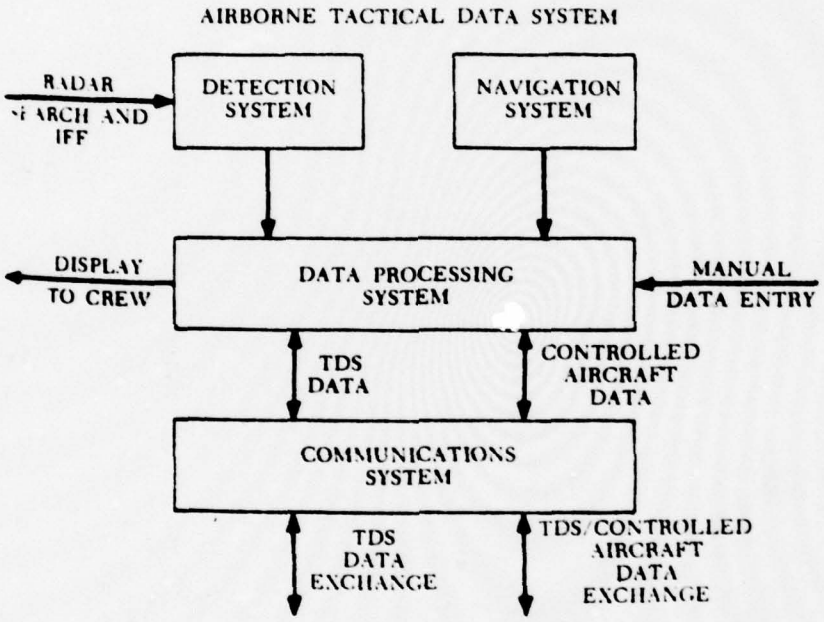


Figure III-7: Airborne Tactical Data System

The digital computer is a general purpose dual processor, expandable memory, solid state computer. In normal operation the computer functions as a dual processor computer, with the program functions shared between the two processors. In the event of failure of one processor, the remaining processor is capable of performing all the functions with no loss of tactical capability.

The computer programs (software are modularly constructed and are divided into three classes: pre-operational test (POT), tactical, and fault isolation program (FIP). The function of POT is to provide pre-launch indication of computer status. The tactical program contains the programs required for operations and for monitoring of hardware. FIP is used to isolate a fault discovered through POT or the tactical program in a particular replaceable assembly. Each of the three programs consists of several subprograms, which perform specific functions.

3. Tactical Antisubmarine Warfare Aircraft Systems

The systems of two existing ASW aircraft are examined in this section, that is of the long range P-3C Orion (Shore-based) and of the short range S-3A Viking (normally aircraft carrier-based).

a. ANEW: The P-3C Orion Data System [4]

The basic functions of the system are:

* Classification

A tentative classification of a newly-detected target can be displayed by the system using stored data, including a pre-mission tape prepared on the ground and data coming in via a digital data link.

* Navigation

Aircraft position, sonobuoy positions and expected submarine positions are continuously updated by the computer system with the aid of inertial navigation and other data.

Additionally tactical maneuvers can be directed by the system, resulting in increased accuracy over all previous systems.

* Tactics

Depending on the tactical situation, the system is able to display alternative employment of sensors, sonobuoys and armament, together with a calculated probability of success for each tactics and keep record of the expendables, as they are used.

* Data Recording

The system automatically records a running history of the flight on a magnetic tape. This tape can be replayed on the ground after landing, to assist in postflight debriefing, further analysis of the data gathered and briefing of succeeding crews. It can also be used as a training aid for flight crews.

Complementing the system of the aircraft and providing support from the ground, is the Tactical Support Center (TSC), a computerized facility at the parent base. Using the same computer as the aircraft, the TCS is able to prepare mission tapes for the crew, with information on possible targets and their specifics, as well as data from previous flights. A digital data link permits the TSC to receive data from an on-station aircraft, and to transmit new information. This link is an extension of the link used by the P-3Cs to exchange data when relieving each other on the scene of an ASW action.

The software modules available in the TSC system are illustrated in Figure III-8.

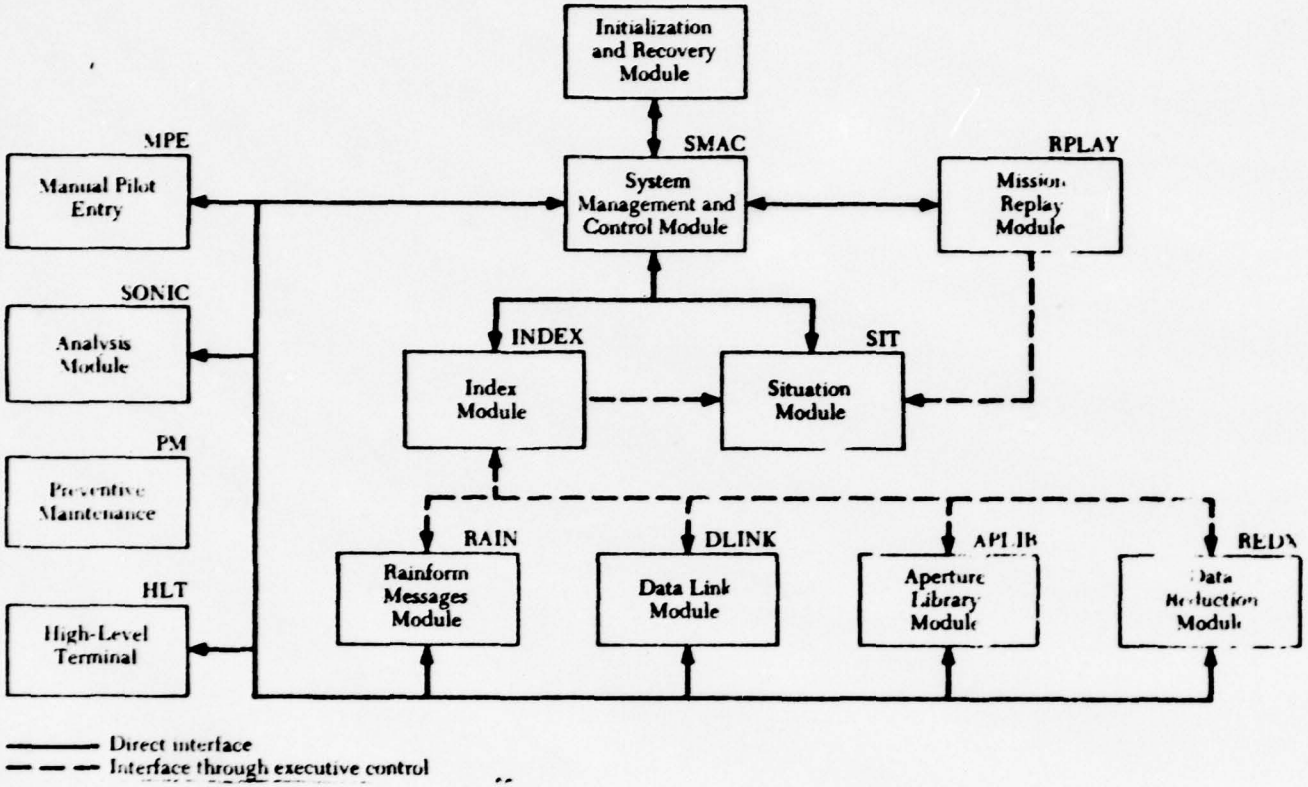


Figure III-8: Tactical Support Center Software

The TSC is linked with other TSCs and the Fleet Commanders through the Force High Level Terminal for strategic level functions.

b. The S-3A Viking Data System [4]

It is the latest development in airborne automated ASW data systems and carries with it the experience of the ANEW system.

The S-3A is designed to accomplish the same ASW mission with a crew of four, that the P-3C does with a crew of ten.

The weapon system software to support the S-3A is illustrated in Figure III-9. Support software is also that software in the TSC, which prepares a mission tape and inserts data for the new crew and which processes data from the post-flight tape. The TSC and its functions remain the same, except that the TSC is on board an aircraft carrier. Instead of the High Level Terminal there is an interface with the NTDS system, so that the complete tactical picture can be presented to the Task Force Commander on one unit.

Mission software regards several categories. A general purpose computer is supported by a number of smaller processors, physically integrated into the sensor system and into other systems. Some of these processors have functions as described below:

* The acoustic data processor software translates the acoustic signal received from sonobuoys to digital

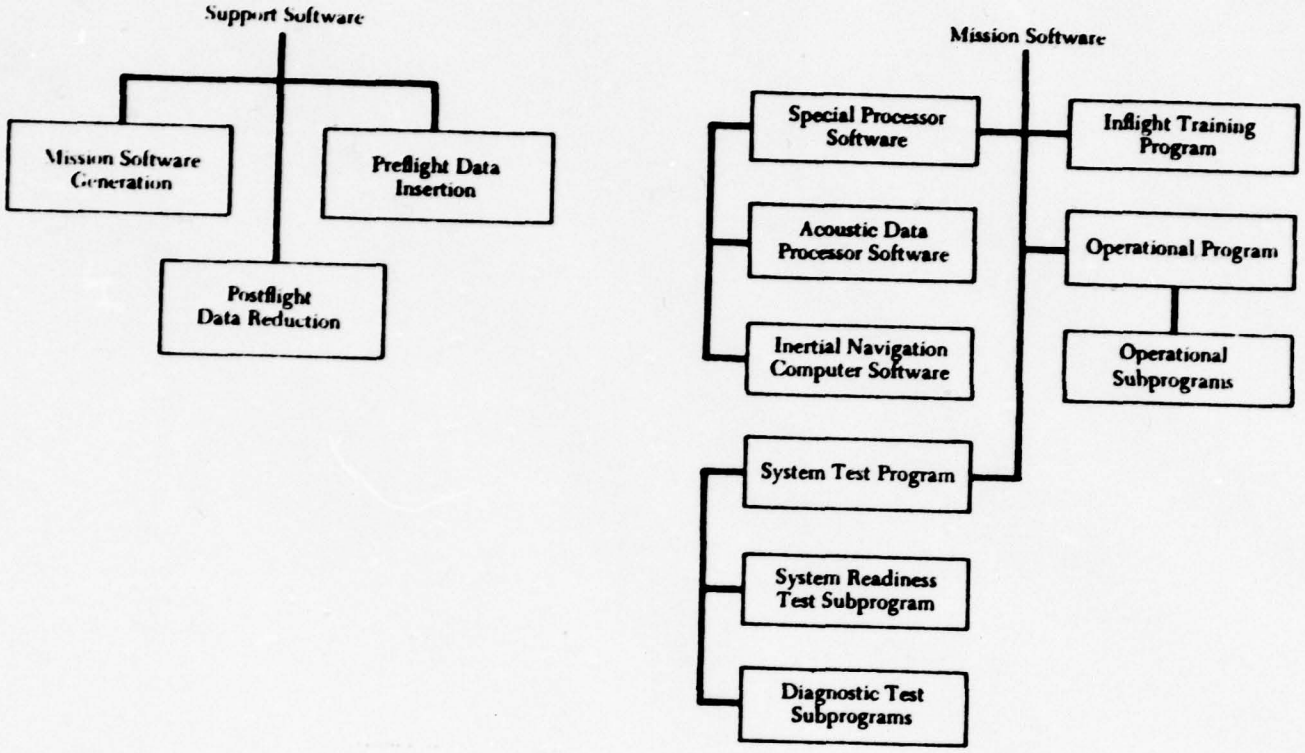


Figure III-9: S-3A weapon system software.

data, which then are put into the system for inclusion in the complete tactical picture.

* The inertial navigation processor software keeps track of the aircraft position and from this reference point provides data for the location of any object or other point on the operators display.

* Special processor software programs provide for activation of systems, which do not have their own dedicated computers.

* The system readiness subprograms provide the crew with a go/no go check on the various avionics systems before starting out.

* The diagnostic test subprograms are used, in case of a system failure to isolate and identify a faulty weapon replaceable assembly. In this way the maintenance of the aircraft and its system is greatly accelerated and simplified.

* The inflight training program is exploited for the training of the crew, using simulated data.

* The operational program is the tool by which the four-man crew manages the ASW function of the aircraft. It receives, records, processes, analyses, correlates and displays tactical data. It accomplishes this with the aid of 21 subprograms shown in Figure III-10 and is controlled through the complete processing systems illustrated in Figure III-11.

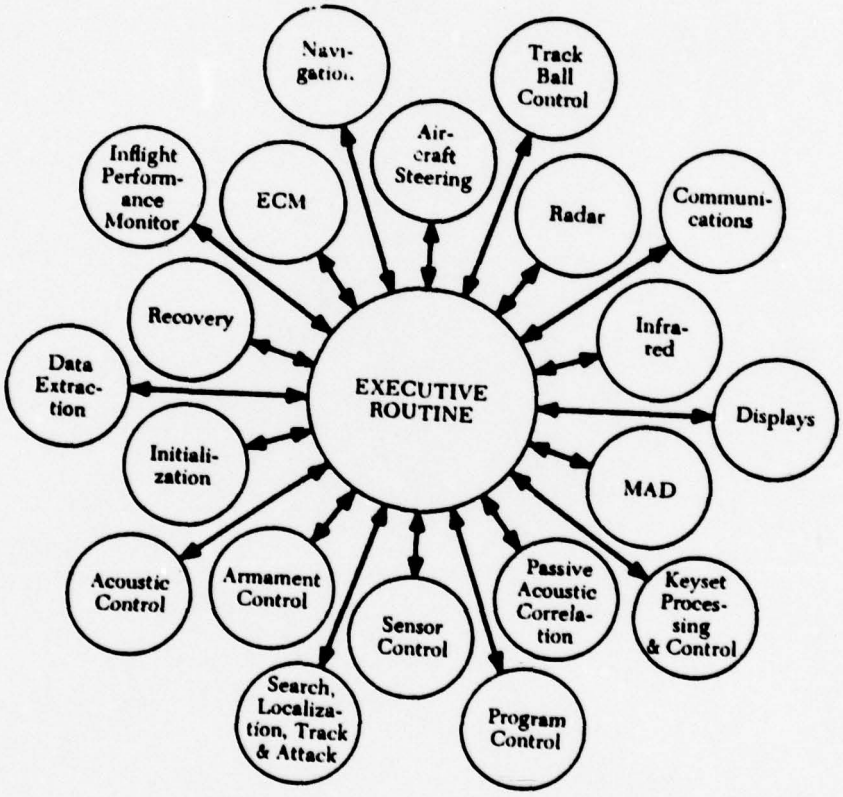


Figure III-10: S-3A operational subprograms.

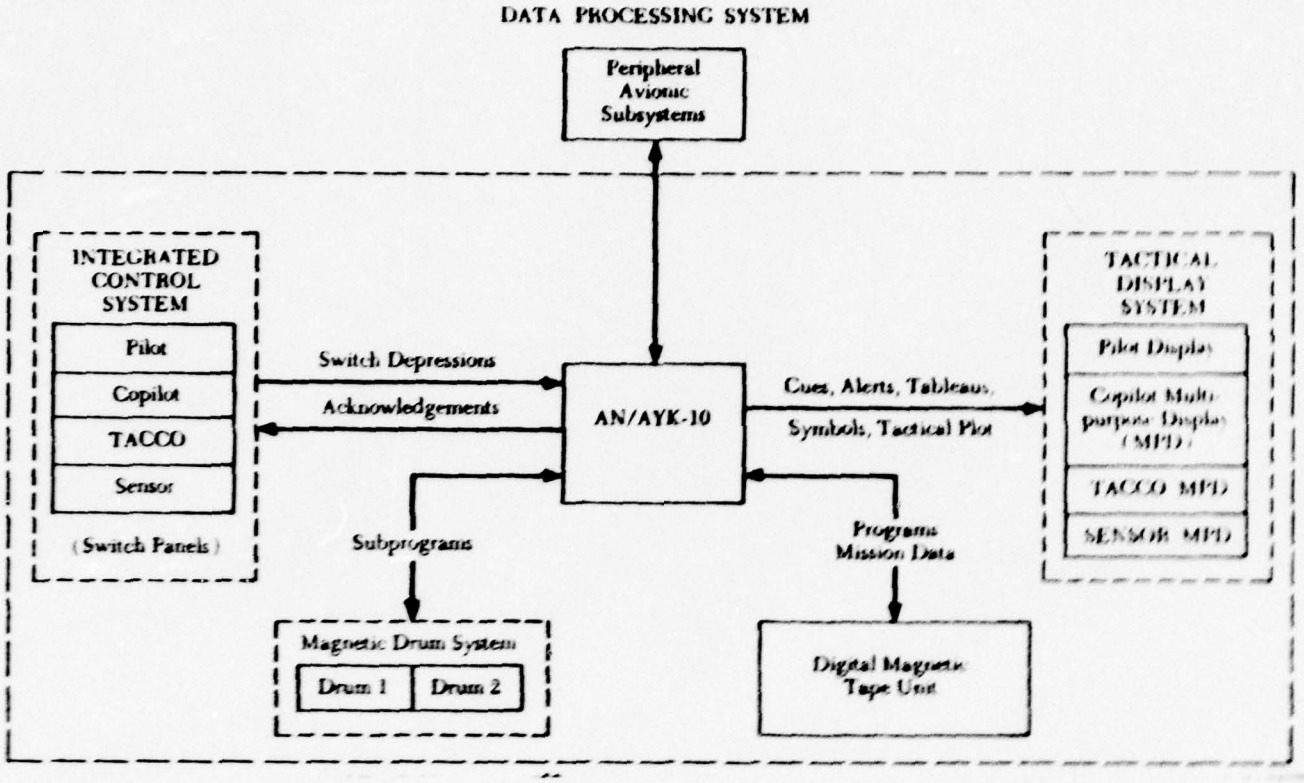


Figure III-11: S-3A data processing system.

Maintenance of the systems software is the responsibility of the Navy.

4. Strategic Systems

a. Amphibious Flag Data System (AFDS) [4]

The AFDS is a command and control system designed to support the Amphibious Operation, which by the very nature of its scope is strategic. However since the landing phase of the operation is tactical, the system encompasses tactical capabilities also.

The system, as it is installed in the LCC amphibious command ship class, is the combination of an Integrated Operational Intelligence Center (IOIC), the Amphibious Support Information System (ASIS) and a Naval Tactical Data System, which are described briefly below:

(1) Integrated Operations Intelligence Center (IOIC). It provides intelligence support for all commanders embarked on the amphibious ship. It communicates automatically with the ASIS.

(2) Amphibious Support Information System (ASIS). This is the strategic component of the AFDS and it is built on NTDS hardware. The basic configuration is portrayed in Figure III-12.

The system is an on-line, general search and retrieval system operating on a large data base. It provides users with the capability of creating, modifying and deleting files, as the amphibious operation progresses. The programming

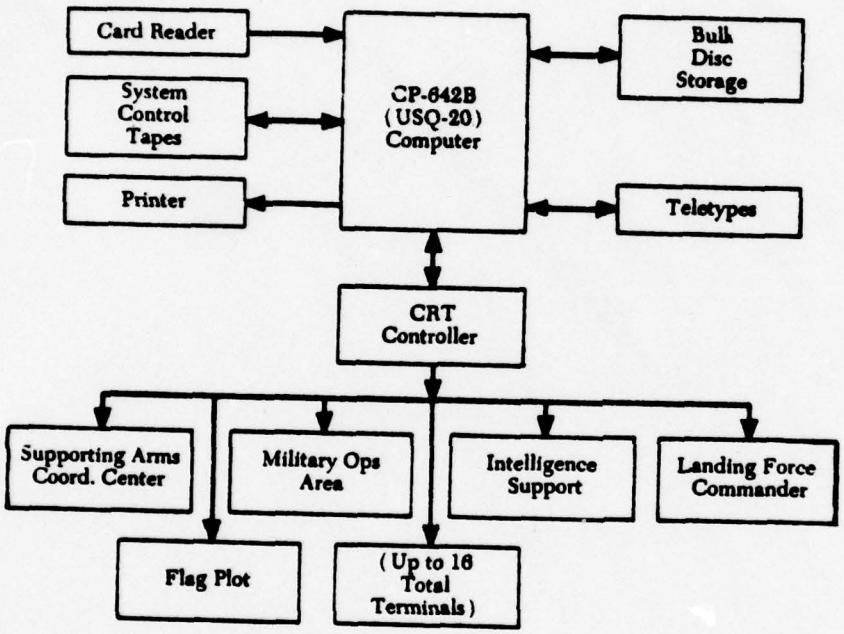


Figure III-12: Amphibious Support Information System (ASIS) hardware.

language is QUEST (acronym for Query, Update, Entry, Search, Timeshare). The files carried include data necessary to support the basic phases of the operation, such as:

* Loading data for the Force
* Landing plan
* Shore fire bombardment/air target file
* Bomb damage assessment file
* Intelligence support file

Based on intelligence received from IOIC the ASIS constructs the Amphibious Plan and continues its function right through the landing phase. With the execution of the landing phase the files of the ASIS are revised as necessary by the various users in accordance with the evolution of the battle ashore.

(3) Naval Tactical Data Systems (NTDS). This is a configuration of the already examined NTDS, suitably tailored for the command landing ship.

It can basically provide:

* Air Defense capability
* An air traffic control function called Objective Area Flight Coordination. This module permits control and coordination of helicopter assault and supply operations, close-in support and air defense.

The combination of the IOIC/ASIS/NTDS provides to all users with a capability to have access to the same up-to-date

operational picture, resulting in the elimination of errors and misunderstanding, which occur when plans or schedules change during an operation.

b. Antisubmarine Warfare Center Command and Control System (ASWCCS) [4]

The function of ASWCCS, formerly a function of the ASW Force Commander ashore, has been absorbed by the Fleet Commander ashore (Commanders Second and Third Fleets), who is participating in the Worldwide Military Command and Control System (WWMCCS).

With the completion of the system, the Tactical Support Centers (TSCs), which are, as already mentioned, computerized facilities at the parent ASW aircraft bases, will be able to exchange information rapidly through the force high level terminal as well as to interrogate the historical files of the ASWCCS, to assist in analyses of the on-going missions and their results.

B. JOINT COMMAND AND CONTROL SYSTEMS

1. Tactical Systems

a. Joint Tactical Information Distribution System (JTIDS) [13]

JTIDS is a high-capacity, time-division multiple access information distribution system providing integrated communications, navigation and identification capabilities. It is being developed to facilitate secure, flexible and jam-resistant information transfer in real-time among the dispersed

and mobile units characteristic of modern armed forces. The most significant and unique characteristic of JTIDS is the system architecture, that simultaneously interconnects all participants. The system can be considered to constitute a pool of information, which is continuously updated by each participant and which can be tapped to the degree desired by any participant.

The basic JTIDS building block is a single communications circuit, that simultaneously services several users. The capacity of the circuit is shared among the participants on the basis of time division, using a technique known as time-division multiple-access (TDMA).

Each participant is equipped with synchronized clock, and is assigned a sufficient number of timeslots to accommodate the number of messages likely to be required by his mission. During his assigned transmit timeslots, each user broadcasts data into a commonly accessible communications data stream, represented by the ring in Figure III-13. All other elements can extract information of the type, they require, by continuously monitoring and sampling the data base. Digital processing provides each participant with selective access to all of the information generated by the other elements by applying fixed and variable filters to incoming messages. The user does not have to request information from a specific party, or wait until he is notified of information

important to his mission; instead he decides what category of data he wants - such as hostile aircraft within a 50-mile range - and he will receive everything the system has in that category.

The typical participation in the JTIDS is illustrated in Figure III-12.

The system is designed to have the highest possible survivability. If a station fails or is knocked out, the loss is confined to the loss of that particular station's information or capability, wihtout crippling the overall net capability.

Since JTIDS operates at microwave frequencies, direct communications are limited to line-of-sight (LOS). Coverage beyond LOS must be provided by relay stations. Any aircraft with a JTIDS terminal can be assigned the relay role and assume it in a few seconds.

JTIDS has no dedicated switching centers or land-lines; the system consists wholly of the deployed JTIDS terminals. To operate within it, each tactical element need only bring a JTIDS into the environment.

Approximately 20 multiple nets can be operated simultaneously in the same geographical area and in the same frequency band, by using code division techniques. Jam resistance is obtained through the use of several techniques, including spread-spectrum modulation.

The JTIDS relative navigation capability places each element in the system in a precise location with respect to

other participants. It may not be necessary to relate this common grid to map coordinates. If required, however, such systems as LORAN and the Global Positioning System (GPS), could serve as the basis for relating the JTIDS grid to map coordinates.

When it enters service use, JTIDS will provide tactical commanders with powerful new capabilities, having the following significantly features:

* A high-capacity, secure, jam protected digital communications primary net.

* Additional nets for specified tactical functions.

* A secure digitized voice capability.

* A jam-protected relative navigation capability, that enables each unit to locate itself relative to other participants.

* Enhanced interoperability among intelligence, command and control and mission execution elements.

* Increased incorporation of IFF, TACAN, missile guidance, and RVP guidance functions; increased flexibility in structuring nets; and low probability, that the enemy will direction-find the signal source.

* A measure of relief for crowded condition in the electro-magnetic spectrum.

JTIDS will be initially implemented in the E-3A aircraft of the Airborne Warning and Control System (AWACS) and will be extended to other large aircraft, fighter aircraft,

tactical ships and other elements. In the case of the NTDS equipped platforms, Link 11 and 4A communications will also be transmitted via JTIDS. The JTIDS development program is constrained to be interoperable with, and essentially transparent to, existing systems, i.e. there are no changes required to utilize JTIDS. There will be no operator-noticeable changes in link procedures; however, the hardware and system implications have not all been assessed as yet.

The Naval Ship Engineering Center under NAVSEASYSCOM is working with NAVELEXSYSCOM, Naval Electronic Laboratory Center San Diego and the Joint Program Office, Hanscom AFB, to define the requirements for interfacing NTDS and JTIDS, and for integrating JTIDS into shipboard combat systems.

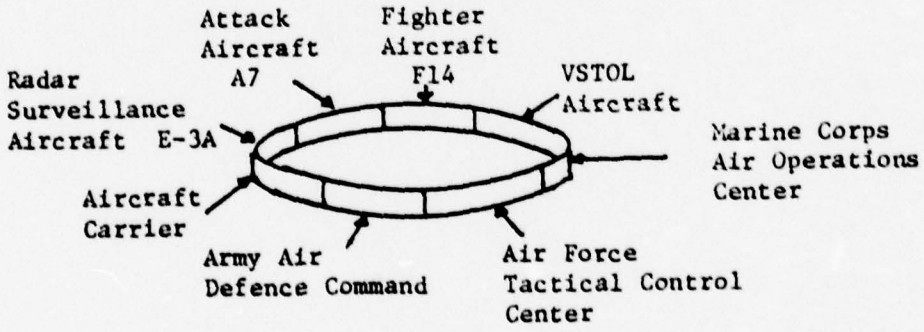


Figure III-13: Typical participation in the JTIDS.

4. Strategic Systems

a. Worldwide Military Command and Control System (WWMCCS) [14-17]

* General Description

WWMCCS is a complex and continuously evolving Department of Defense (DoD) project designed to provide the National Command Authorities (NCA), comprised of the President and Secretary of Defense, with optimal capability to command and control the Military Forces in a crisis or war situation.

Conceptually, WWMCCS is a Management Information System (MIS), whose basic components are sensors, data storage and processing equipment, input and output devices, communication links, intelligence, information, people, logistics and previously established plans. These components are integrated into a dynamic, real-time C^3 system, in order to effectively respond to the demands of the environment in which the system operates, so that it can influence the decision maker, when he is facing a crisis.

Responsibilities within the WWMCCS ADP program have been assigned as follows:

* The Joint Chiefs of Staff (JCS) have the overall management, centralized planning and direction of the program under the Secretary of Defense.

* The WWMCCS Council provides liaison, in order to increase organizational coordination between the Office of Secretary of Defense (OSD) and the JCS with regard to WWMCCS matter.

* The Joint Technical Support Activity (JTSA), a field activity of the Defense Communication Agency (DCA), provides centralized technical support.

* The Air Force, acting as the Executive Agent provides support of common ADP training and logistics support, receiving guidance from the WWMCCS managers. Figure III-14 illustrates the described WWMCCS program organization.

* Major Hardware/Software Overview of WWMCCS

Starting in 1966, a plan was developed to acquire a set of standard computers and programs to support the WWMCCS. The selection of Honeywell Information Systems as the winning bidder was announced in 1971. Honeywell was to provide 35 off-the-shelf 6000 series computers for the various sites of the WWMCCS.

Figure III-15 shows the major functional hardware and distinguishes between the three major hardware classification: Force Control (FC), General Staff Support Large (GSS/L), General Staff Support Medium (GSS/M), utilized throughout the WWMCCS network. All systems have the capability for multiprogramming, GSS/L and FC classes can utilize multiprocessing and can be classified as general purpose, medium to large scale processing systems capable of both scientific and business oriented operation.

* Central Processor Units (CPU)

The specific CPUs utilized in WWMCCS are HIS Models 6080. The major characteristics of these units are summarized in Appendix E.

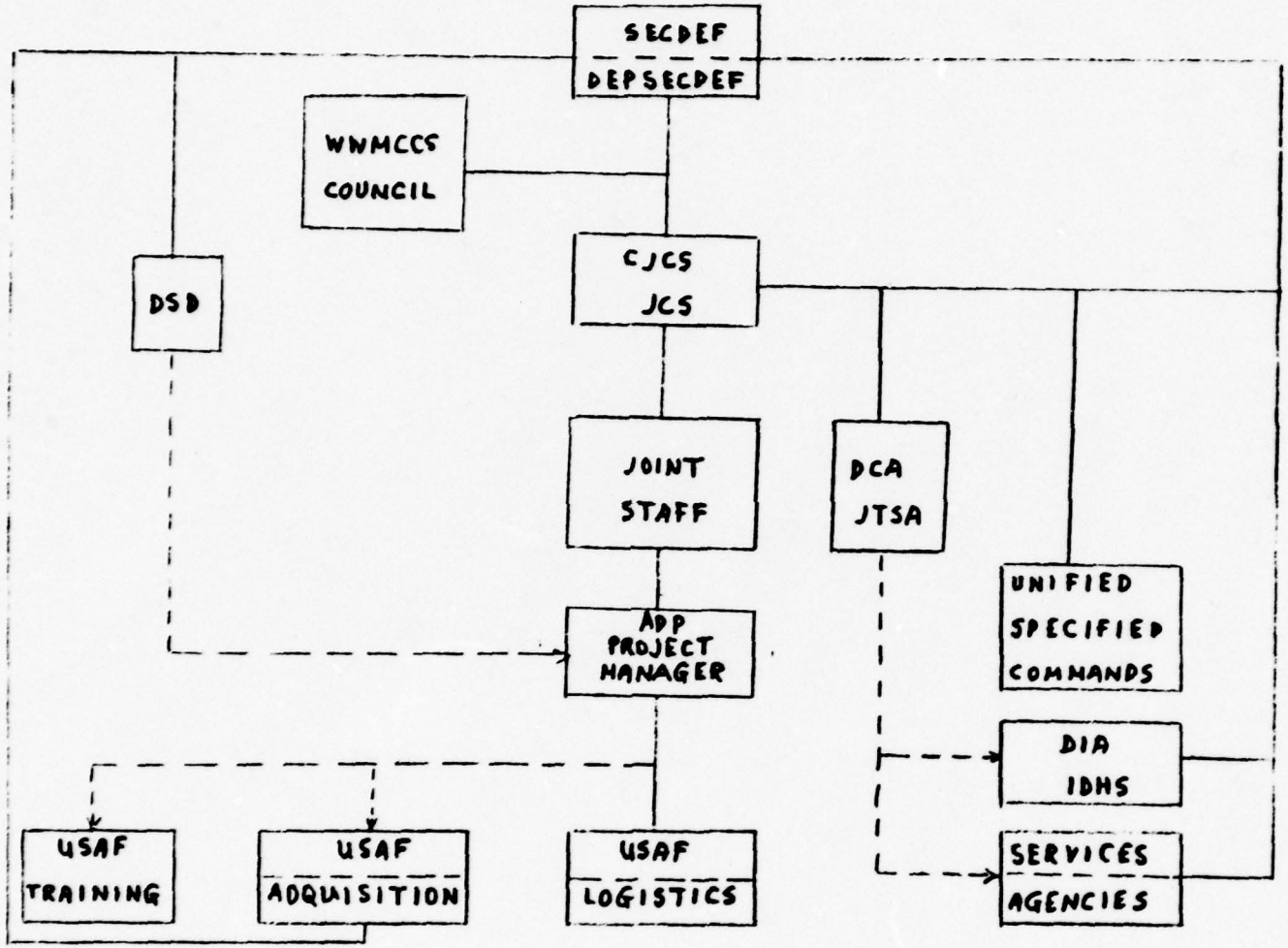


Figure III-14: WWMCCS management organization.

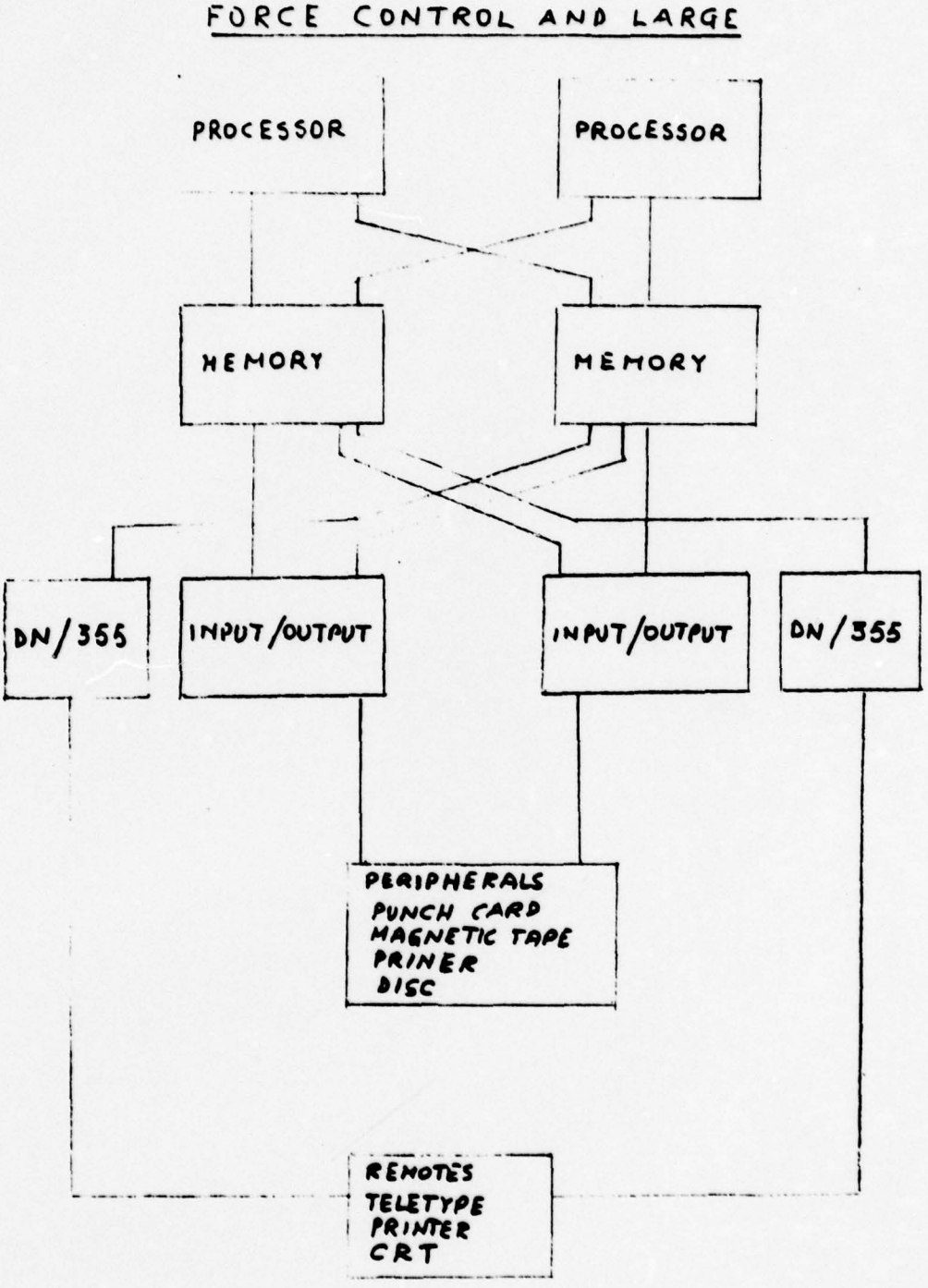


Figure III-15: Major functional hardware force control and GSS/L.

Remark: A medium system has only one set of processors.

The percentage of memory dedicated to system software and in application programs is about 70 and 30 respectively.

* Communications Network Processor

The HIS Datanet 355 (HIS/DN 355) communications processor is being utilized. Major characteristics are listed in Appendix C.

* Operating System (OS), Timesharing (TS) and Applications Software.

- Operating System (OS)

The General Comprehensive Operating System (GCOS III) is the OS developed by Honeywell for use in the 6000 series computers. Provided with GCOS III are COBOL, FORTRAN and SIMSCRIPT compilers.

- Time-Sharing (TS)

The time-sharing system software, known as the Worldwide Data Management (WWDMS), was designed and developed by HIS specifically for WWMCCS use. It is a terminal oriented system, which includes the functions data base creation and maintenance, retrieval of information and report generation. Since numerous requirements cannot be anticipated during crisis situations, WWMCCS has been designed to satisfy unpredictable as well as routine requirements, utilizing simple operation commands. Essentially, this means that, while normal terminal queries are addressed to specific files with standard output

formats in time of crisis, WWMCCS provides flexibility by accessing portions of multiple and outputting only the specific data requested.

- Applications Programs

The applications software includes nine subsystems listed below:

1) General War Subsystem (GWS)
2) Reconnaissance Information Subsystem (RECON)
3) Joint Operation Planning Subsystem (JOPS)
4) Current Situation Subsystem (CURSIT)
5) Exercise Plans & Analysis Division Subsystem (EP & AD)
6) Counterinsurgency and Special Activities Subsystem (C & SA)
7) Logistics Planning Support Subsystem (LPS)
8) Processing & Display Subsystem (PDS)
9) Resource Monitoring Subsystem (RMS)

* The WWMCCS Prototype Intercomputer Network (PWIN)

It is a three-node, experimental intercomputer test-bed consisting of Standard WWMCCS computers. It is used as a design tool for the systematic development of operating procedures, software and design criteria in implementation of the total interactive WWMCCS network. Figure III-16 shows the conceptual framework of the final system design.

PWIN has numerous similarities with Advanced Research Project Agency (ARPA) intercomputer network. The

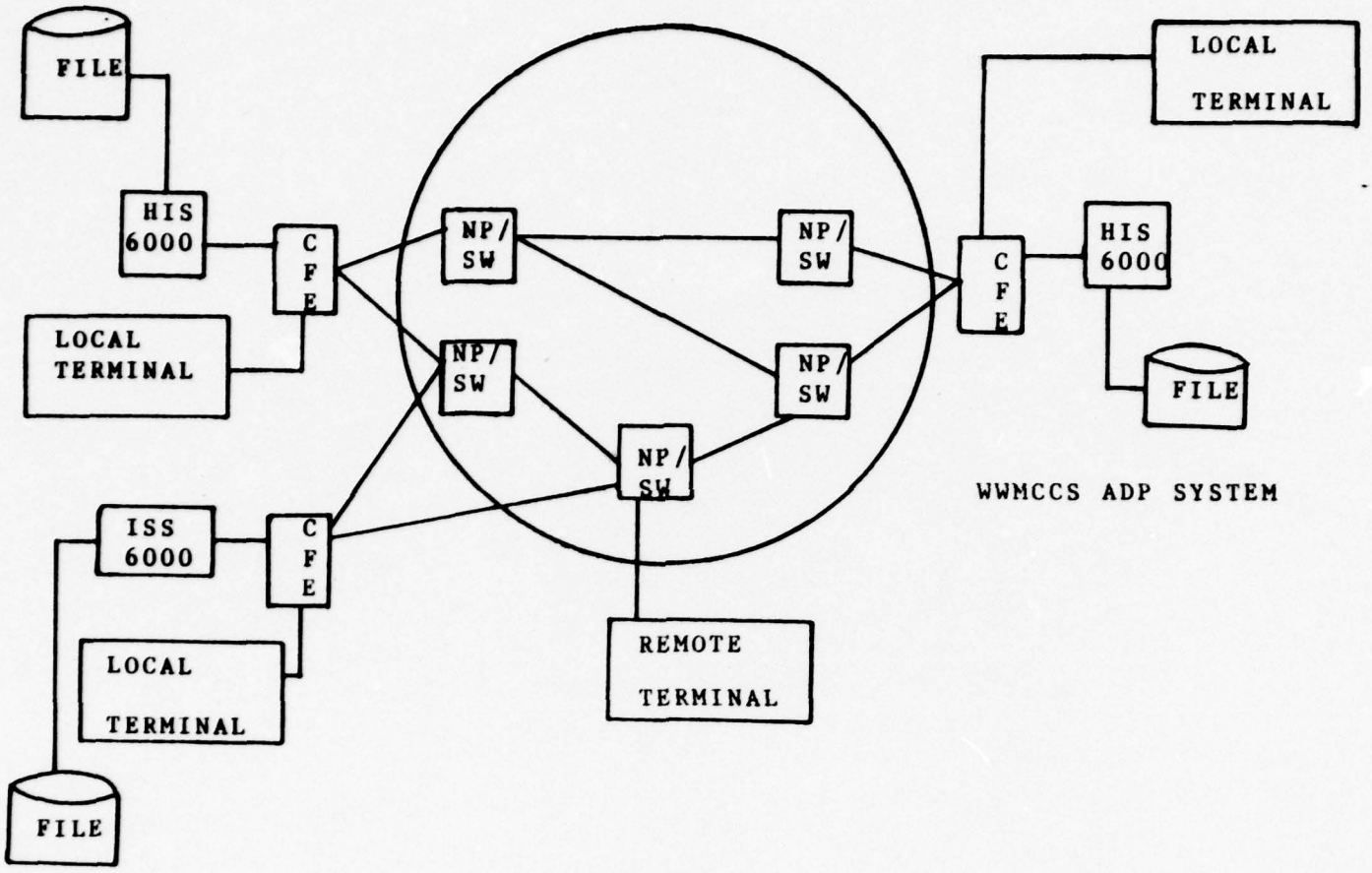


Figure III-16: A "conceptual" WWMCCS ADP Intercomputer Network.

Remarks: 1. HIS = Honeywell Information System Series 6000
2. CFE = Communications Front End
3. NP/SW = Network Processor Switch

most significant is packet switching, which indeirectly increases the system security, since the packet takes different routes depending on line availability from source to destination.

PWIN utilizes the Distributed Data Base (DBB) approach. This implies that all the information required for WWMCCS command and control is not located at every Command Center, but is distributed within the WWMCCS community in accordance with operational needs and expected usage. The inherent outcome of this utilization is increased survivability during a nuclear war.

After the operational testing of the PWIN it will be expanded to connect all WWMCCS sites.

* Navy Interface with the WWMCCS

Navy Role in WWMCCS - Naturally, the Navy participates extensively in the WWMCCS. The most important responsibilities include functions essential to command and control, such as:

- Operational Direction. Involves the employment and coordination of combatant forces.

- Resources Management. Involves the allocation of combatant forces for operational use.

The Navy interface with the WWMCCS is schematically portrayed in Figure III-17. The Navy has operational direction responsibilities shared between the Unified/Specified Commands and the Fleet Commander-in-Chief (CINC) and resources

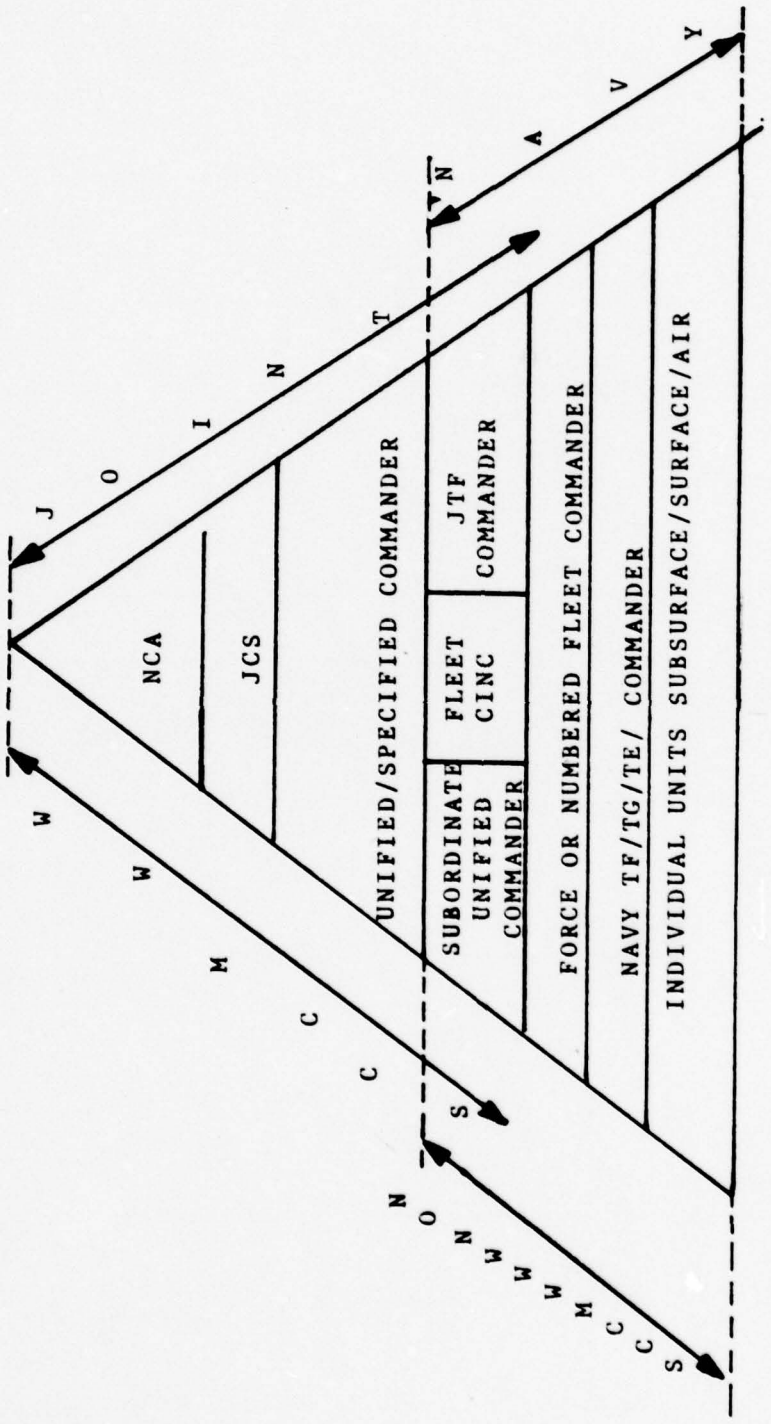


Figure III-17: Operational Chain of Command.

management responsibilities divided between Chief of Naval Operations (CNO) and National Military Command System (NMCS); and the CNO and the Fleet CINCs.

Additionally, the Navy has the following responsibilities within WWMCCS as a part of its role:

- WWMCCS support of CINCLANT and CINCPAC.
- Operation and maintenance of Navy WWMCCS Command Center.
- Telecommunications support specific dedicated networks.
- Software standardization in support of the Navy WWMCCS program.
- Funding support for WWMCCS as apportioned by DoD.
- Management information system and command and control system support for WWMCCS requirements.

* Organization Structure

Command Relationship - The WWMCCS operational chain of command is illustrated in Figure III-18. The following are notable in this schematic:

- The WWMCCS is extended from NCA down to and including the Subordinate Unified Commander/Fleet CINC/JTF Commander Level.
- All the Navy service level there are both WWMCCS and non-WWMCCS echelons of command.

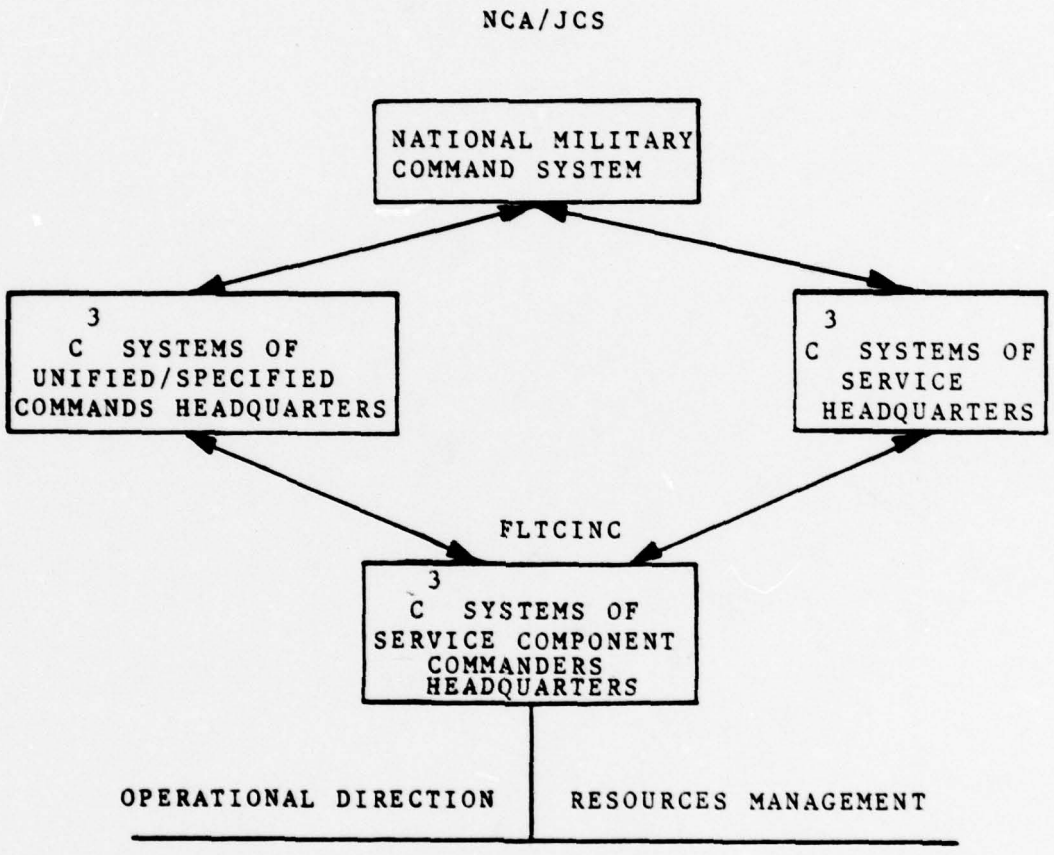


Figure III-18: WWMCCS (Navy Interface).

- The interface of the WWMCCS and non-WWMCCS activities occur at the Major Fleet Command Level.
- The WWMCCS does not include the operating forces themselves.

* Operational Responsibilities

Generally, the primary Navy operational responsibilities within WWMCCS are those which are necessary to support the needs of the NCA. There are six operations-related Command and Control functional tasks which satisfy the major operational direction capabilities required by the WWMCCS: 1) monitor the current situation, to include the status of US and non-US forces, 2) respond to warning and threat assessment, 3) employ forces and execute operations plans, 4) perform attack strike and damage assessment, 5) reconstitute and redirect forces, and 6) terminate hostilities and active operations.

* The Unified and Specified Commands

The Navy is required by DoD directives to support CINCLANT and CINCPAC and their subordinate Unified Commander. This support includes Command and Control related matters.

* Service Component Commands

The Navy's major Fleet Commanders CINCLANTFLT, CINCPACFLT, and CINCUSNAVEUR are involved in the WWMCCS in their roles as service component commanders. The command and control systems of their headquarters are known as Fleet Command Support Centers.

* Service Chiefs- Headquarter Commands

The Chief of Naval Operations (CNO) supports the National Military Command System (NMCS) through his Command Center Facility. This support comes from the CNO Management Information System and from the Navy Command Information System (NCCIS).

* Information System Support

The CNO Management Information System, though not being considered part of the operational chain of command, supports the information requirements of the CNO and the OPNAV Staff, serves as the interface with higher echelon Information Systems and functions as the information system supporting the Navy Headquarters Subsystem of the WWMCCS.

The NCCIS on the other hand is an integration of the command and control subsystems serving the CNO, the Navy component Commanders and their subordinate Commanders down to the Task Force Group level. The purpose of the NCCIS is to accumulate and synthesize the output from various single-purpose Informations Systems, even though these systems may be satisfactorily meeting their own individual requirements. NCCIS upon completion will be comprised of the following nodes: 1) the Navy Command Support Center (at CNO/HQ), 2) the Fleet Command Support Centers (at CINCLANT, CINCPAC, CINCUSNAVEUR HQs), 4) the Tactical Fleet Command Centers (afloat).

* Major Command Support Centers

(i) Fleet Command Support Centers (FCSC)

The FCSC program integrates the Operations, Intelligence and Logistics information together with Telecommunications Systems in order to provide information on a real-time basis to the Command, thereby facilitating the command and control function.

The FCSCs are in fact a full time and control system capable of satisfying the needs of the Fleet Commanders by providing them with the means to respond more efficiently to tasking from the NCA in response to crisis situations.

FCSCs are expected to perform the following functions: 1) all source data integration, 2) situation analysis, 3) resource analysis, 4) generation of plans of action, 5) resource allocation, 6) operational control, 7) operational assessment, 8) report generation and dissemination. These functions enable the major Fleet Commanders to fulfill their WWMCCS responsibilities in the areas of operational direction and resource management.

(ii) Navy Command Support Center (NCSC)

The NCSC operates as an integrated command support facility for the CNO. It can coordinate actions affecting the Navy with higher authorities in the NMCS, other agencies of the Federal Government, the other military services and intra-service matters. Since CNO does not participate in

the operational chain of command of the WWMCCS, the NCSC functions only as a coordination and as an information gathering facility.

The NCSC in fact is capable to monitor the entire spectrum of the Navy-related command and control information. In this way NCSCS provides the Navy Department Officials with a composite worldwide overview of Naval Operations.

C. LIFE CYCLE MANAGEMENT CONCEPTS [18]

1. Automatic Data Processing (ADP) Classifications

Because of the diverse requirements of systems, that utilize computer and the complex management problems associated with acquiring and developing computer resources for these systems, Department of Defense (DoD) has divided computer systems into two basic classifications:

* Embedded Computer Systems (ECS)
* Automatic Data Processing Systems (ADPS)

An ECS is embedded within and integral to weapons, communications, command and control, intelligence and air defense systems.

An ADPS is a type of a computer system used to support administrative or management functions.

The life cycle management concepts of ECSs and ADPSs are similar but differ in many important respects.

2. Life Cycle Concepts

There are different directives and chains of command that manage the life cycle of each class of ECS and ADP systems. DoD Directive 5000.29 establishes policy for the management and control of ECSs during the development, acquisition, deployment and support of major defense systems. A Management Steering Committee for Embedded Computer Resources (MSC-ECR) oversees the DoD ECS program and coordinates and integrates the ECS program into the normal defense acquisition process. The MCR-ECR reports directly to the Deputy Secretary of Defense.

DoD Directive 5100.40 governs the DoD ADP program. This directive establishes the policy for the management and control of the development, acquisition, deployment and support of ADPSs. The Assistant Secretary of Defense (Comptroller) administers the DoD ADP program.

A requirement to use computer resources, regardless of the class, must be documented as a result of a thorough system analysis. Requirements for computer resources originate in a number of ways. Normally the requirements for computer resources evolve from overall system requirements, as a result of applying system engineering disciplines. When computers are to be utilized as part of the overall solution to a system requirement, specific regulations and procedures must be followed.

ECS requirements originate in a number of ways. They may originate, for example, in master plans of commands or organizations, or as a result of specific mission or functional

analysis studies. Computer resource requirements may also originate as a result of system development efforts, which are undertaken in response to a validate Statement of Need, also known as a Required Operational Capability (ROC).

ADPS requirements are examined thoroughly to determine if a computer is the proper means to satisfy the requirement. Once an analysis of all alternatives/solutions has been made and it has been decided that an ADP solution is the best approach, the result of this analysis is documented in a Data Automation Requirement (DAR). The DAR is jointly prepared by the appropriate functional area and data automation personnel. The appropriate functional authority must sign the DAR.

Regardless of how the computer resources are documented i.e., ROC or DAR, a Program Management Directive (PMD) or Data Project Directive (DPD) will be used to provide management direction for satisfying the approved requirement. A PMD authorizes the use of embedded computer resources in major defense systems. A DPD authorizes the use of ADP computer resources. Once authority is given to use a computer to satisfy a specific requirement, configuration management (CM) procedures are used to manage the acquisition and maintenance of both hardware and software items through their life cycle. The following paragraphs discuss the life cycle phases of ECSs and ADPSs, respectively.

a. ECS Life Cycle

The weapon system acquisition process provides a basis for the life cycle management of an ECS. The life cycle until 1978 consisted of five major phases with three Defense Systems Acquisition Review Council (DSRAC) reviews. These phases along with the reviews were in the following order:

1) Concept Formulation
1a) DSARC I
2) Validation
2a) DSARC II
3) Full Scale Development
3a) DSARC III
4) Production
5) Deployment

Concept formulation is the initial planning phase. Technical, military and economic bases are established through comprehensive feasibility studies, experimental development and concept evaluation. The objective of this phase is refining proposed solution or developing alternative concepts to meet an operational capability, such as software to monitor an engine during flight. Proposed solution are refined using feasibility assessments, estimates (cost and schedule, intelligence, logistics, etc.), trade-offs and other logical analysis, such as top-down techniques. The major document resulting from this phase is the Initial System Specification, which documents total system requirements. It documents software requirements

as well as relevant design and technology constraints. Also it contains superficial definition of essential interfaces between all computer and communications equipment and personnel. This document is normally referred to in the Decision Coordinating Paper (DCP).

During the DSARC I review phase the DCP is reviewed for completeness and relevance to DoD needs. With assistance from the Management Steering Committee on Embedded Computer Resources (MSCECR) and the Cost Analysis Improvement Group (CAIG), DSARC approves or disapproves the DCP. Results are reported to the Secretary of Defense for his approval. If approval is granted, the validation phase is entered.

The validation or advanced development phase is the period in which major system characteristics are refined through studies, system engineering and preliminary equipment and computer program development, test and evaluation. The objective is to validate the chosen alternatives and to determine whether or not to proceed into the next phase. For computer resources, the major definitive documents resulting from this phase are the authenticated system specification, the preliminary development specifications containing system functional requirements for computer programs and equipment and initial Computer Resources Integrated Support Plan (CRISP). The initial preparation and coordination of the CRISP are accomplished as soon as possible to permit the program manager to accommodate appropriate CRISP provisions in the full-scale development contracts.

During the DSARC II review phase, the DSARC reviews the DCP for completeness and accuracy. The CAIG and MSC-ECR groups assist in this review. Results are reported directly to the Secretary of Defense. The DCP is updated subsequently and the development phase is started.

During the full-scale development phase, the system, equipment, computer programs, facilities, personnel subsystems, training and the principal items necessary for support are designed, fabricated, tested and evaluated. These include a system which closely approximates the production item, the documentation necessary to enter the production phase and the test results, which demonstrate, that the system to be produced will meet the stated performance requirements.

The development specifications are finalized and authenticated. Authentication of any development specification establishes the allocated baseline. Preliminary design reviews (PDRs) are held for computer program items to review the preliminary design against the respective authenticated development specification. Formal engineering change control procedures are used to prepare, propose, review, approve, implement and record engineering change to the allocated baseline.

Design of a computer resource item begins following the acceptance of a PDR for the item. This activity produces engineering documentation such as flow-charts, input output specification, and test plans. For computer programs, design specification includes documentation of logical flows,

functional sequences and relations, formats, constraints, and the data base specification. This documentation is reviewed by software engineering personnel prior to the Critical Design Review (CDR). The CDR assures that the recommended design satisfies the requirements of the development specification. Specified portions of the product specification, which will be released for coding and testing.

Development, test and evaluation and initial operational test and evaluation are conducted. Testing of computer programs (configuration items) is performed according to the formal test plans initially submitted in preliminary draft form from review at CDR, and finalized prior to the start testing. These activities normally proceed in such a way that testing of selected program function begins early during development and proceeds through successively detailed levels of assembly to the point where the complete computer program is tested.

During the DSARC II review phase, the DSARC, with assistance from CAIG and MSC-ECR, reviews the updated DCP and reports results to the Secretary of Defense for his approval. The DCP is updated to reflect all the decisions and changes up to this point in the life cycle of the system.

The production phase is the period from production approval until the last system item is delivered and accepted. The objective is to produce and deliver supportable systems to the using commands. Provisions are made in contracts and follow-on support arrangements to maintain the currency of the

equipment/computer program configuration and associated documentation in accordance with standards. Failure to consider these provisions may result in support complications, obsolete documentation and costly "modernization" programs.

The deployment phase commences with delivery of the first operational unit and terminates when the system is removed from the operational inventory. Operational test and evaluation is performed on all operational configuration items to assess the system operational effectiveness and suitability in a deployed configuration. The CRISP continues to be an active document during this phase. It is the basic agreement between the using and supporting commands for managing the computer resource. After a system is in operational use, changes to computer programs may be necessary to remove program errors, improve coding or operation, adapt to changes in system requirements, or incorporate knowledge gained from operational use. Once deployed, ECS management integrates almost inseparably into the system's management.

b. ADPS Life Cycle

The life cycle of ADPS consists of five phases:

1) Conceptual
2) Definition
3) Development
4) Test
5) Operation

The activities performed during the conceptual phase are: identifying the operational requirements, defining initial system concepts, conducting system feasibility studies, performing system requirements analysis, and determining the design requirements and gross system functions.

The function and system requirements are determined by the user with data automation support. The system requirements are first documented formally in a Data Automation Requirement (DAR). Review and approval of a DAR are conveyed in a Data Project Directive (DPD). After a DAR is prepared and a DPD is issued, a Data Project Plan (DPP) is developed to detail the development and testing required to insure that the system meets operational requirements on time. Milestones defined in the DPP delineate "go/no go" decision points in which development progress is reviewed and formal decisions made to continue, halt, or delay development of the system or a component thereof. Cost criteria are heavily weighted in the decision making process.

In the definition phase the developer defines the system's requirements. Examples of these are: interface control, expanded system requirements, computer programs, manual tasks, equipment, system/subsystem specifications, the data requirements document and the data base specification.

Analysis, design, coding, debugging, integration and development testing of computer program configuration items are done in the development phase. All activities such as

reviewing system/subsystem specification, defining preliminary program design, conducting preliminary design reviews, defining detail function designs, developing test plans, conducting critical design reviews, coding programs, testing programs, developing software validation and configuration reviews and audits are also performed in this phase.

The activities accomplished during the test phase are system testing and validation testing. The completed programs and documents are reviewed for accuracy and completeness. The system is run and reviewed by the customer to validate its responsiveness to the requirement. Validation that the system meets the specific objectives and requirements of the master plan and approval of the system for operation completes this phase.

During the operational phase, system turnover requirements are updated and the system operated on a routine basis. Routine maintenance is performed to meet changing operational requirements. All changes to the system configuration are rigidly controlled through a well defined system configuration change control procedure. As a minimum, change control documentation will contain the proposed change, the specification, analyses of functional and technical impacts and cost data. The procedure requires the approval signature of an authorized official and retention of the documentation for the system's life cycle plus four years. Periodically the master plan will be updated to reflect necessary changes resulting from ADP system reviews.

D. COMPUTER STANDARDIZATION EFFORT WITHIN THE DOD

1. Realization of the Need for Standardization

Since the successful employment of the general purpose computer in defense systems, a large number of computers have been introduced in the U.S. Forces. The Army and the Navy currently use and maintain an inventory of over one hundred computer types. Practically all of these machines have a design personality architecture, which must be catered to through specialized software and maintenance support.

This proliferation of computers has created quite a number of problems such as the following: [5]

* Small and uneconomical procurement problems, since quantity buy discounts could not be exploited.
* Untenable material and logistic support posture.
* Increased scope of personnel training requirements.
* System interface and integration problems.
* Software incompatibility.
* Proliferation of support software such as compilers libraries, simulators and debugging aids.

This problem has been felt in embedded computer application where at least 450 general purpose programming languages and (incompatible) dialects are used in DoD. [19] Whenever a computer architecture is changed, these costly software packages have to be redeveloped.

Recognition of these problems prompted the individual services as well as the DoD itself to direct their efforts towards the standardization of hardware and software of computer systems.

2. Definitions of "Standardization" and "Standard"

Standardization can be defined as the art of being able to set "standards" at the problem state with respect fo a given technology. [20]

A Standard can be defined as a specific entity, which will be used in every application, where an entity of that general description is required. [5]

3. Navy Standardization Efforts

The U.S. Navy has officially established since 1971 the AN/UYK-7 processor as the standard large shipboard computer, the AN/UYK-20 as the standard shipboard minicomputer and the CMS-2 high level language as the standard language to be employed in the production of operational programs for tactical data systems. These hardware standards, both supplied by Sperry-UNIVAC, had the advantage of commonality across multiple users in the Fleet, but they could never be acquired competitively. As a matter of fact, the Navy had some requests to deviate from the AN/UYK-7 standard. The most significant justification given was that this computer was too large and expensive ($720,000 average cost) for the intended application.

Economies of scale were realized as the following examples show:

* Economies due to large scale production

As of May 1976, 824 AN/UYK-20 data processing systems had been ordered and 637 produced. Although actual savings

realized are not known, the economies of large scale production can be seen in the volume order production for an AN/UYK-20(V) DPS basic configuration in FY 1974.

Quantity	xth Unit Cost	Estim. Learning ($S=2^{-b}$)
50	$ 25,966	---
100	$ 24,735	0.9526
150	$ 24,324	0.9596

* Cost savings in material support

It has been estimated [5] that the Navy realizes $20,000 to $40,000 per year in Integrated Logistic Support (ILS) cost savings through a standardized system like UYK-20.

* Economies due to reduction of the scope of maintenance training.

It has been estimated [5] that the Navy saves about $409,000 per year in technical training costs, through the existence of the standard UYK-20.

* Elimination of repetitive acquisition costs of unique systems.

These costs include the recurring costs for ILS, software maintenance and also the one-time development cost. The UYK-20 acquisition required $1.3 million in R&D funding for militarization of commercial hardware, support software, documentation, etc.

4. Army Standardization Efforts

The same situation has been experienced in the Army with the AN/GYK-12 and the AN/UYK-19 computers, although the Army never officially proclaimed these computers as standards.

5. Exploitation of the Emulation Technology

The advent of the emulation technology, whereby the introduced new computer emulates the previous computer's architecture, coupled with the DOD's desire to standardize without loss of competition, has led to the more recent standardication philosophy exhibited in the procurement of the Navy's AN/UYK-14 computer, which uses the latest technology, as well as the previously developed support software for the AN/UYK-20, the Army's competitive alternative to the AN/GYK-12 and the Air Force's B-1 aircraft (whose development is presently stalled in favour of cruise missile's development) alternative computer competition. As a result extensive recurring costs were saved.

6. Joint Army and Navy Efforts [21]

The Army and the Navy have jointly been engaged since 1975 in a cooperative program to develop a software-compatible family of military computers based upon a common architecture and suitable for a wide range of military land, sea and air applications. That project is known as the "Military Computer Family" (MCF) Project and the architecture to be used by the MCF is known as the "Computer Family Architecture" (CFA).

The MCF is based upon a strategy that included the following:

* Selection of architectural design or designers unbundled from implementation or implementers.

* Standardization on architecture design as the foundation on which software investment is made.

* Consideration of commercially successful architectures for which software already exists as candidates for DoD adoption.

* Technology independence, that is the anticipation of multiple implementations of the same architecture, implementations, which might differ in technology (e.g., semiconductor vs. magnetic memory), environmental specifications (e.g., volume or power constraints) or reliability assurance (e.g., MIL-qualification vs. warranties or incentives).

* Multiple sources of supply for the various processors and other modules of the family.

* Support (probably via emulation) of existing software for principal existing Army, and Navy military computers.

The first step in the development of the MCF was the selection of the architecture to be used. The selection was made by an Army/Navy CFA Selection Committee, during the period between October 1975 and August 1976, as a result of evaluating and comparing candidate architecture. The CFA committee began with the initial selection of nine candidate architectures, that is:

- BURROUGHS B6700
- IBM S/370
- INTERDATA 8/32
- GYK-12
- DEC PDP-11
- ROLM 1664
- SEL 32

- UYK-7

- UYK-20

The committee narrowed the initial set to three finalists, that is:

- IBM S/370

- DEC PDP-11

- INTERDATA 8/32

Finally the committee chose the DEC PDP-11 architecture. Technical details of this computer are provided in Appendix F.

The MCF has chosen to standardize at the instruction-set level, which presently provides the only answer to complete software transportability across a wide range of computer implementations, which has stood the test of time in industry-wide applications.

7. Standardization of Processing throughout the WWMCCS [16]

The effort to standardize processing throughout the WWMCCS, led to the Honeywell Information System off-the-shelf equipment. The essential factors for the selection of Honeywell were the quantity buy discount and reduced selection and acquisition costs. The General Services Administration (GSA) has stated that based on federal supply prices, a 70% cost saving has been attained in WWMCCS hardware and software acquisition. Furthermore, the multiyear buy, when viewed in WWMCCS evolution, has significantly eased interfacing problems. For example, in 1971 the WWMCCS community consisted of 31 data processing activities with 81 separate ADP systems. Following the Honeywell purchase

and a period of parallel operation, the WWMCCS configuration has been reduced to 46 systems, of which 35 are HIS/WWMCCS computers.

In the same effort must be included the Navy WWMCCS Software Standardization (NWSS) Program, which was developed by NAVOSTAT to standardize ADP software, in order to satisfy the requirement for timely information in support of the command and control of the Naval Forces.

8. The Common Language Effort of the DoD [19]

Digital computer systems costs inthe DoD for 1973 were estimated at $7.5 billion. About one third of this amount originated in each Service and a breakdown as to where this money was spent is as follows:

* Computer hardware : 16 per cent or $1.20 B
* Computer operations support & supplies : 38 per cent or $2.85 B
* Computer software : 45 per cent or $3.375B

Most of the software costs in the DoD go for embedded computer systems as illustrated in Figure III-19.

The programming language is the central element in the design, development and maintenance of software. The large number of programming languages (at least 450) in embedded computer systems and the lack of any widely-used languages, had many ill effects such as excessive cost, slow communication, scattered research effort, unnecessary ties to vendors, diversion from important tasks, diffused expenditures and risk in using extinct languages.

The common language effort is an attempt of the DoD, begun in 1974, to improve the quality and to reduce the cost of development and maintenance of software for embedded computer systems in DoD, by standardizing on a signle Higher Order Language (HOL).

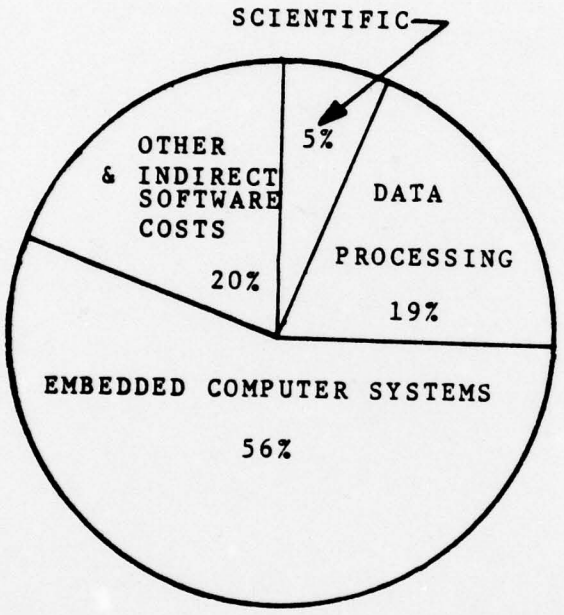


Figure III-19: Breakdown of estimated $3 billion annual DoD software costs.

Following is a rough account of the sequence of activity until now:

* Adoption of an interim list of approved languages

The DoD in an effott to reduce the number of languages used in DoD, issued the instruction 5000.31, which specifies that the only approved HOLs to be used for the development of new defense systems software are CMS-2, SPL-1, TACPOL, J3 JOVIAL, J73 JOVIAL, ANSI COBOL and ANSI FORTRAN.

* Determination of technical requirements

The characteristics of an appropriate language were developed in 1975 in the form of technical requirements, acting as constraints on an acceptable language, without specifying the details of specific language features.

* Language Evaluation

During 1976, twenty-three programming languages were evaluated against the technical requirements. Included were languages currently being used in embedded computer applications in the DoD (e.g., CMS-2, JOVIAL, SPL-1 and TACPOL), languages used for process control and similar applications in Europe (e.g., CORAL-66, LIS, LTR, PEARL and RTL-2), research languages (e.g., EUCLID, MORAL and ECL) and languages that are widely used outside the DoD (e.g., COBOL, FORTRAN, PASCAL and PL/1).

The resulting report contained the following findings:

- None of the evaluated languages satisfied the requirements in such a degree so that it could be adopted as a common language.

- All the DoD approved interim languages were found inappropriate as a basis for developing a common language.

- It was considered as currently possible to produce a single language, that would meet essentially all the requirements. Such a language was identified as a derivative of ALGOL-68, PASCAL or PL/1.

* Competitive design of a Common Language

Based on the previous results, the Services undertook a joint effort for the design of a Common Lanugage, that will be a derivative (but not necessarily upward compatible with) one of the above three languages (e.g., ALGOL-68, PASCAL, PL/1). Four parallel design efforts were begun on August 1977. All four are based on PASCAL. The contractors were CII-Honeywell Bull, Intermetrics, Softech, and SRI International.

The milestones for this competive design were:

- May 1978: Decision on 2 design competitors.
- March 1979: Decision on winner of the design competition.
- May 1979: Testing.
- October 1979: Reports due.

IV. C^3 MANAGEMENT IN OTHER NATO COUNTRIES

Following is a literature survey of Command and Control systems existing in or developed for the Navies of other NATO countries. [22]

A. CANADA

1. CCS-280 Command and Control System (DDH-280 Class)

CCS-280 is an automatic data-handling and display system fitted on DDH-280 class antisubmarine destroyers. It is based on Litton L-304 single processor version digital computer Designated AN/USQ-501(V) Data Processing Set, it comprises one digital computer, eight multifunction displays and other peripherals. Equipment was designated to meet MIL shipborne and environmental specifications.

The computer is fitted with a 40K 32-bit word memory, capable of expansion to 80K words by addition of plug-in modules to the existing processor mainframe. Increased I/O capability can be achieved by adding units to the existing data bus, using available minor channels in the I/O modules. The system includes radar signal simulation, which provides super-imposed or separate radar targets, variable in size and brightness, for training. Quick action buttons (QAB) in the displays provide mode selection and independent selection of up to seven separate radar videos, with independent offset and range 1-152 miles in binary increments. Category selection of symbology is programmable on-line, independently at each.

Basic operating software has been provided by the manufacturer. The operational software was written by personnel of the Canadian Armed Forces in a program generation facility provided by the contractor.

The operational program combines the speed and processing power of the digital computer, the comprehensive, flexible display and the automatic link equipment, into a system which provides real time coordination of sensors and weapons by the Command at the Unit and Force level.

The system has the following functions:

* Collect, process and display tactical information from all systems.
* Control aircraft and other remote surface units.
* Direct own ship's and Force weapons.
* Control ship manueuvers.
* Exchange information and orders within own ship and Force.

The system is currently operational at sea in HMC Ships Iroquois, Athabascan, Huron and Algoquin. A shore installation for training and program maintenance and generation is fitted in the Combined Support Division, Halifax.

Manufacturers: Litton systems (Canada) Limited
Collins Radio (Automatic Link Equipment)

B. DENMARK

1. Electronic Plotting (EPLO) System

The Danish Navy has acquired an advanced version of the Swedish EPLO (Electronic Plotting) System. EPLO has been designed to overcome the difficulties of manual techniques for handling the tactical situation on board small ships, such as fast patrol boats and corvettes. The level of sophistication of the system is governed by cost-effectiveness and space restrictions. The system compiles a tactical picture of an area of operations and effects data exchange with other ships and with naval radar control centers.

The operational functions performed by the system are:

* Target selection and semi-automatic tracking by means of track symbols on a true motion stabilized raw radar picture.
* Exchange of tactical data via a data link with other ships and shore radar stations.
* Target assignment to fire control systems of the ship by special assignment symbols.
* Synthetic presentation of the plotted tactical situation, including reference information such as maps, geographical grids and shore radar stations.
* Continuous recording on magnetic tape of the tactical situation.

Raw radar information is presented on a horizontal 16" PPI, whereby synthetic data including video maps, target

tracks, alignment symbols, pointer symbols and vectors is presented on a 24" vertical display.

Updating of target position is achieved in cyclic fashion and there are two available updating times: the slow cycle time of two minutes for updating 24 tracks and the fast cycle time of 30 seconds for updating six targets.

Manufacturer: Stansaab Electronik AB, Jarfalla, Sweden.

C. FRANCE

1. SENIT Naval Tactical Data Handling Systems

The Naval Tactical systems of the French Navy are known under the acronym SENIT (Systeme d'Exploitation Navale des Informations Tactiques) and five versions of this system have been identified. All of them have the broad function of gathering, coordinating and distributing sensor data and other information, presentation of information on appropriate displays and certain weapon control functions. A digital computer or computers provide the central processing facilities. The five systems differ in accordance with the nature and operational role of the vessel fitted and the sensors and armament installed.

Following is a brief description of these systems:

a. SENIT 1

Fitted on the Suffren class guided missile frigates Suffren (D602) and Duquesne (D603). A version of the system is fitted on the antiaircraft cruiser Colbert (C611).

The system employs three IBM computers.

b. SENIT 2

This system is produced in two further versions to equip vessels, whose mission is Fleet air defense and radar picket in support of the French national air defense network.

The system is fitted on the following types of ships:

* Four ships of the T47 "Surcouf" class frigates, that have been upgraded to carry the Tartar SAM system, that is the Kersaint (D622), Bouvet (D624), Dupetit Thuars (D625), Du Chayla (D630).

* Three radar picket ships of the T53R class, that is the Le Bourdonnais (D634), Tartu (D636), Jaurequiberry (D637).

* The experimental ship Duperre (A633)

The SENIT 2 employs only one computer, probably a Hughes machine.

c. SENIT 3

Intended principally for use in ships with antisubmarine and antiship missions, the system provides also quick reaction against air targets for self-defense.

It is fitted on the "Aconit" class frigates Aconit (F703), De Grasse (F612), Tourville (F610) and Duguay-Truin (F611). The ships are armed with the Malafon antisubmarine rockets, MM-38 Exocet SSMs, Crotale SAMs, 100mm AA guns, L5 antisubmarine torpedoes and can carry two WG 13 helicopters capable of use in antisubmarine role with MK 44 torpedoes or against surface targets with AS-12 missiles.

SENIT 3 employs 2 computers with expanded memory facilities. The computer equipment is provided by Hughes, in conjunction with a Hughes display subsystem which is produced under license by Thompson-CSF.

d. SENIT 4

This version is a new system now under development, employing French design and manufactured hardware.

The system is fitted on the aircraft carriers Clemenceau (R98) and Foch (R99) and the new C70 class frigates.

SENIT 4 employs the IRIS 55M computer in conjunction with ten VIZIR displays developed by Sintra. Each of these display consoles has a potential capacity of up to 130 track symbols with speed vectors, an information display with 12 lines of 18 characters, four circles, 23 lines of target vectors. The consoles are provided with tracker balls and key board data entry facilities. The synthetic information display has a character repertoire of 96 different symbols.

e. SENIT 5

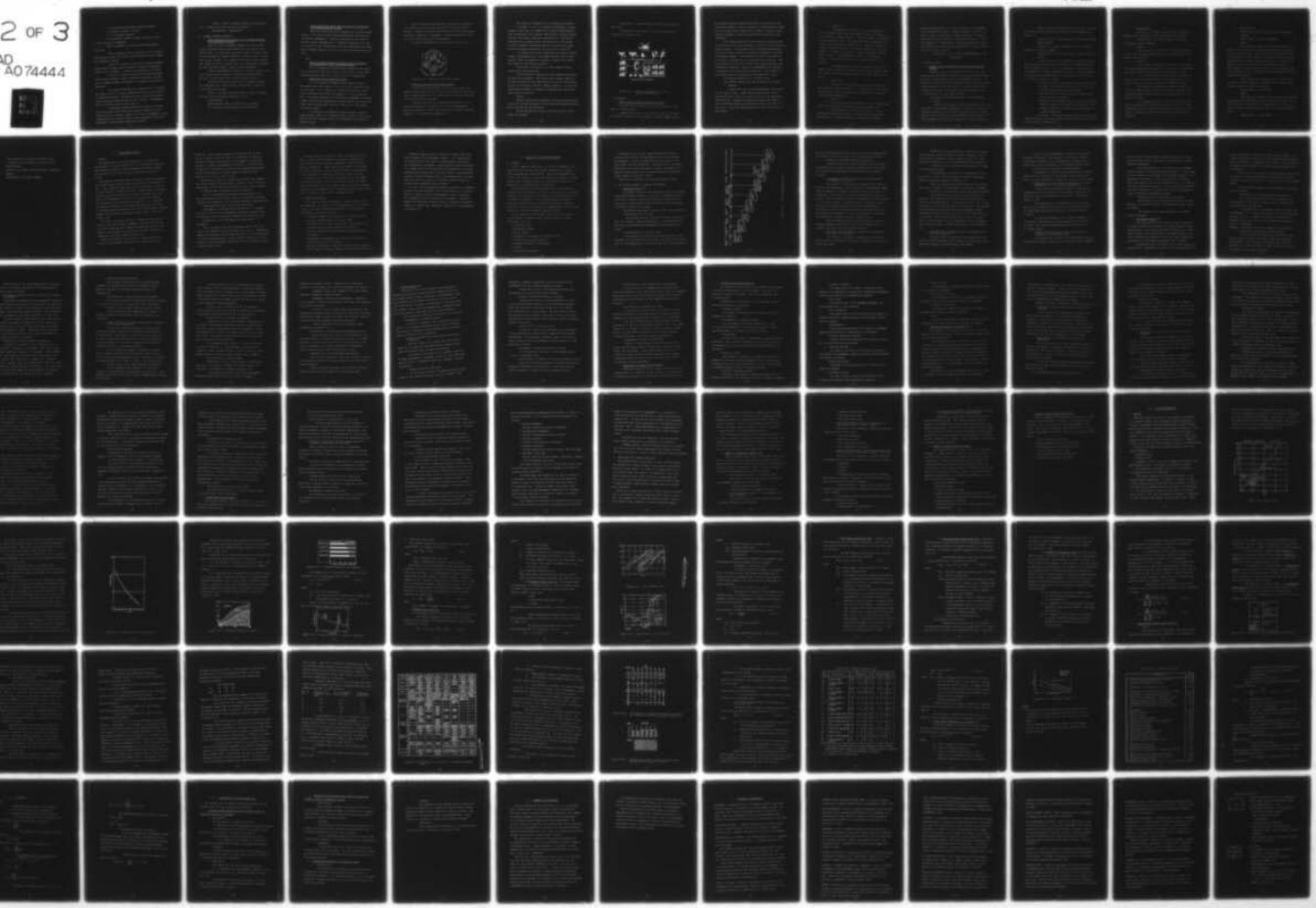
This version has been optimized as a basic system upon which a range of command and control systems, that are compatible with varying operational requirements and weapon fits, can be based. The basic system is organized around the following equipment:

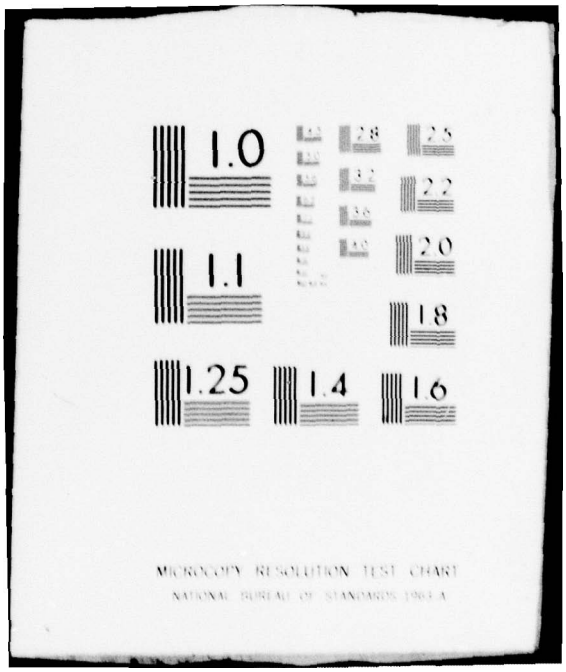
* A CII 15M computer with 64K-bytes of store
* Tape reader
* Three-function radar video processor

AD-A074 444 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2
A STUDY OF ALTERNATIVES FOR COMPUTER SYSTEMS ACQUISITION FOR TH--ETC(U)
JUN 79 V A NAOUM

UNCLASSIFIED NL

2 OF 3
AD A074444




1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

* One analog-digital/digital-analog converter

* A display interface (coupler)

* 2 to 5 display consoles

In this manner systems suitable for ships of 2000 to 6000 tons can be assembled.

SENIT 5 performs the following main operational functions:

* Data acquisition from navigation, air and surface surveillance radars, IFF, SONAR, and EW sensors, optical sights, and navigation sensors.

* Tactical situation compilation including computer-aided manual tracking and/or automatic tracking of submarine, surface and air targets. Creation and up-dating of EW tracks.

* Tactical situation exploitation including weapon system status displays, threat evaluation assistance, designation, to weapon systems, assistance in navigation and tactical maneuvers.

The display subsystem has a repertoire of 96 different symbols and up 70 tracks (symbol and velocity lead vectors), 20 additional symbols, one marker and one close control symbol can be displayed simultaneously on all consoles. Alphanumeric data are also available at each console.

The modular design permits the addition of further facilities such as ASW helicopter control, automatic IFF/SIF code extraction, computer extension for Link 11 management, intercommunication with several weapon system computers, disk store management, recording and additional consoles.

Status: SENIT 5 has been produced in series since 1978. Several countries navies have adopted it.

Manufacturer: Thompson-CSF

D. GERMANY (FEDERAL REPUBLIC)

1. AGIS (Automatisiertes Gefechts und Informationsystem fur Schnellboote) System

AGIS is a fully integrated command and fire control system, developed for the West German Navy new type 143 fast patrol boats. The principal subsystem of AGIS are radar, optical and probably EW sensors, computer, displays, data link and fire control system. The weapons of this type of ship include four single fixed MM 38 Exocet SSMs, Seal wire-guided torpedoes and two Oto-Melara 76mm/62 compact dual-purpose guns.

Display arrangements provide for 2 torpedo control consoles, a gun control console, a horizontal conference-type tactical control display and a vertical tactical plot.

The radar sensors are those of the Hollandse Signaal M27 integrated fire control system, which forms a large part of the AGIS installation.

Status: The AGIS project included 10 type 143 vessels, with the first ship being operational in 1976.

Manufacturers:

AEG-Telefunken (Main Contractor), West Germany

NV, Hollandse Signaalaparaten, the Netherlands

2. SATIR (System zur Answertung Tactischer informationen auf Raketenzerstoren) System

SATIR is the tactical data automation and display system fitted in the three Lutjens class guided missile destroyers of the German Bundesmarine. SATIR was developed in its stead to meet the German requirement and in general compliance with the B-2 Concept, a NATO standardized system for destroyers and above. Beyond theparticipation of Univac, no other reliable details are known.

E. ITALY

1. SADOC (Sistema Automatico Direzione delle Operazioni di Combattimento) Naval Processing System

It is a completely integrated data system for the execution of operations and was fitted on ships of the Italian Navy during the period 1968 to 1973 and it is still operational.

2. HYDRA Naval Data Automation System

It is an advanced integrated naval combat system for use aboard fast naval craft of 220 to 250 tons, armed with SSMs and dual guns. It is designed to coordinate the ship's operational functions and control the weapons in a fully automated mode. The system provides for simultaneous engagement of multiple targets, controlling independently medium caliber guns (typically 76mm), small caliber guns (typically 40mm) and an SSM battery (typically OtoMat).

It includes as subsystems the Dardo automatic digital weapon system, RAN-11 L/X Integrated Radar System the IPN-10 Command and Control system and the ISN-2 integrated detection and jamming system.

Data collected are processed by computer and presented on a conference display, which provides a synthetic tactical display. Computer assistance also provides for real-time threat evaluation, prediction, assessment of proposed maneuvers, overall ship's fire control and ECM management.

A typical arrangement is illustrated in Figure III-1.

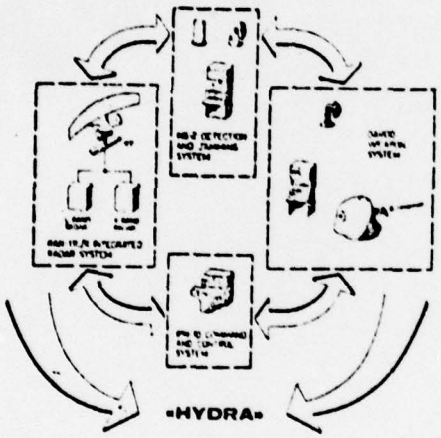


Figure III-1: Elements of the Hydra system.

3. IPN-10/20 Tactical Data Display System

IPN-10 system is designed for use on board small and medium tonnage ships, such as Corvettes, Frigates and Destroyers, where a simple system for the control of the tactical situation and of the weapons is required.

The main tasks of the system are presentation of the tactical situation, by display in a clear self-explanatory mode and automation of some of the functions, normally performed manually (e.g. tracking, vectoring etc.).

The system is comprised of the following subsystems:

* A number (1 to 6) of standard configuration display consoles, which are of a common type and are interchangeable. Their operational role includes surveillance, tracking, weapon control, ASW control, tactical evaluation etc. The display consoles can be changed during the operation of the system, so that the Combat Information Center of the ship can comply with the changing requirements of the tactical situation.

* A Central Unit, which interfaces the sensors (radars, sonars, interceptors etc.), the weapon systems (Fire Control Systems, SSMx, Antisubmarine torpedoes), medium or low speed data links and the display consoles.

Technically, the IPN-10 is a fully digital system, using solid state technology.

IPN-20 version is designed for more complex and sophisticated data management. The system uses a larger number of the same consoles and the same Central Unit as the IPN Systems as well as the Selenia CDG 3032 digital computer. In this configuration the system is capable of handling high speed data links (such as link 11).

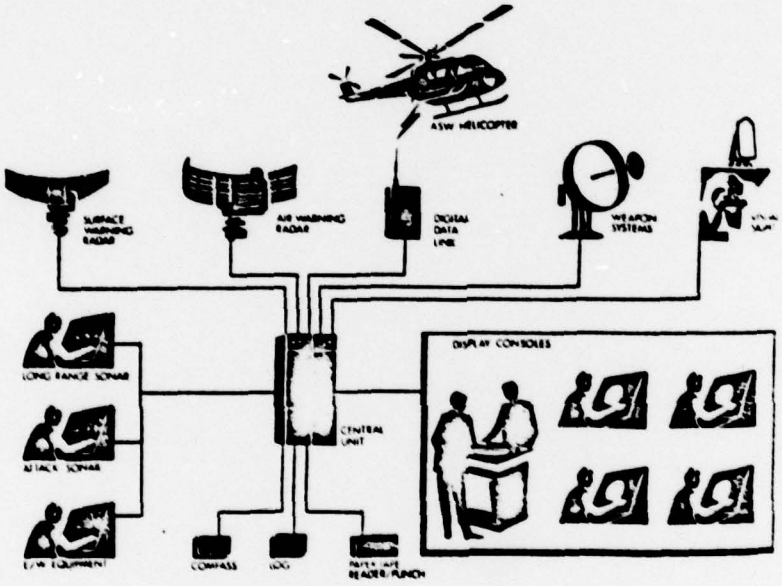
Status:

IPN-10 systems are in production and are being installed on the CNR 2400 fast frigates for Peruvian and Venezuelan Navies and on the CNR 600 tons corvettes.

IPN-20 systems are fitted on the Italian Navy "Lupo" class new frigates.

Manufacturer: Selenia-indutrie Elettroniche Associate SpA, Italy.

A schematic of the IPN-10/20 tactical data system is illustrated in Figure III-2.



Schematic diagram of IPN-10/20 tactical data system

Figure III-2: Schematic diagram of IPN-10/20 Tactical Data System.

F. NETHERLANDS

1. SEWACO (Sensor, Weapon and Control System)

The Naval Tactical systems of the Dutch Navy are known under the acronym SEWACO.

A common system, that has been built within each SEWACO, is the DAISY (Digital Action Information System) command system.

This system provides direct and indirect interfaces with the sensors and weapons. Direct interface is accomplished by means of 24-bit input/output channels and indirect interface, using digital-to-digital, analog-to-digital and digital-to-analog converters.

The display subsystem includes vertical labelled position displays and tabular displays, as well as a horizontal conference display. The displays offer interlaced presentations of both raw radar picture and synthetic information available from computer memory. The operator can specify the quantity of synthetic information to be presented on his display. Tabular displays are used for specific Tote pages providing supplementary information, computer recommendations and answers to problems put by the operation personnel. Each display is multi-functional and theoretically can perform all tasks.

The digital computer is of the SMR-S type.

Four SEWACO systems have been developed as follows:

a. SEWACO-I

Tailored for the two new guided-missile destroyers of the "Tromp" class: Tromp F-805) and De Ruyter (F-806), which were built to take the place of the two old cruisers of the Dutch Navy. Among the weapons associated with SEWACO I are the NATO Sea Sparrow point defense system and a twin 4.7" gun turret. Also the system provides for the inclusion of an SSM system.

b. SEWACO II

Designed for installation on 12 new frigates of the "Cortenaer" class, which are presently built to replace 12 old destroyers of the Dutch Navy. The 12th frigate is expected to be delivered in 1985. The system provides 12 operator consoles for the following functions 1) Air situation display, 2) Surface situation display, 3) Electronic warfare and data link control, 4) Command and control, 5) Weapon direction, 6) Helicopter control, 7) Antisubmarine warfare, 9) Hull-mounted sonar, 10) Sonar signal injector, 11) Weapon control, 12) TV console.

The armament associated with SEWACO II includes 8 Harpoon SSMs, the NATO Sea Sparrow point defense system, 1-76mm OTO Melara, MK-46 ASW homing torpedoes and 1 Helicopter (WG 13 Lynx).

c. SEWACO III

Designed for installation onboard the intermediate age frigates of the "Van speijk" or S class (based on the design of the British Leander class), which are presently undergoing large-scale modernizations.

d. SEWACO IV

This system has been designed for the new "westhinder" class of escorts under construction for the Belgian Navy.

The functions of the SEWACO IV include: 1) Air, surface and subsurface warning, 2) Compilation and display of tactical air, surface and subsurface situations, 3) Threat

evaluation and automatic air target selection for engagement by the appropriate system, 4) Weapon assignment and fire control by the WM 25 fire control system against air, surface, subsurface and shore targets, 6) Electronic Warfare, 7) Data link operation, 8) Assistance in tactical operations, including ASW helicopter direction and navigation, 9) Target simulation for training purposes.

Manufacturer: NV Hollandse Signaalaparraten, The Netherlands.

G. UNITED KINGDOM

1. ADAWS and CAAIS Automatic Data Handling and Display Systems

The Naval Tactical Systems of the UK Navy are known under the acronyms ADAWS (Action Data Automation Weapon System) and CAAIS (Computer Assisted Action Information System). The former is a full action data automation system with six versions being developed for various classes of ships of the UK Navy, whereby the latter is a cheaper and less complicated system intended rather for other smaller navies.

Following is a brief description of these systems:

a. ADAWS 1

Based on two Ferranti Poseidon computers, the system provides for the automatic compilation of the air, surface and subsurface tactical situation picture using the ship's radars, sonars, passive direction finding equipment and data link, target recommendation and target designation to ship's weapons.

ADAWS 1 has been installed since 1965 on four "County" class guided missiles destroyers (Antrim, Fife, Glamorgan and Norfolk).

Ferranti retains a responsibility for software support.

Manufacturers:

Ferranti Ltd, Digital Systems Division, Berkshire, England all digital equipment.

Plessey Radar Ltd, Surrey, England - display consoles.

b. ADAWS 2

It was the first of a new series of action data automation systems developed for the UK Navy. It uses two Ferranti FM 1600 computers with a total of 128K words of core store and the Plessey MK 8 digital autonomous displays, to provide improved facilities for the presentation of operational data in the form of a combined synthetic picture.

The display system is comprised of the following units:

* A number of labelled plan displays (LPD) used for picture compilation and weapon control.
* Two 3-man tactical displays used for appreciation of the tactical situation.
* Electronic data displays (totes), that provide alpha-numeric readouts of information in response to the operators inputs to the computer system through individual keyboards.

The ADAWS 2 system was designed for the type 82 guided missile destroyers and was installed in HMS Bristol, the only ship that was finally constructed.

Manufacturers:

Ferranti Ltd, Digital Systems Division, Berkshike, England - main contractor, digital equipment and software.

Plessey Radar Ltd, Surrey, England - Displays, interface and drive equipment.

c. ADAWS 3

This system was designed for the aircraft carrier CVA 01, but was never materialized due to cancellation of the project for that ship.

d. ADAWS 4

This system is essentially a version of the ADAWS 2 and uses the same hardware, modified as necessary to accommodate the type 42 guided missile destroyer (GMD) class of ships.

It is being produced in two slightly different versions, one for the UK Navy type 42 GMDs (eight ships) and the type 42 class (two ships) on order for the Argentine Navy.

The central data processing complex consists of a suite of Ferranti FM 1600 microcircuit digital computers, with a Plessey digital display system for the presentation of the tactical picture.

The computers provide a total of 128K words core store and gather data from all the sensors of the ship. This information as well as that from other ships obtained via data link is assembled, correlated and presented on appropriate consoles.

Manufacturers:

Ferranti Ltd, Digital Systems Division Berkshike, England.

Plessey Radar Ltd, Surrey, England.

e. ADAWS 5

This system is essentially a further version of the ADAWS 2 system and uses the same basic hardware and large proportion of the software of that system, but configured to meet the requirements of the "Leander" class ASW frigates, which are equipped with the Ikara antisubmarine rocket. One major difference noted is, that here only one computer is employed. The software is generated and maintained by a Ferranti program team, working at and in collaboration with the Admiralty Surface Weapons Establishment.

Seven systems have been installed on seven ships of the Batch I Leander conversion: the HMS Leander, Ajax, Naiad, Galatea, Euryalus, Aurora and Arethusa.

Manufacturers: As per ADAWS 4

f. ADAWS 6

This system is intended for the UK Navy new "through-deck cruiser" HMS Invincible. From the statements made, it can be said that two Ferranti FM 1600 computers will be used and that the Plessey contribution will reach the 1M sterling pounds level.

Manufacturers: As per ADAWS 4

g. CAAIS (Computer Assisted Action Information System)

CAAIS differs from ADAWS in that it provides action information only, and it is designed to assist operators to perform their tasks with greater efficiency rather than perform these tasks automatically.

The functions performed by the system are:

* automatic tracking of surface and air targets.
* automatic compilation of the tactical situation from radar, sonar, and electronic warfare sensor data.
* relaying of weapon direction and target identification to individual weapon control systems.
* automatic communication with ships in company, via data link.

The system employs one Ferranti FM 1600B computer and the Decca CA 1600 16" displays.

An estimated total of 50 CAAIS installation have been ordered in various configurations for specific customers. Noteworthy configureations are:

* The DBA series ordered by the UK Navy and specifically;
  - DBA 1 for general fitting or retrofitting in frigates.
  - DBA 2 for the type 21 "Amazon" class frigates.
  - DBA 5 for the Batch III "Leander" class frigates and the type 22 "Broadsword" class frigates with additional core store.

* The CAAIS 400 in the MK 10 "Niteroi" class frigates for the Brazilian Navy built in UK.

Manufacturers:

Ferranti Ltd, Digital Systems Division, Berkshire, England.

Decca Radar Ltd. London, England.

V. C^3 MANAGEMENT IN NATO

A. GENERAL

Even though many of the NATO countries, as it has been acknowledged in Chapters III and IV, have successfully developed and implemented various automated command and control systems, the Alliance itself lacks, with one exception, a unified C^3 system.

The exception is NADGE (NATO Air Defense Ground Environment), the largest single defense project authorized and completed in NATO. This complex system, which can be considered as part of a future unified C^3 system, involving a large number of locating sites and consisting primarily of radars, computers, and electronic data transmission facilities, provides early warning and response to hostile aircraft and missiles. It supplements and modernizes elements in nine European countries, following a continuous North-South sweep, that runs through Norway, Denmark, Germany, The Netherlands, Belgium, France, Italy, Greece, and Turkey. [23]

This lack of a modern unified C^3 system has been felt recently by the NATO Military Commanders, who have agreed that their most pressing need is improved command and control. [3] However, the problems for fulfilling the need in NATO and its components have their inherent difficulties.

NATO is an alliance of 15 distinct, sovereign nations, joined in a cooperative enterprise, that makes decisions by

committee. NATO defense programs are organized for the most part on a strictly national basis. The existence of national systems of finance places limits on the degree, to which integration of a common programme can be effected. The difficulty to achieve C^3 coordination in NATO can be appreciated considering the difficulties of a joint capability among the four Services of the United States alone.

Nevertheless, there are a number possibilities for cooperation effort towards the improvement of NATO C^3, which have been acknowledged in NATO [23] and in USA [3] , [24] , e.g., through:

* Standardization (see glossary for definition)

The use of standardized hardware and procedures, as it has been examined in this study, solves the problems of system interface and integration, and minimizes software incompatibility.

* Interoperability (see glossary for definition)

Enforcement of interoperability allows the C^3 to operate as a whole, with data and information exchanged throughout the C^3 structure, even if different hardware and software is used.

* Cooperation in the development and production of C^3 equipment.

This is a particular form of standardization, which can exploit the benefits of scale and reduce unit costs. Cooperation between the European and North American parts of the Alliance has to be facilitated through a "two way street" concept, as it was provided in the Ministerial Guidance 1975 for the Defense Policy of the Alliance. [23]

The main method to achieve the above capabilities within NATO is through the development of relevant standards by a wide variety of NATO technical committees, panels, working groups and agencies. These bodies are generally slow moving, but they do provide a vehicle for the nations of the Alliance to be represented on important matters of common interest.

Noteworthy is the fact that the compatibility of command and control systems is one of the subjects, where NIAG (NATO Industry Advisory Group) has offered special industrial management advice on procedures to be followed in international cooperation in industry with the NATO Governments. [23]

B. CURRENT PROCEEDINGS

Of the many initiatives underway to improve NATO C^3, the most important are found in the recommendations of Task Force 6, a part of the NATO Long Term Defense Plan. These proposals cover areas as listed below: [3]

* Strategic communications systems to connect many NATO nations, organizations and military units.
* Data-processing systems for both NATO organizations and national tactical units earmarked for NATO use.
* Tactical communications systems and command-and-control procedures.
* Comprehensive management of air defense, air operations and air-space control functions.
* Warning, intelligence support for NATO political authorities, regional military commands and national tactical units.

Strategic communications is handled by NICSMA (NATO Integrated Communications Management Agency). This organization is responsible for architecture, system design, acquisition and configuration management. Participating nations contribute to a common fund for the acquisition and operation of the NATO communications systems and the agency follows a return-sharing formula in expending these funds. The responsibility for operating the system is vested in a central authority under the Supreme Allied Commander Europe (SACEUR).

As far as data processing systems are concerned, a different approach has been followed. Due to non-existence of a central agency to provide overall ADP management, each of the major NATO commanders tries to develop his own system, SACEUR for example has organized a project office to design and implement command and control ADP information system for his headquarters in Belgium. [3]

VI. ANALYSIS OF SYSTEMS DEVELOPMENT

A. GENERAL

A major computer application, as is the development of an automated tactical real time system, constitutes a complex, time consuming and costly process. Due to the complexity involved, this process, by which the automation proposal is transformed from a conceptual idea to a sound reality, has to be approached via systems analysis techniques.

Systems analysis is a methodology by which facts about a system and the environment, in which it operates, are collected, organized and evaluated. The objective of systems analysis is to examine all aspects of the system-equipment, personnel, operating conditions and its internal and external demands to establish a basis for designing and implementing a better system in the best cost/effectiveness combination. [26]

For the purposes of this study, the activities involved are organized according to eight distinct phases of system development: [25, 26]

* Definition Study
* Preliminary Design
* Detail Design
* Preparation of Request for Proposals (RFP)
* Evaluation of Vendor Proposals
* Program and Human Job Development
* Testing
* Operations and Support

The organization of these stages identifies the basic milestones and divides the developmental effort into manageable segments. Specific information must be created in one stage before the next can proceed. The principle of concurrent development may be executed within a stage, but should not be employed across stages, for the simple reason that different stages operate at different levels.

Figure VI-1 illustrates the System Development Phases.

B. THE ACTIVITY PHASES OF SYSTEMS DEVELOPMENT

1. Definition Study [26]

The primary purpose of the definition study is to transform broad problem statements into precise statements of objectives and to resolve questions concerning the feasibility of developing a system to meet these objectives, before committing a prohibitive amount of resources.

Definition study should include the following activities:

* Definition of the Problem

The existing problem is stated as clearly as possible. This will enable a common interpretation to be maintained from the outset and through the developmental and implementation stages.

* Establishment of the study scope

The scope of the study effort should include a clear statement of the objectives of the study, any restrictions placed upon it, the composition of the study team, the study milestones

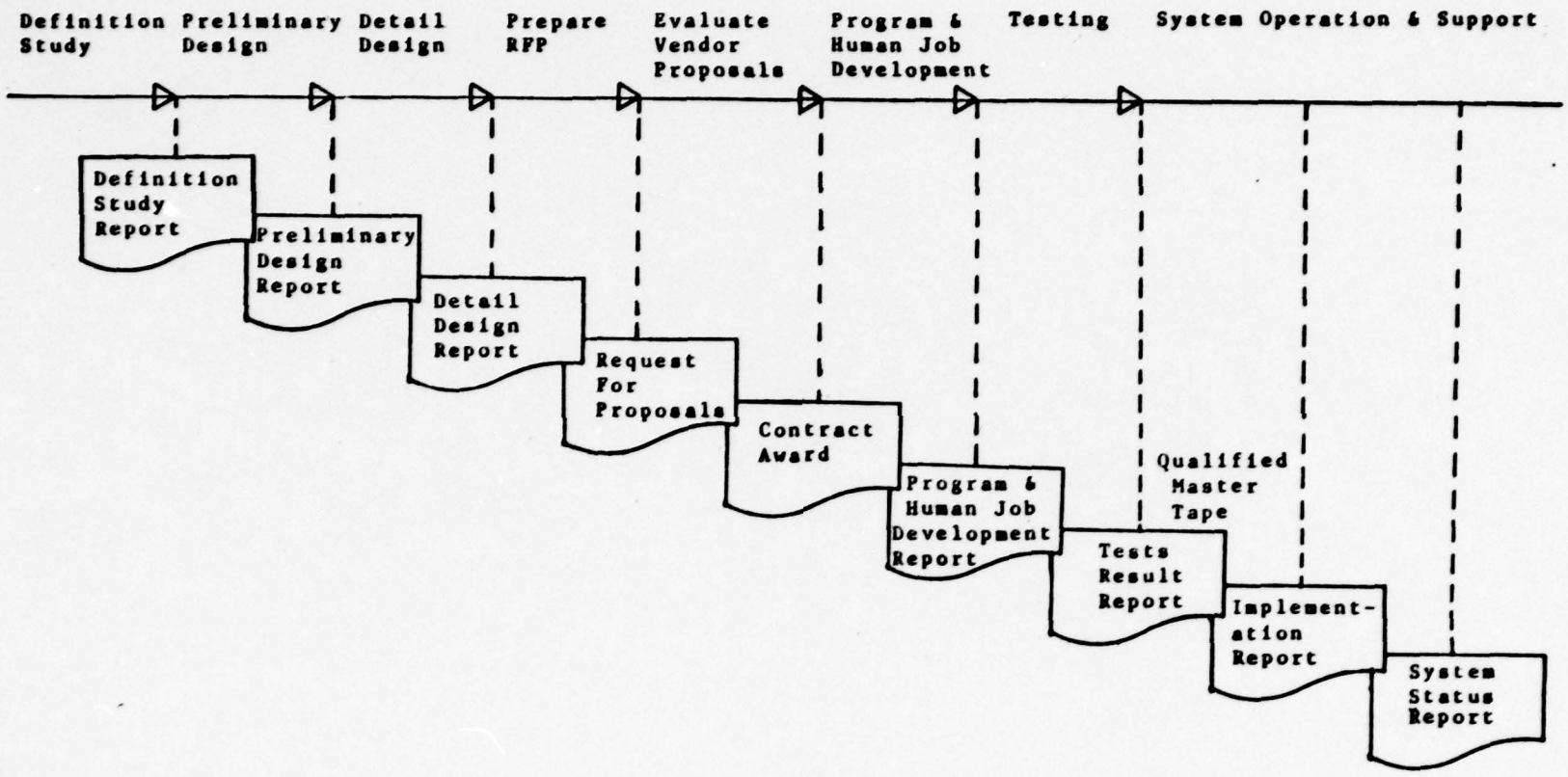


Figure VI-1: System Development Phases.

and the corresponding times by which they are to be accomplished, the content of the final report and the resources to be used.

* Analysis of existing methods and procedures

Information about the existing methods and procedures are gathered and analyzed, existing problem areas in the system are evaluated and suggestions for improvement to the system are made.

* Development of system objectives and establishment of performance criteria

The needs of the Service are translated into clear objectives or requirements to be satisfied by an operational system. They are generally a statement of prime results the system is to accomplish. They should be identified in sufficient detail, in order to permit a measurable level of performance to be prescribed for the system design. A preliminary set of objectives can be developed by using estimates of the output characteristics, which can be derived from descriptions of user activities, such as information requirements of the Service, means of expressing the information, frequency accuracy, quantity, etc.

The performance criteria are identified and stated in a manner demonstrating how the system will contribute to, and enhance the overall organizational objectives.

* Determination of resources, constraints, assumptions and items for resolution

All resources, constraints, assumptions and items for resolution are identified and evaluated to determine their effect on the new system.

Resources are the capabilities available in building the system and include hardware, software, personnel, facilities, modes of expression (charts, tables, codes, etc.) and finances (estimates of costs for implementation and operation).

Constraints are the limits of the resources in terms of resource capacities.

The assumptions are based on considerations such as historical studies, Service background, industry standards, general statistics, empirical evidence, etc.

Items for resolution are those questions or problems that must be answered or agreed before the system development.

* Analysis of system outputs, inputs and major functions

All system outputs and inputs are identified and analyzed. This analysis will indicate the major system functions and data organization. During this activity, only a rough outline of the functions that must occur is given. More detailed specifications are developed during the preliminary design, transforming functions (what must be done) into processes (how it is done). The functions identified at this time, usually fall in the categories of input preparation, input validation, data base maintenance, information retrieval, output handling etc.

* Determination of System capability requirements and potential approaches

The major system functions and data groups are analyzed to determine general system capability requirements (e.g., data base storage and organizational requirements).

Based on these requirements, a number of system approaches is developed. The purpose in considering a number of alternative approaches is to determine a configuration approach, that is most favorable on a cost/effectiveness basis.

* Evaluation and selection of system approach

The approach of the previous activities are evaluated objectively and selection of the approach, which provides the same, or better performance at lower cost than any of the others.

* Determination of the types and magnitude of expected implementation and conversion requirements

In many systems development efforts the problems of implementation and conversion are of paramount importance, requiring in some cases expenditure as the rest of the system effort in total.

The requirements of implementation and conversion must be fully understood and explicit at the definition stage and appropriate techniques developed for the resolution of problems.

At this stage however, only a rough assessement of the manpower resources and the time required for this effort are made. More detailed examination will be undertaken at the Design stages.

* Preparation of the overall plan and cost/benefit analysis of the proposed system

The overall plan is derived from the system objectives and from the overall structure of the selected system configuration.

This plan will identify those processes necessary to build a system, which will meet the system objectives for the system approach recommended.

The construction of a cost picture of the proposed system is based on the best estimates available. At this stage the cost figures will be rough, but reasonably accurate. The estimation of benefits is often more difficult, especially since many benefits may not be quantifiable (e.g., the destruction of an approaching missile). In such cases the quantified costs must be applied to benefits that cannot be expressed in terms of relatable quantifiable savings.

* Preparation of the Definition Study Report

The results of all the previous activities are summarized in this report, which is considered as a final technical report.

This report will be the "base line" for further system development.

2. Preliminary Design [26]

This stage provides the transition from a definition of system approach and objectives to a fully designed system specification. This system specification consists of a preliminary design of the entire system, including hardware, software and personnel subsystems to the point where programs and procedures may be designed and hardware may be ordered.

Considerable effort and money will be spent during the design period. Experience has shown that the final product may

differ significantly from the initially defined product. It is therefore desirable to arrive at a preliminary design milestone, before the design is completed in every detail. In this manner the design can be changed, if necessary, before further detail is developed.

The result of this stage is the Preliminary Design Report. The report will provide the Service with concise information necessary to decide on whether to continue the system development effort.

The activities included in the Preliminary Design Phase are the following:

* Specification of system expansion requirements

The initial system design should be examined from the standpoint of what may be required after the system has been operational for some time.

Two approahces are considered in order to ensure system expandability: 1) design the system with considerable overcapacity built-in, 2) design the system to accommodate the immediate future load, but with easy upgrading techniques included in the design.
Normally a combination of the two approaches is preferable.

* Definition of the overall system capacity

In this activity the hardware and software environments required to enable the system to meet the objectives are specified. This will lead to decisions for providing capabilities, that are lacking or, where this is not feasible to changes in the systems objectives.

The following requirements should be considered, when defining the system environment:

(1) Applications requirements

- multiprogramming/multiprocessing
- batch processing (remote)
- demand processing (time sharing)
- real time processing
- processing time-frame and load

(2) Data communications requirements

- number of data sources and points of distribution and their locations
- volume and response time of collection and distribution of data
- number and type of terminals
- degree of accuracy and penalty for system failure

(3) Data Base requirements

- storage
- access methods
- organization
- recovery and fall-back
- security

(4) Physical facilities

- facilities needed and their locations
- existing facilities

* Description of Subsystems

A description of the system segmentation is prepared, together with a functional description of the processing requirements for each subsystem.

* Development of subsystem input, output and interface requirements

Input/output requirements are determined by analyzing the system processes occuring in each subsystem. The subsystem designers interact to combine their respective input and output requirements into subsystem interface records. The interfaces represent the degree of subsystem interaction and data flow.

The data catalog is established at this time and will be maintained and updated throughout the life of the system. This catalog lists all the data items used within the system, using standard name and number. Associated with these items will be the processes and files, that use the data items as the processes and files are developed.

* Preparation of system/subsystem flowcharts

A process is the logic of how a particular set of functions within a subsystem is performed. This logic is most conveniently described in terms of a flowchart. This description will vary between a batch and a real time system. For a batch system, the actions will be grouped into a process according to a similar logic and intermediate results for inputs or transactions passing through the subsystem. For a real time system, an individual transaction is described by a string of specific actions, which that particular action requires.

* Develop process descriptions

The processes identified on the system/subsystem flowchart are expanded into descriptive narrative detail Narratives support the flowcharts providing details and explanations.

* Specification of system protection requirements

This activity specifies appropriate corrective action in cases of failure error or compromise within the system.

* Identification of human engineering problem areas

All areas, where human engineering is required to optimize personnel effectiveness, comfort and safety, are identified.

* Design of logical data base structure and definition of the access techniques

The records currently being used by the existing system are evaluated against the information requirements of the new system. This results in the selection of records to be used for generating the data base of the new system and in the establishment of a data base structure and a system for retrieving the data in the manner required.

At this stage the designer is primarily concerned with the logical structure of the data base, that is, how it will be seen by the programmer and user. Similarly the logical accessing methods (e.g., chain, list etc.) are described now.

* Specification of data communications requirements

All the communications equipment and facilities required by the operational system should be identified. Comprehensive specification for the relevant communications network are developed.

In the specification of the communications requirements factors, such as the availability of existing communication facilities (lines, microwave, etc.), loads (peak, average), future expansion in traffic load, should be considered in order to calculate trade-offs between capacity and cost and in order to make necessary design adjustments.

Various forms of networks should be considered. Packet switching, message switching, or centralized processing (star-shaped) configurations may all be possibilities.

* Specification of hardware configuration

During this activity, the hardware configuration is specified. Processing requirements should be calculated and CPU and I/O device requirements should be determined. These requirements will be used to determine the CPU size and necessary speed. The general characteristics of the I/O and data base devices must be also defined.

The hardware should include: 1) central processor, 2) support processors, 3) core modules, 4) I/O devices, 5) channels, 6) mass storage devices, 7) control units, 8) peripheral devices, 9) special purpose devices (e.g., modems).

* Specification of system software

During this activity, the software characteristics, which will be required to support the designed system, are specified. Some important software components are:

- Operating System (partitions, multiprogramming/ multiprocessing, roll-out/roll-in, control language,

diagnostics, hardware control, compatibility, interrupts, librarian, system generation, priority assignment and control).

- Languages (high level languages to be used, report generator, assembler, special purpose).

- Management software (data management, communications handlers, transaction processing, real time, time sharing, miscellaneous packages).

- Utilities (sorts, merges, data transcription, file maintenance, debugging aids, error handlers, checkpoint, accounting routines, dumps, labeling routines, copy routines, extracts, conversion routines, program maintenance routines).

- Special software (statistical packages, other mathematical packages, etc.).

* Preparation of development and implementation plan

A more detailed plan, than the one developed in the Definition Study, is prepared during this activity. It should describe, in detail, the activities for the remainder of the effort, including all milestones and manpower requirements. Several methods, such as PERT charting may be used.

* Preparation of the Preliminary Design Report

This activity produces a report with the most accurate and complete picture of the system, using the up-to-date information available.

After revision and approval, the Preliminary Design Report becomes the base document for future development.

3. Detail Design [26]

During this phase the preliminary system design is expanded and refined to produce detailed specifications for all program modules, manual procedures, and information files. The resultant specifications are precise, explicit and in sufficient detail to support the next phase. This means that all computer logic, data base organization, hardware/software specifications are completely specified and documented.

The activities included in this phase are as follows:

* Development of human procedures

A series of statements, that establish a course of action to be followed consistently under stated conditions. They instruct personnel in their responsibilities and they improve performance and organization.

* Design of manual forms and computer input/output interfaces

Here the design of all manual operating forms is completed, including all subsystem written communications, except those which exist as computer interfaces.

Furthermore, the design of all computer interfaces to human procedures is completed, considering the human engineering factors of emphasis, legibility, alignment, spacing and brevity.

* Design of physical data base

This activity consists of synthesizing the requirements of the logical data base, specified hardware, data

management, hardware requirements and the user-processing requirements into an optimum implementable whole.

During this activity the required data base items are grouped into physical data record formats.

* Design of subsystem protection features

During this activity, those requirements for level of protection (already specified in the Preliminary Design), which are supported by subsystem design, as it now stands, are translated into a form design. Features requiring design may include failure isolation, fall-back, recovery, data base reconstruction, data privacy, data integrity, facility security, hardware reliability.

* Definition of subsystem programs

During this activity, the processes to be performed in the subsystem are combined and/or subdivided into program descriptions. This grouping is normally done on the basis of logic similarity, data requirements, function sequence or some combination thereof.

* Development of logic flowcharts and tables

In this activity, the logic of each program module is laid out in graphic form.

* Specification of software utilities and common routines

During this activity programs and program modules are analyzed to determine the possibility of replacing them with utility software, in order to reduce programming effort.

* Specification of data communication requirements

In this activity all communications equipment and facilities required by the operational system, should be identified. These needs are then refined, combined and adjusted to develop comprehensive specifications for the communications network required.

* Development of subsystem test plan

In this activity a test plan is prepared to satisfy those specifications established during the design activities, expanding and detailing previously defined test requirements.

The plan should include schedules, desired results, evaluation of results, configuration required (core, disc, tape drives, etc.), volumes to be used, duration, variable requirements, time and manner simulation is to be used, personnel required, processes to be tested (manual and machine).

* Preparation of Detail Design Report

After all detail design activities have been completed and the system design is agreed upon the Detail Design Report can be prepared. Beyond describing the ststem as it currently stands, the report should highlight those changes introduced during the Detail Design, that will have impact on the system's planned performance.

4. Preparation of Request for Proposals

During this phase the Detail Design Specifications are communicated to the vendors, who are invited to participate in a competitive procedure of offering their products and services.

5. Evaluation of Vendor Proposals

In this phase the proposals of the vendors are evaluated on criteria established in the RFP document. The most cost/effective proposal is selected. This stage culminates with a Contract Award.

Following is an indicative list of criteria, on which vendor proposals may be evaluated:

* Personnel support

The numbers and qualification of analysts and programmers are examined.

* Performance on Benchmark Programs

The performance on benchmark programs for single job and multiprogramming is evaluated with respect to time, equipment needed, O/S needed and reliability shown.

* Programming

Factors such as ease of programming, language features, compilation speed, compiler diagnostics, execution diagnostics and program size, as they were determined from benchmarks, are considered.

* Operating System

The resident O/S system is evaluated with respect to components (Scheduler, Memory Management, etc.), Resources used (Storage Devices, CPU Time), Maintenance required, Partitioning of programs between internal and external storage.

* Checkout facilities provided

The place, time, duration, hardware configuration and available software of the checkout facilities provided, is considered.

* Program Conversion

The capabilities with regard to Conversion from Language A to Language B, cleaning up of incompatible commands, emulation and rewriting of programs are examined.

* Test Tools

The existing test tools (debugging packages, test data generator etc.), are considered.

* File Conversion

The capabilities for file conversion with respect to program conversion, support provided and Media/Format conversion are checked.

* Documentation

The existing documentation with respect to Programs, Operating Instructions and Maintenance, is evaluated.

* Physical facilities required

Physical facilities required, such as Space, Electrical, Air Conditioning, are examined.

* Training provided

The quantity and quality of the training provided, with respect to programmers, computer operators, data operators, is carefully considered.

* Backup facilities

The number of backup facilities in the area and at plant are compared.

* Acceptance tests

The time, location, duration of acceptance tests, as well as the specific test programs are examined.

* Vendor record

The performance, hardware reliability and the support of the vendor are examined.

* Installation Schedule

This schedule with respect to hardware installation, program and file conversion and turnover are considered.

* Expansion Capabilities

The expansion capabilities provided with regard to hardware and operating system are compared.

* Costs

Relevant costs regarding hardware, software, people, maintenance and other services are examined and compared.

6. Program and Human Job Development [26]

The purpose of this phase is to covert the completed design into executable manual jobs and computer programs, which can be tested subsequently.

In practice, this phase is devoted to the process of verifying the program specifications produced in the Detail Design, producing coding diagrams, coding the programs, compiling and desk checking the programs and finally debugging the programs to the point at which they may be turned over to those responsible for System Testing. The process is normally the responsibility of the programmer, who is the author of a particular program.

Any methods of saving programming time and cost, such as automatic coding, program generation, desk debugging,

flowcharting and documentation aids should be considered and applied, whenever possible. Special care should be paid in the case of multiprogramming and multiprocessing, due to the increased complexity of the programming effort. This is also true of programming for communications systems.

Program Development includes the following activities:

* Synthesis of position descriptions for personnel subsystem

In this activity the procedures developed in the Detail Design are grouped into position descriptions for personnel. A position or job is defined as the workload assigned to one man in the proposed system. There may be one or more procedures involved in one personnel position description or one procedure may be divided into several positions, depending on the size and complexity of the procedures.

* Establishment of personnel and environmental requirements

During this activity, initial estimates are made of the types and numbers of personnel required by the system. A training plan should also be prepared to include: 1) system orientation at all levels, 2) specific position training, 3) cross position training.

In addition, detailed human engineering specifications are prepared for those system operations, which require the presence of people.

* Development of detailed program flowcharts

This activity, optional in some cases, provides for the detailed flowcharting, that will approximate the instruction level and be complete enough to guide the coder in the performance of his task.

* Coding of Programs

This activity incudes the coding of theprograms, using the specified language for this purpose. The most important control for ensuring good coding is to have a comprehensive and useful set of coding standards, plus a method for ensuring that these standards are followed.

It is very important that consistent and effective use is made of comments. Each routine should have at least a brief explanation.

* Preparation of source decks and compile/assemble programs

This activity includes the completion of coding to a final, fully prepared source deck, which together with the necessary control cards (compiler control cards, source image library control cards, special purpose control cards), is submitted for compilation/assembly.

After each compilation/assembly, the programmer should carefully check the output produced for errors.

When correcting discovered errors, particular care should be taken to ensure that new errors are not created.

* Preparation of program (module) debug data

The purpose of these data is to: 1) execute every routine, 2) execute every instruction, 3) remove the inherent

bugs created during coding and source preparation, 4) test the limites of arithmetical and edit instructions, 5) test the correctness of the internal program (module) logic.

The responsibility for this activity belongs to the programmer. The methods of preparing debug data include singly or in combination branch point analysis, general analysis, test data generators and samples from existing files or inputs.

* Debugging of Programs

This activity begins after a completely coded program has been compiled without error and job control is complete. The program may consist of one or more load modules. All possible functions and conditions, contained in the program must be checked out, utilizing the branch point analysis chart prepared in the previous activity. Here it is where the rigorous debugging of an individual program occurs.

This is the stage of testing, where every instruction within the program is exercised and where the verification of internal consistency within the program is made.

* Preparation of Program and Human Development Report

During this activity, the documentation must be revised to ensure that the description of the system continues to be accurate and complete.

All changes made to the system, during this stage should be identified and their impact on the overall system reassessed to ensure that they are consistant with the overall system objectives and that they have caused no performance degradation, when related to any portion of the system.

The final report should subsequently be prepared and should include a section for personnel related information and a section for program (module) related information.

7. Testing [26]

Testing is the phase, where the validation of all system components occurs. It constitutes the basis for accepting a system and its completion represents one of the more significant milestones. Testing at any level may result in the redesign of a portion of the system.

The extent of testing is one of the most critical points during the system life cycle and one that frequently causes the most troubles and lost time.

The typical activities included in testing are as follows:

* Development of detailed test plan and procedures

Even though extensive test planning was accomplished in Detail Design, a more detailed planning will be required at this point by those responsible for conducting the tests.

A detailed test plan is prepared for testing, as a whole and for each identifiable test that will take place. Furthermore standard procedures, such as "always dump core", or "always write out register contents", when a program halts, must be prepared.

* Preparation of site and installation of hardware and facilities

This is the latest point in time for this activity to take place. Any delay in the preparation of the site or installation of hardware can disrupt the entire schedule.

* Development of job stream and job control

In this activity the designers, responsible for job stream preparation and those concerned with system performance, will develop the best operational arrangement for program modules.

Job streams are the linking of program modules into executable sequence. Implicit in the development of job stream is the development of the job control, since this is usually the only way of defining the load module structure and attendant factors.

* Test training courses, work aids and human procedures.

Previously developed training programs are tried out on groups of people in practice training environment.

Work aids are exercised to determine whether they support the task, as anticipated.

All human procedures that were completed in the Detail Design are validated in simulated work environment.

* Build test data base and transaction files.

This is a critical activity, as the entire testing of the automated system and future reliability of that system will probably related to the adequacy of the test data used.

The employment of an automated test data generator, either as an "in house" product or as an outside service should be considered, especially in large system efforts.

The importance of the test data base cannot be overstressed. Enough data must be used to create the anticipated operational conditions and enough volume to cause an overloading of the system. An inadequate test data base could lead to the acceptance of a system, that could not survive under loaded operational environment.

Test transaction files should be created either through a test data generation or by collecting a representative sample of live transaction. Coordination between the test data base and the test transactions is required.

* Test subsystem/system.

Subsystem testing includes a careful testing of a program in its environment, that is, the adjacent programs.

For batch systems this includes job stream testing.

For on-line or real time systems the process is to test the transaction paths, which involves exercising all programs or processes required for a unique input transaction type.

In this activity the project team creates and executes a test procedure to ensure that the completed subsystem can operate without disturbance, its performance is up to standard and that it meets the objectives originally set forth. The work load will be as in a normal work day and will then be expanded to overaload conditions.

The objectives of the subsystem testing is to detect deficiencies not found in debugging and to assure, that the

subsystem will function as required under conditions closely related to those encountered in the operational environment.

Once subsystems have been independently tested, they must be brought together into a system test. This step is concerned with the testing of all relationships between subsystems and of the relationship between the system and the outside environment.

* Performance of acceptance test

During this activity it is verified to the user, that the system meets the system objectives and performance criteria and is operationally sound.

In a small system it is possible to conduct this step immediately after implementation. However in large systems the process of conversion and implementation is so large, that it is appropriate to ensure the workability of the system before undertaking an expensive and possibly useless implementation and conversion. Successful completion of acceptance test is the main criterion for final turnover of the system for implementation. The acceptance test should be therefore thorough and complete.

* Preparation of the Test Results Report

The test results report is a final technical report that should reflect the testing phase.

8. System Operation and Support

This phase represents the normal operation of the system after its implementation and continues as long as the system is maintained operationally.

The activities required during this phase include:

(1) For the operation task [26]

* Establishment of implementation control plan

A detailed implementation plan is prepared to include solutions to problems identified during testing in the areas of hardware, software and man/machine interfaces. Completion of materials required for implementation (user guides) and operation (work aids) and turnover of the implemented system.

* Training of operations and maintenance personnel on hardware, software and the new system

Proper attention should be given to personnel training on hardware, software and the new system, to minimize the impact of operator's error during the operation of the new system.

* Turnover of system and documentation of implementation

This activity summarizes the implementation effort, which culminates when the turnover of the system to the operational personnel occurs.

* Developing and monitoring critical indicators

Certain critical indicators are established and monitored, to provide advance warning of conditions, which may cause system degradation or collapse.

* Scheduling of computer operations

The scheduling of computer operations is completed in accordance with the operational demands and resources available.

* Prevention and recovery from run failures

Precedures for the prevention and recovery from run failures would be established in order to insure smooth functioning of the system and minimize repair time. They should be derived from the design and programs for fallback, recovery and correction.

* Monitoring disaster control and security plans

Disaster control includes all measures required to avoid the loss of software, beginning with alternate storage of all programs.

Security control regards all measures, that should be taken to prevent unauthorized access or sabotage.

(2) For the maintenance task [27]

* The software of real time tactical systems is one of the largest and most complex applications software in existence today. Naval tactical software provides the control of electronic sensors and displays, computations of tactical and navigational algorithms, control of weapons and ordnance and real time response to man-machine interface demands. These automated tasks are performed by software for the purpose of accomplishing naval mission requirements and pursuing objectives of Navy interest.

In order to accomplish the support taks, an "in house" Software Management Organization should be established. The organization should begin with the highest echelon of command, which will establish overall objectives and provide funding

and include appropriate organizational divisions to carry out the required tasks. A Work Breakdown Structure might be as follows:

* Project Management

  Provides direction, plans and schedules.

* Configuration Management

  Provides configuration baseline control.

* Quality Assurance

  Provides verification and validation.

* Software Production

  Provides engineering analysis, design, code and debug.

* Laboratory Facilities

  Provide operation and maintenance, generation, integration and training facilities.

The support phase is the key to the ability of the Navy for maintaining the readiness of its software in the face of changing threat conditions.

The maintenance task begins at the time the software is released to the Fleet for operational use and remains in effect as long as the specific tactical real time system is in operational use. During this phase, Fleet users submit requests to the Software Management Organization for software modifications, which result in periodic updates and reissues of functionally improved software.

The support task minimizes the obsolescence of the platforms (ships, aircraft), on which the system has been installed.

Because the platforms are programmable, it is possible to recode portions of the software and to create new functional capabilities. In their use of the platforms, Fleet operators may determine many unique improvements to the operation of the platform. Many of these improvements can be implemented via software or as a software work-around for a future hardware change.

Periodically, but not frequently, the Navy can make major hardware revisions an/or additions to a platform.

Through periodic software re-issues (say 1 to 1½ years), the Fleet can obtain system improvements and thereby increase tactical proficiency, in order to handle dynamic threat conditions.

C. STRUCTURING THE SOFTWARE BASE [28]

The software base is structured, in part, by partitioning the software tool types according to the specific development activities they support. In order not to ignore that part of the software base, that supports the operation of such tools and to indicate clearly those tools which support more than one activity, the software base is structured further through a layered approach, that provides insight among the software base components.

There are at least five distinct virtual layers, associated with an operational computer system and three layers of software, that support the software development process. Layer 0, the innermost layer, represents the bare computer hardware,

including items such as processors, channels, main storage, mass storage, bulk I/O, archival storage, hardware monitors, terminals, sensors and communications interface devices. Layers 1 through 3 "reside on the hardware and collectively provide the virtual machine capability, that is necessary to support layer 4, the applications software.

The following paragraphs provide a short description of the layers 1 through 3 along with a listing of the tools, that reside on those layers and the relationship of such tools to the various development activities. A concise description of each tool is provided in the glossary section of this study.

1. Layer 3: Functional Support Tools

Layer 3 contains those tools, that provide direct support to the activities of the software development phases. These are the tools with which the applications software developer has the most interaction. Layer 3 tools are related to the specific development activities they support.

a. Tools supporting Phase 1 (Definition Study)

The types of tools that are directly applicable to requirement analysis are listed below:

* General Purpose Simulators

* System Description Languages & Analyzers

b. Tools supporting Phases 2 & 3 (Preliminary and Detail Design)

The types of tools that are directly applicable to software design are the following:

* Computer System Simulators

* Data Base Design Aids

* Data Dictionary System

c. Tools supporting the activity "Development of test plan" in Phase 3 (Detail Design)

The types of tools required to support system test construction are listed below:

* Test Data Generators

* Test Data Auditors

* Test Case Design Advisors

* Test Instruments & Analyzers

d. Tools supporting Phase 6 (Program Development) and Activity "Testing Subsystem" of Phase 7 (Testing)

The types of tools required to support the program development and the subsystem test activity are listed below:

* Assemblers

* Compilers

* Linkers

* Debugging Aids (Interactive Symbolic Debuggers, Non-Interactive Symbolic Debuggers, Interactive Absolute Debuggers, Non-Interactive Absolute Debuggers)

* Module Libraries & Change Control Systems (Basic Libraries, Integrated Libraries, Automatic Software Production & Test Systems)

* Performance Monitors (Language Dependent Monitors, Language Independent Monitors)

* Standard Enforcers

* Preprocessors and Reformatters

e. Tools supporting Activity "Testing System of Phase 7 (Testing) and Phase 8 (System Support)

There are no unique layer 3 tools that exist to support these activities. The tools, that were listed to support the aforementioned phase and specific activities, are generally applicable. Most of the tools used in practice, that are specifically oriented to system test activity, are special-purpose, e.g., test environment tools (Emulators, Hot Benches, System Integration Lab. support, virtual machines), test drivers, and special purpose monitors.

2. Layer 2: General Support Services

The primary function of layer 2 tools is to provide a framework of common services, that will allow the output of the third layer function to be stored, retrieved and inter-communicated. Second layer functions should be usable for common purposes across different third layer functions and should serve to hide (where possible) differences between first layer and third layer functions. Layer 2 tool types provide general support to all of the software development activities. These tools are listed below:

* Data Base Management System
* PERT/CPM Systems
* Project Estimation Systems
* Documentation Aids (Text Processing Systems, Flow-charts Construction Languages and Automatic Flowcharters)
* Data Manipulation Utilities
* Information Retrieval Systems (Query Language System, Report Writers)

3. Layer 1: Operating System Services

Layer 1 implements the operating system services, that present a "virtual machine" interface to the services/tools at layers 2 and 3 and manage the real time hardware. The layer 1 tool types are generally applicable across all of the software development activities. Layer 1 tool types are listed below:

* Basic Operating System (BOS)
* Multiprogramming Operating System (MOS)
* Multiprocessor Operating System (MPOS)
* Virtual Machine Monitor (VMM)
* Time Sharing Operating System (TSOS)
* Real Time Operating System (RTOS)

VII. COST CONSIDERATIONS

A. GENERAL

While not directly involved in the technical performance of a computer system, cost is perhaps the most significant factor in procurement and can overwhelmingly dominate performance specification, evaluation and contractor selection. The cost elements are at least as complicated as the technical aspects. It is important that the purchaser have a good knowledge of the entire spectrum of costs, related to a computer system, or the cost can rapidly escalate beyond his resources.[29]

Basically, the costs of a computer system fall into two broad categories:

* Hardware costs
* Software costs

In general, it can be said that software costs rapidly outgrow hardware costs. Two reasons are behind this trend:

* The reduction of hardware cost due to the advances in the Computer Electronics Technology:

Three major factors are considered as pivotal to computer/processor development: 1) density, 2) speed, 3) cost. A brief consideration of these factors is necessary to explain the reduction of hardware cost with the advance of technology

Dr. Gordon Moore (founder of Intel Corp., with Robert Noyce) noted in 1964, that the density, or number of components per chip (circuit) had doubled every year since 1959. Moore

further predicted, that this trend would continue. Moore's curve, as illustrated in Figure VII-1, appears to be continuing its trend, supported by technology. This curve is the basis for predicting the future of computer/processor, since most all characteristics of the computer/processor can be related to density with a high degree of confidence. [29]

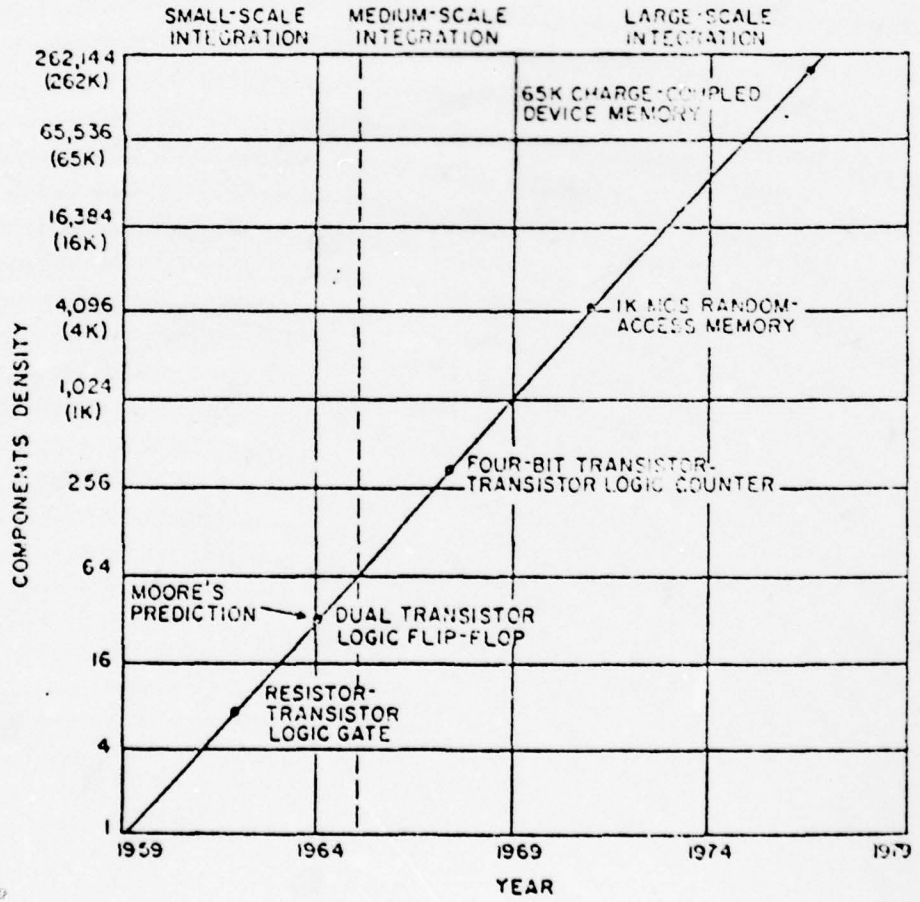


Figure VII-1: Moore's curve

The instruction timing of a computer/prcocessor (or inversely the number of operations or throughput, that an architecture can expect) is related to density by the relation:[29]

$$\frac{1}{t_i} = 0.115\ N_c^{0.448} \qquad \text{(VII-1)}$$

$$r^2 = 0.948$$

where:

t_i = time to execute an average instruction (us)

N_c = number of circuits

r^2 = correlation coefficient

Evidently, as the density increases, the processing capability for a given volume or weight improves, as indicated in Moore's curve.

The cost of a computer/processor is related to the density by: [29]

$$C_{hw} = 594.74\ N_c^{0.795} \qquad \text{(VII-2)}$$

$$r^2 = 0.965$$

where:

C_{hw} = cost of hardware in dollars

N_c = number of circuits

r^2 = correlation coefficient

Therefore it can be said that the computer/processor can be related to throughput with the above relations VII-1 and VII-2.

If the cost to produce N_c (number of circuits or chips) goes down, the cost of performance (or throughput = $\frac{1}{t_i}$ in MIPS) will decrease also (see Figure VII-2).

The curve of Figure VII-3 protrays the trend of silicon-device prices that dominate the costs of circuits, dominating computer/processors. This curve is used as the basis of trend predictions for the cost projections of Figure VII-2.

The general trend, that is obvious, is that an order of magnitude improvement is seen in the cost versus speed relationship every ten years. [29]

* The continuing escalation of software costs.

Both the major phases of software life cycle, that is, software development and software maintenance, are contributing to this cost escalation.

Software is generally on the critical path of system development. Any slippages in the software development schedule translate directly to slippages in the overall delivery of the system. Software development tools are also expensive. A new architecture may involve software tools in order ot $24 to $28 million. A decision to use a new architecture, with a resulting higher program costs, can skyrocket the software costs, even though the cost per instruction may seem an insignificant percentage.

Software maintenance costs are even more high and will continue to escalate as long as there is a continuing addition to the inventory of code via development at a faster rate than code is made obsolete. For example, on one aircraft computer, software development costs were roughly $75 per instruction, while maintenance costs ran as high as $4000 per instruction. [32]

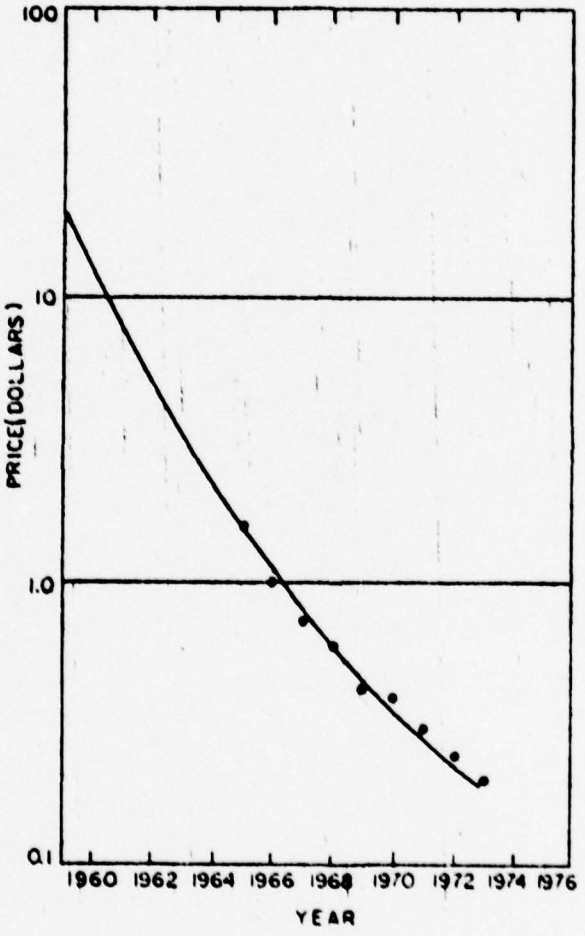


Figure VII-2: Trend of silicon device prices.

Figure VII-3 portrays the software life cycle cost breakdown on four large scale software activities, while figure VII-4 illustrates the hardware/software trends in the time fram 1953 to 1985. [32]

Estimates of the overall annual cost of software in the United States alone range from $15 to $25 billion.

U.S. DoD spends an estimated $3 billion per year. Some 4 to 5 per cent of the U.S. Air Force budget is in computer software.

It is important that the magnitude of the above figures are realized, when a trade-off decision is to be made between hardware and software costs, where either is impacted by the other. A decision to use a new architecture with a resulting higher program cost can skyrocket the software costs, even though the cost per instruction may seem an insignificant percentage. [29]

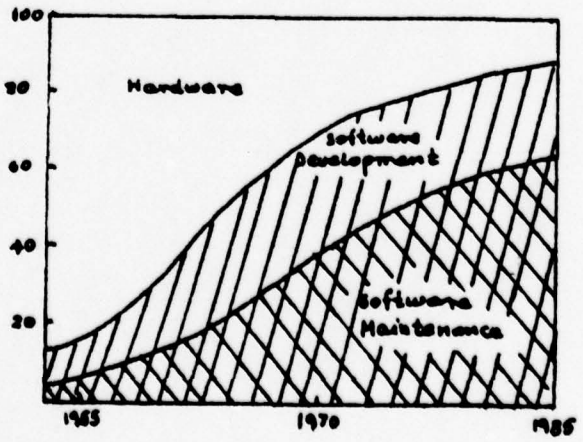


Figure VII-3: Hardware-software cost trends.

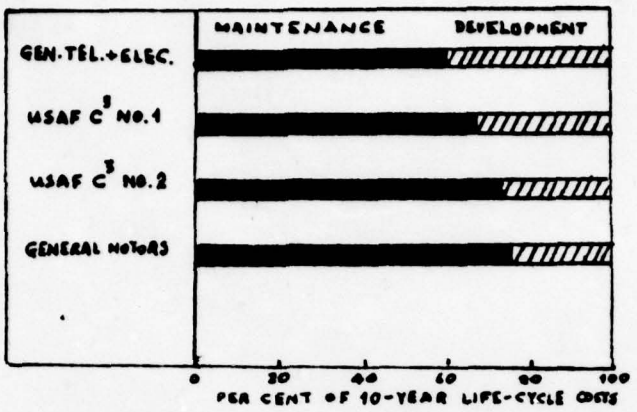


Figure VII-4: Software life-cycle cost breakdown.

The computer cost is significantly related to total investment in an architectural base by: [29]

$$C_{si} = 42.34\ B_2^{-0.149} \qquad \text{(VII-3)}$$

$$r^2 = 0.902$$

where:

C_{si} = cost per instruction

B_2 = total dollar value of architecture inventory ($M)

r = correlation confidence (1.0 = perfect)

Figure VII-5 illustrates a comparison of the cost of a family of computers based on the above relation.

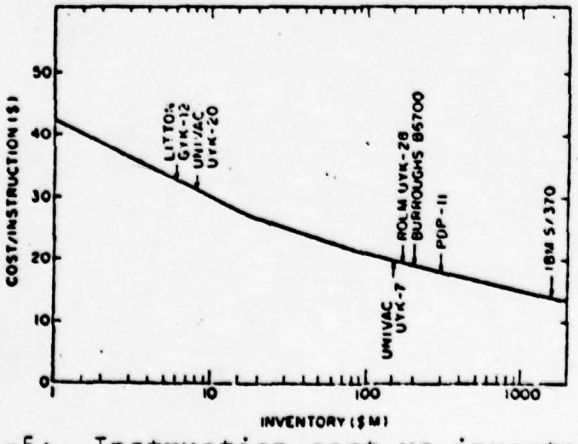


Figure VII-5: Instruction cost vs inventory investment.

B. TOTAL LIFE CYCLE COSTS

The computer resource life cycle cost for system i and architecture j is defined as: [29,30]

$$C_{ij} = HW_{ij} + ASw_{ij} \qquad \text{(VII-4)}$$

where:

HW_{ij} = hardware life cycle cost

ASw_{ij} = applications software life cycle cost

Specifically for a microcomputer system, the applications software life cycle cost ASw_{ij} may far outweight the hardware life cycle cost HW_{ij} (e.g., a microwave-oven manufacturer can spend several hundred thousand dollars to develop a software program to run a computer costing twenty dollars or less). [20] For this reason the computer resource life cycle cost for this system has to be amortized over hundred or thousands of units. Therefore the relation VII-4 becomes for the case of micro-computers as follows: [31]

$$C_{ij} = HW_{ij} + \frac{ASw_{ij}}{N_{ij}} \qquad \text{(VII-5)}$$

where:

N_{ij} = number of systems i of architecture j produced.

1. Hardware Life Cycle Cost

a. The Computer Family Architecture (CFA) Life Cycle Cost

The computer hardware life-cycle cost for a given system, using a specific CFA (ranging from mini to large systems) is defined as: [29,31]

$$HW_{ij} = n_i L_h (P_{ij} + MM_{ij} + SM_{ij}) \qquad \text{(VII-6)}$$

where:

i = index of the system

j = index of architecture

n_j = number of systems to be produced for system i

L_h = hardware life-cycle cost factor, i.e., ratio of total hardware life-cycle cost to hardware acquisition cost. This factor is assumed to be 2 for a 10 year life cycle.

P_{ij} = processor acquisition cost

MM_{ij} = main memory acquisition cost (see Fig. VII-6)

SM_{ij} = secondary memory acquisition cost (see Fig. VII-6)

(1) Processor Acquisition Cost. Principally based on technology, the processor speed is a factor of memory speed. At present there are three technologies that dictate three ranges of memory access time:

* core
* metal-oxide semiconductor (MOS)
* bipolar

Corresponding costs are ranging from 0.1 cents/bit to 1.0 cents/bit.

These technologies represent memory access time (and processor speeds) of from 300,000 IPS to 80 MIPS, as shown in Figure VII-7.

The processor acquisition cost for system i, using architecture j is defined as: [29,30]

$$P_{ij} = k\,(a_{ij}\,Mr_i)^{0.4} \qquad \text{(VII-7)}$$

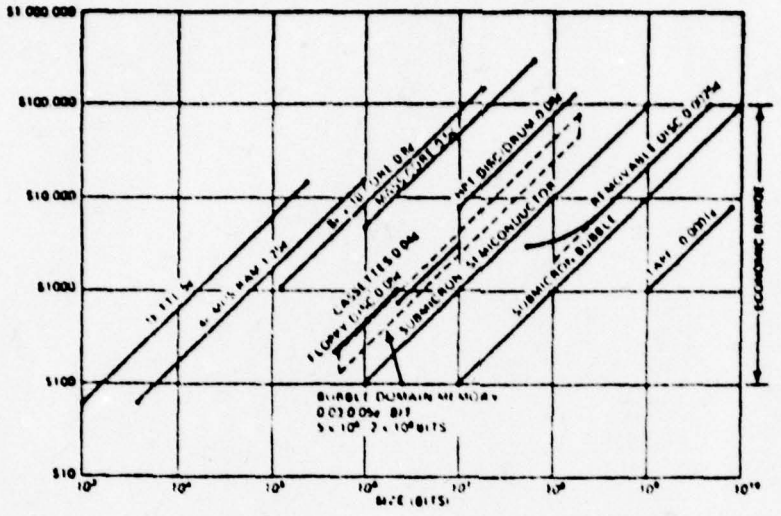


Figure VII-6: Cost vs memory size.

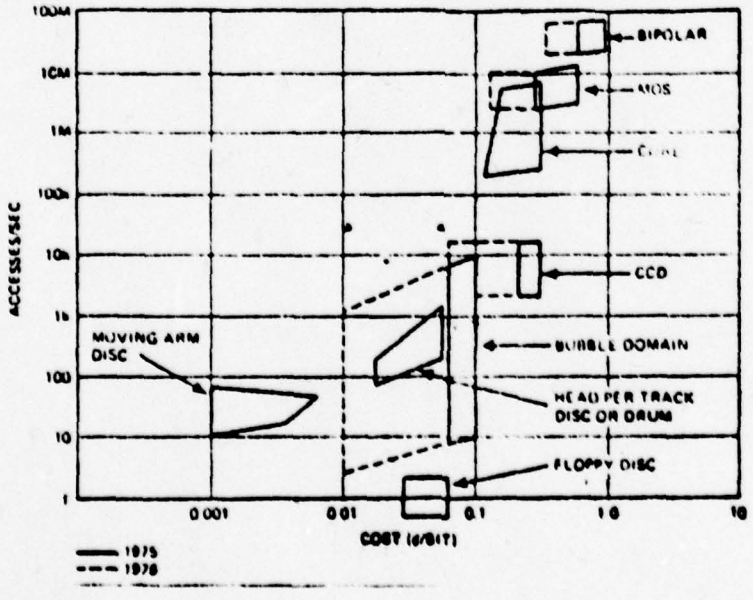


Figure VII-7: Access time vs cost (1975-1976).

where:

k = constant relating to processor cost

a_{ij} = processor speed ratio

Mr_i = operating speed in millions of instructions per second (MIPS)

Equation VII-7 follows from a commonly cited relationship between performance and cost, namely:

Performance = Constant * Cost ^{g}

To obtain a value for constant k in equation VII-7, CFA used the fact, that recent cost/speed data for several military processors seem to indicate that speeds of 0.5 MIPS and processor costs of $48,000 are representative values; therefore:

$$k = 48 * 10^3 / (0.5)^{0.4} = 6.3 * 10^4$$

This value of k refers to 1976 processor cost estimates and it is reduced by a factor of 10 for 1985 cost estimates, based on an assessment of hardware cost reduction over the decade 1976-1985.

The initial acquisition cost of a processor with $\bar{r}_r$ average instruction processing time is: [29]

$$C_{Hi} = \frac{k}{\bar{r}_p^{\,g}} \qquad \text{(VII-8)}$$

where:

C_{Hi} = cost of the ith processor

k = $6.3 * 10^4$

g = 0.4

$\bar{r}_p$ = average instruction execution time (in μsec)

(2) Main Memory Acquisition Cost. Presently, there exist three types of main memory technologies, that is core, MOS and bipolar with distinct characteristics in access time, capacity and cost.

The main memory acquisition cost for system i, using architecture j, is defined as: [29,30]

$$MM_{ij} = C_b(b_{ij}\ PM_i + DM_i) \qquad \text{(VII-9)}$$

where:

MM_{ij} = main memory acquisition cost (in dollars)

b_{ij} = static storage ratio

PM_i = main memory (in bits) required for program i: M_i is derived from system requirements; P is estimated fraction of M_i dedicated to program storage vs. data storage.

DM_i = main memory (in bits) required for data storage in system i; M_i is derived from system requirements; D is estimated fraction of M_i dedicated to data storage vs. program storage.

C_b = cost per bit of main memory. Examination of the price per bit of recent militarized equipment indicates an average cost of 4 cents per bit in 1976; i.e., $5,000 per 16,000 by the memory module. This value in 1979 in the civilian sector is already .2-.4¢/bit and is expected to be reduced by a factor of 10 (that is 0.02-0.04 cents per bit) in 1985.

(3) Secondary Memory Acquisition Cost. Inherently characterized by slower access time and enormous storage capacity, secondary memories are important adjuncts to a computer/processor. As expected, secondary memories have lower acquisition cost.

The secondary memory acquisition cost for system i, using architecture j, is defined as: [29,30]

$$SM_{ij} = C_a(b_{ij}\ P'Ma_i + D'Ma_i) \qquad \text{(VII-10)}$$

where:

SM_{ij} = secondary memory acquisition cost (in dollars)

b_{ij} = static storage ratio

$P'Ma_i$ = secondary memory (in bits) required for program storage in system i; Ma_i is derived from system requirements, whereas P' is the estimated fraction of Ma_i, used for program storage vs. data storage.

$D'Ma_i$ = secondary memory (in bits), required for data storage in system i; Ma_i is derived from system requirements, while D' is the estimated fraction of secondary memory, used for data storage vs. program storage.

C_a = cost per bit of secondary memory

(See Figures VII-6 and VII-7)

Examination of the price per bit of current militarized disc systems indicates an average cost of 0.2 cents per bit, e.g., a 36 Mbit disc system costs $72,000. This value

is used in 1976 cost estimates; a cost reduction of 10:1 in the next 10 years is assumed, so a price of 0.02 cents per bit is foreseen in 1985 cost estimates (see also Figures VII-6 and VII-7). [29]

(4) Measure of Performance Costs. [29,31] The processor speed a_{ij} of equation VII-7 and the static sotrage ratio b_{ij} of equation VII-9 above attempt to capture the ability of the jth architecture relative to the system i. They are derived by measuring the performance of the architecture j on benchmark test programs and by estimating the relative importance or weight of each of these programs to computation characteristics of each system. The performance of an architecture on test program is characterized in what are called the S measure (measure of Space) and the M and R measures (measure of Execution Time) where:

S_{kj} is a measure of the amount of memory (in 8 bit bytes) needed to represent test program k on architecture j.

M_{kj} is a measure of the processor/memory transfers required to execute test program k, when using architecture j.

R_{kj} is a measure (in 8-bit bytes) of the number of internal register-to-register transfers required by the processor to execute test k on architecture j.

For the Military Computer Family (MCF) project twelve benchmark test programs, as mentioned in Appendix G, were used for the derivation of a_{ij} and b_{ij}.

The relevancy of the kth test program to the ith system is given by factors W_{ik}, which were obtained by first dividing benchmark programs into two categories: 1) programs that relate principally to I/O, and 2) programs that are associated with traditional processor/memory functions. Within these categories, varying degrees of functional overlay occur among the test programs. Initially a gross value was estimated for each category; subsequently this value was distributed across the programs of the category.

The processor speed ratio a_{ij} and the static storage ratio b_{ij} were obtained by combining the above quantities as follows:

$$a_{ij} = \frac{3 \sum_{k=1}^{12} W_{ik}(3M_{kj} + R_{kj})}{\sum_{m=1}^{3} \sum_{k=1}^{12} W_{ik}(3M_{km} + R_{km})} \qquad \text{(VII-11)}$$

$$b_{ij} = \frac{3 \sum_{k=1}^{12} W_{ik}(3M_{kj} + R_{kj})}{\sum_{m=1}^{3} \sum_{k=1}^{12} W_{ik}\, S_{km}} \qquad \text{(VII-12)}$$

2. Applications Software Life Cycle Cost

a. General

Cost estimation for large-scale computer software has been traditionally a risky undertaking. The reasons can

be found in the complexity of the systems programming effort, coupled with the rapid change in technology and applications. In accordance with F.P. Brooks, Jr. [33] rule of thumb estimation:

* The promotion of a simple debugged program to a generalized, tested, documented and maintenable programming product (so that anyone can use, fix and extend) costs three times as much as the debugged program itself.

* The promotion of the same program to a programming system, which is a collection of interacting programs, coordinated in function and disciplined in format, so that the assemblage constitutes an entire facility for large tasks, costs at least three times the program itself.

* The promotion of the same program to a programming system product, which differs from the simple program in all of the above ways costs nine times as the program itself.

Figure VII-8 illustrates the evolution of the programming system product.

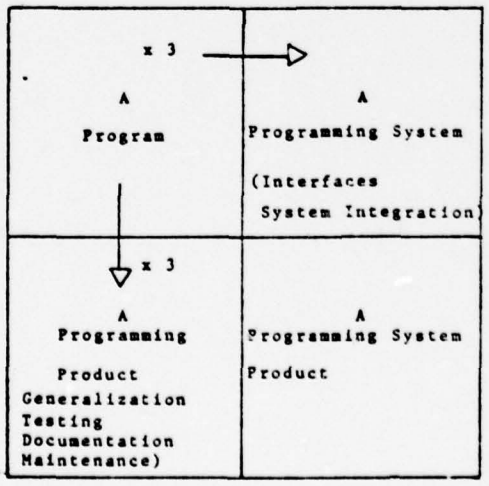


Figure VII-8: Evolution of the programming systems product.

Traditional cost estimating methods have used one or more of the following techniques: [34]

* Top Down Estimating

The estimation of the cost of all, or large portions, of the project to be estimated relies on the corresponding costs of previous projects, that have been completed. The disadvantage is that there is 1) substantial risk of overlooking special or difficult technical problems that may be buried in the project tasks and 2) lack of details needed for cost justification.

* Similarities and Differences Estimating

The jobs to be accomplished are broken to a level of detail, where the similarities to and differences from previous projects are most evident. Work units which cannot be compared are estimated by some other methods.

* Ratio Estimating

The estimation relies on the sensitivity coefficients or exchange ratios, which are invariant, within limits, to the details of design. The software analyst estimates the size of a module by its number of object instructions, classifies it by type and evaluates its relative complexity. An appropriate cost matrix is constructed from a cost data base in terms of cost per instruction for that type of software, at that complexity level. Other ratios, empirically derived, can be used in the total estimation process e.g., CPU time per instruction, peripheral usage to CPU usage etc. The method is simple, fast, convenient in the proposal environment and beyond. It suffers,

as all methods do, from the need for a valid data base for many estimating situations (business vs. scientific, real time vs. non-real time, operational vs. non-operational).

* Bottom-Up Estimating

This is the technique used mainly when estimating Government research and development contracts. The total job is broken down into relatively small work packages and work units, until it is reasonably clear what steps and talents are involved in doing each task. Each task is then estimated and the costs are summed-up to form the total project cost. An advantage is that the job of estimating, can be distributed to the people, who will do the work. A difficulty is the lack of immediate perspective of the most important parameter of all: the total cost of the project. In software engineering cost estimates the top-down estimation is used as a check on the bottom-up technique.

b. The TRW Software Cost Estimation Algorithm [34]

TRW has developed a cost estimation algorithm, based on the assumption that costs vary with the number of instructions. For each identified routine, the procedure combines an estimate for the number of object instructions, category, relative degree of difficulty and historic data in dollars per instruction from the cost data base to give a trial estimate of the total cost.

The first step in the method is to categorize the software routines, which were considered in the preliminary

design phase. Categories which have stood the test of usage in several proposal and preliminary design are:

* Control routine (C), which controls execution flow and is non-time critical.

* Input/Output routine (I), which transfers data into or out of the computer.

* Pre or post-algorithm processor (P), which manipulates data for subsequent processing or output.

* Algorithm (A), which performs logical or mathematical operations.

* Data management routine (D), which manages data transfer within the computer.

* Time critical processor (T), which is highly optimized machine dependent.

The next step is size and complexity estimates by routine, or subprogram by the designer. To balance cost and risk, the designer may decide to use software elements, which are available to him from a software algorithm and needs only some degree of modification or adaptation. To account for the degree of difficulty of a given kind of routine, the designer estimates a risk or complexity factor. This is the most crucial step in the estimating process, for it establishes the cost of the routine with all direct and indirect charges amortized against it. The other steps determine how the total cost wil be spread over the development cycle. The simplest technique for narrowing down the many subjective choices early in the

estimation process is to ask, if the routine is new or old on one hand and if it is easy, medium or hard on the other. Therefore six levels of difficulty for each routine are realized as follows:

	Easy	Medium	Hard
Old:	OE	OM	OH
New:	NE	NM	NH

The only parameter that changes, as a function of degree of difficulty (OE through NH) is cost per instruction.

The third step is to identify the various software development phases from the conceptual stage to delivery of the operational software to the user. For each phase an estimate is required for the fraction of the total amount to be allocated to it.

The fourth step is to define the activities in each development phase by means of an activity array and associated cost matrix. The activities as a function of development phase is a Number of activities times the number of phases matrix. A typical activity array is illustrated in Figure VII-9, whereby the corresponding cost matrix is shown in Figure VII-10.

The final step in setting up the initial conditions for the cost estimation algorithm, is to provide schedule data based on the customer's statements of work, or other management considerations. Schedule data is input as months from go-ahead for each of the milestone periods. Burden rates are input for projected overhead rates, general and administrative

and so forth. Labor mix is defined and unburdened bid rates for the labor grades, desired for the phase, is defined. Other direct changes are input, which for software is typically travel as a percentage of direct labor costs, such as 3 per cent and documentation at 10 per cent.

Computer usage data for a CDC 6500 class with time shared central processor unit based on sampled data from a programming department by month has been:

No. of MTS	Total No. of processor units used	No. of processor hours per MM	Peripheral hours per MM
36	55.6	2.20	7.6
35	31.0	1.27	4.4
35	30.0	1.30	4.5
33	46.5	2.02	7.0

If the computer was a 370/155, a figure of 3 hours per man per programmer man-month would be reasonable. At an average productivity of 1 object instruction per hour, this is equivalent to 156 instructions per man-month, or 1.2 minutes per instruction. The data in the third column are based on 1.42 processor units, corresponding to one hour of computing time. A computer hours matrix in terms of usage rate per instruction by phase is shown in Figure VII-11.

The summary output from the cost estimation algorithm includes:

* cost per routine, based on historic or proposed burden rates.

ACTIVITY \ DEVELOPMENT PHASE		PHASE A	PHASE B	PHASE C	PHASE D	PHASE E	PHASE F	PHASE G	PHASE H
		PERFORMANCE AND DESIGN REQUIREMENTS	IMPLEMENTATION CONCEPT AND TEST PLAN	INTERFACE AND DATA REQUIREMENTS SPECIFICATION	DETAILED DESIGN SPECIFICATION	CODING AND AUDITING	SYSTEM TESTING	CERTIFICATION AND ACCEPTANCE	OPERATIONS AND MAINTENANCE
MANAGEMENT	1	PROGRAM MANAGEMENT	PROGRAM MANAGEMENT	PROGRAM MANAGEMENT	PROGRAM MANAGEMENT	PROGRAM MANAGEMENT	PROGRAM MANAGEMENT	PROGRAM MANAGEMENT	PROGRAM MANAGEMENT
	2	PROGRAM CONTROL	PROGRAM CONTROL	PROGRAM CONTROL	PROGRAM CONTROL	PROGRAM CONTROL	PROGRAM CONTROL	PROGRAM CONTROL	PROGRAM CONTROL
REVIEWS	3		PRELIMINARY DESIGN REVIEW (PDR)	INTERFACE DESIGN REVIEW (IDR)	CRITICAL DESIGN REVIEW (CDR)	SYSTEM TEST REVIEW (STR)		ACCEPTANCE TEST REVIEW (ATR)	OPERATIONAL CRITIQUES
DOCUMENTS	4	DOCUMENT AND EDIT	DOCUMENT AND EDIT	DOCUMENT AND EDIT	DOCUMENT AND EDIT	DOCUMENT AND EDIT		DOCUMENT AND EDIT	DOCUMENT AND EDIT
	5	REPRODUCTION	REPRODUCTION	REPRODUCTION	REPRODUCTION	REPRODUCTION		REPRODUCTION	REPRODUCTION
REQUIREMENTS AND SPECIFICATIONS	6	REQUIREMENTS DEFINITION	PDR MATERIAL	EVENT GENERATION INTERFACE	PRODUCT CONFIG DETAILED TECH DESCRIPTION (PART II) (WITHOUT LISTINGS)	TECHNICAL DESCRIPTION UPDATE	PRODUCT CONFIG DETAILED TECH DESCRIPT UPDATE (PART II) (WITH LISTINGS)	REQUIREMENTS CERTIFICATION	SOFTWARE PROBLEM REPORTS (SPR)
	7	REQUIREMENTS ALLOCATION	PERFORMANCE AND DESIGN REQUIREMENTS (PART I)	COMMAND DEFINITION I/F		TRAINING DOCUMENTATION		FINAL DOCUMENTATION UPDATE (PART II)	
	8			TELEMETRY DEFINITION I/F			INTERFACE SPECIFICATIONS UPDATE		
	9			OPERATIONAL ENVIRONMENT I/F					
DESIGN	10	TRADE STUDIES	TRADE STUDIES	TRADE STUDIES	TRADE STUDIES				
	11	INTERFACE REQUIREMENTS	FUNCTIONAL DEFINITION	DATA DEFINITIONS	ALGORITHM DESIGN	ALGORITHM UPDATE	PROGRAM UPDATE		
	12	HUMAN INTERACTION	STORAGE AND TIMING ALLOCATION		PROGRAM DESIGN				
	13	STANDARDS AND CONVENTIONS	DATA BASE DEFINITION	DATA BASE DESIGN		DATA BASE UPDATE	DATA BASE UPDATE		
	14		SOFTWARE OVERVIEW (PRELIMINARY)		SOFTWARE OVERVIEW UPDATE				
CODING	15				PROTOTYPE CODING	OPERATIONAL CODING			
TESTING CONFIGURATION CONTROL AND C&A	16		PRODUCT AND CONFIGURATION CONTROL	PRODUCT AND CONFIGURATION CONTROL	PRODUCT AND CONFIGURATION CONTROL	PRODUCT AND CONFIGURATION CONTROL	PRODUCT AND CONFIGURATION CONTROL	PRODUCT AND CONFIGURATION CONTROL	PRODUCT AND CONFIGURATION CONTROL
	17		DATA BASE CONTROL	DATA BASE CONTROL	DATA BASE CONTROL	DATA BASE CONTROL	DATA BASE CONTROL	DATA BASE CONTROL	DATA BASE CONTROL
	18	TEST REQUIREMENTS	TEST PLANS	INTERFACES	TEST PROCEDURES	DEVELOPMENT TESTING PLANNING	SYSTEM TEST PLANNING	ACCEPTANCE AND TEST PLANNING	INTEGRATION TESTING
	19					DEVELOPMENT TEST	SOFTWARE SYSTEM TEST	ACCEPTANCE DEMONSTRATION	SPR CLOSURE
	20						HARDWARE/ SOFTWARE SYSTEM TESTING		
	21	ACCEPTANCE TEST REQUIREMENTS	QUALITY AND RELIABILITY ASSURANCE PLANS	QA AND RA MONITORING	QA AND RA MONITORING	Q AND RA MONITORING	Q AND RA MONITORING	Q AND RA MONITORING	Q AND RA MONITORING
	22					TEST SUPPORT	TEST SUPPORT	TEST SUPPORT	TEST SUPPORT
	23								
OPERATIONS	24		OPERATIONAL CONCEPT	OPERATIONAL TIMELINE	OPERATIONAL CONCEPT UPDATE	USER'S MANUAL (PRELIMINARY)	OPERATIONAL TIMELINE UPDATE	USER'S MANUAL UPDATE	INTEGRATION SUPPORT
	25		TRAINING PLAN		TRAINING PLAN UPDATE	TRAINING	TRAINING	TRAINING AND REHEARSAL	TRAINING AND REHEARSAL

Figure VII-9: Activities as a function of Software Development Phase.

* average cost per instruction, number of instuctions and category.

* simple graphic display of schedule and events.

* cost breakdown by development phase per routine in total dollars and per cent.

* cost breakdown by activity, by routine and summed up over all routines in the software, such as management, review, documentation, specification, design, coding, and testing.

* manloading and cost summary by segment showing for each month the labor breakdown for senior staff, staff, technical, clerical; also computer hours by month, other direct charges, cost by month and cumulative costs.

The outputs from the cost estimation algorithm are considered a "trial set" for the cost estimation group. The trial set is used in combination with all other sources of data to test cost position against objectives. The approved trial set, which contains the best judgemental and quatitative measures of cost per software element and per activity, becomes the input to the official pricing computer run. The official cost figures are produced by cost guidelines, approved rates and procedures, which have been established by the in-house pricing group and approved by the Government auditor.

This cost estimation approach has the following advantages:

* the amount of data required to obtain a cost estimate is minimal.

ACTIVITY	PHASE A (8)	B (10)	C (3)	D (14)	E (23)	F (21)	G (12)	H (8)
1	10	6	8	6	7	5	10	2
2	8	3	3	3	3	3	3	6
3		6	4	6		3	8	6
4	13	8	5	6	5	6	4	2
5	5	2	2	3	3	2	2	1
6	22	8	7	12	3	7	5	6
7	10	8	7		2		6	
8			7			5		
9			6					
10	17	10	10	8				
11	2	10	10	9	7	8		
12	2	5		13				
13	4	7	10		3	5		
14		4		3				
15				5	25			
16		4	4	5	4	4	10	10
17		3	5	5	5	5	8	10
18	5	6	3	5	5	2	5	10
19					10	15	14	5
20						10		
21	2	4	5	5	7	9	9	3
22					5	6	6	4
23								
24		4	2	2	3	5	3	25
25		2		1	2	3	5	10

Figure VII-10: Cost matrix data showing allocation of resources as a function of activity by phase (category P).

PHASE	CATEGORY C	I	P	A	D	T
A	0	0	0	0	0	0
B	0	0	0	0	0	0
C	0	0	0.0017	0.0007	0.0007	0.01
D	0.0017	0.0017	0.0017	0.0017	0.0017	0.01
E	0.007	0.005	0.008	0.007	0.007	0.04
F	0.008	0.005	0.01	0.007	0.007	0.04
G	0.005	0.004	0.005	0.004	0.004	0.005
H	0	0	0	0	0	0

EXAMPLE: THE COLUMN SUM OF CATEGORY C, CONTROL ROUTINE, IS 217×10^{-4} HRS/INSTRUCTION, OR 1.4 MIN/INSTR. THE TOTAL RESOURCE REQUIRED IS THEN DISTRIBUTED OVER PHASES D THRU G AS SHOWN ABOVE, e.g., 0.42 MIN/INSTR FOR PHASE E, CODING AND AUDITING.

Figure VII-11: Computer hours matrix, showing computer usage allocation as a function of phase.

* the software designer immediately sees the cost consequences of his design.

* to the extent the routine is directly traceable to a requirement, the requirement is directly traceable to a total cost.

* comparisons or alternate schedules or resource allocations can be made rapidly.

* cost justification is produced, which can stand the test of a Government audit.

c. The Military Computer Family (MCF) Application Life Cycle Cost

The applications software life-cycle cost for system i, using architecture j is defined as: [29,30]

$$ASw_{ij} = Cs_i \; S_i \; L_s \qquad \text{(VII-13)}$$

where:

C_{si} = cost (in dollars) per instruction of applications software for architecture j

S_i = applications software size (in instructions), derived for the proposed system's data (see Table VII-I)

L_s = applications software life cycle cost factor, i.e., ratio of applications life-cycle cost to initial acquisition cost (=5.5 presently).

Figure VII-5 showed a strong correlation between total dollars investment and the cost-per-instruction for software. This correlation is even stronger, when cost per instruction is related to the Tool Availability Index (TAI): [29,30]

Table VII-I: Proposed System's Data

Sys. #	System Mission	n_i	Mr_j (MIPS)	*PM_i	*DM_i	*PMa_i	*DMa_i	S_i**
1	Medium Search	192	1.33	.414	.101	1.9	7.7	32
2	Medium Command & Control	27	.26	1.638	.412	2.2	8.8	144
3	Small Search	100	1.00	.512	.128	2.8	11.2	20
4	Large Command & Control	178	.20	13.600	3.400	4.0	16.0	375
5	Medium Command & Control	64	.50	1.600	.400	15.8	63.2	47
6	Large Command & Control	30	.40	3.321	.820	8.4	33.6	250
7	Small Command & Control	832	.75	.618	.152	.4	1.6	100
8	Large Communications	616	.18	3.200	.800	13.4	52.0	175
9	Small Communications	800	.16	.116	.031	0	0	8
10	Small Communications	9	.53	.408	.102	3.2	12.8	83
11	Small Special Purpose	30	.48	.328	.082	.1	.3	14
12	Large Data Management	16	.02	1.600	.400	573.4	2293.6	324
13	Medium Search	50	.35	3.712	.928	3.2	12.8	28
14	Medium Data Management	8	.80	3.200	.800	1912.0	7648.0	1
15	Small Guidance & Control	3325	.20	.006	.002	0	0	1

* P and D are fractions applied to the proponent's stated memory requirements which reflect an estimate of memory used for programs (P) vs. data (D). Values are expressed in megabits.

** S_i stated in 10^3 instructions.

$$C_{TAI} = 1138\ (TAI)^{-1.07} \qquad \text{(VII-14)}$$

$$r^2 = 0.99$$

where: TAI = Ratio of available software tools to the ideal set of software tools. The "ideal set" has been defined for the MCF Project. By knowing the TAI for a given architecture for any point in time, one can obtain from Figure VII-12 [30] an estimated cost per instruction.

r^2 = correlation coefficient

However TAI is not as easily evaluated as investment inventory and the relationship of Figure VII-5 is more usefully applied.

MCF provides an evaluation of the costs of each element of the TAI elements, which are presented in Table VII-2.

d. A Cost Model Proposed by L. H. Putnam [35]

Putnam has examined the problem of an early sizing, cost and schedule estimates for an application software system and has proposed the following software equation:

$$S_s = C_k\ K^{1/3}\ t_d^{4/3} \qquad \text{(VII-15)}$$

where:

S_s = size of project in source statements

C_k = state of technology constant

K = life cycle effort in man-years = $D\ t_d^3$ where D is the magnitude of the difficulty gradient, empirically found to be related to

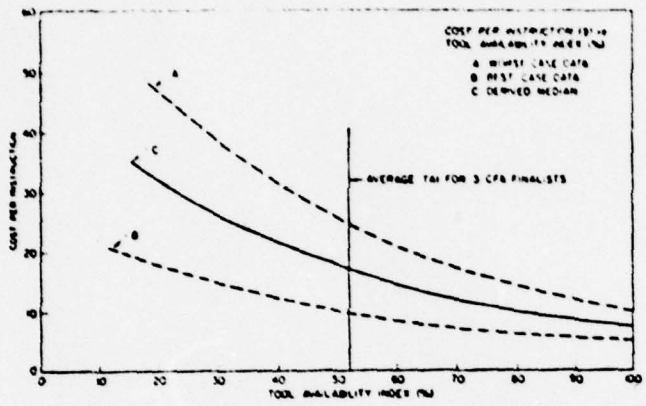


Figure VII-12: Availability Index (%)

REMARKS:

1. The curves show the variation of cost per instruction as a function of the Tool Availability Index (TAI) for three conditions: (a) worst case, (b) best case and (c) derived median.
2. As the percentage of available software tools increases, the cost per instruction of application software can be seen to diminish.

Table VII-II: Software Tool Costs

	Software Tool	Cost (1977)
1.	Noninteractive Debugging Aids (CMS-2)	$50K
2.	Language-Independent Monitor	210K
3.	Compiler + Cross-Compiler (CMS-2)	3300K
4.	Real-Time + Time-Sharing Operational System	3500K
5.	Computers + Cross-Compilers (TACPOL)	1000K
6.	Test Case Instrumenter + Analyzers (TACPOL)	280K
7.	Noniteractive Debugging Aids (TACPOL)	50K
8.	Language-Dependent Monitor (Assembler)	50K
9.	Test Case Instrumenter + Analyzers (CMS-2)	280K
10.	Assembler	135K
11.	Macro Assembler	800K
12.	Basic Linker	130K
13.	Simple Overlay Linker	210K
14.	Interactive Debugging Aid (Assembler)	300K
15.	Test Data Auditor	140K
16.	Test Data Generator	350K
17.	Integrated Library	100K
18.	Data Base Management System	4200K
19.	Text Processing System	630K
20.	Interactive Source Editor	100K
21.	General Purpose System Simulator	700K
22.	Instruction Simulator	350K
23.	Reformatters (Fortran)	110K
24.	Test case Instrumenters + Analyzers (Fortran)	280K
25.	Compilers + Cross-Compilers (Fortran)	1000K
26.	Noninteractive Debugging Aids (Fortran)	50K
27.	Extended Overlay Linker	500K

system development characteristics measuring the degree of concurrency of minor task accomplishments.

t_d = development time in years

Equation VII-15 is derived from the Norden/Raleigh distribution equation:

$$\dot{y} = \frac{K}{t_d^2} t e^{-\frac{t^2}{2t_d^2}} \quad \text{in MY} \qquad \text{(VII-16)}$$

where: y = manpower at any time t

K = area under the curve and is the nominal life cycle effort in man-years

t_d = the time of peak manpower in years and corresponds very closely to the development time for the system.

The state-of-technology constant C_k can be determined by calibration against the software equation, using data from projects developed by the same software house, using similar technology and methods.

Figure VII-13 illustrates a parametric form of equation VII-15.

(1) Initial Project Sizing

Given the broad preliminary size of the project, PERT technique is applied to get an overall system size range and distribution as follows:

* An estimate of the expected value of a beta distribution is:

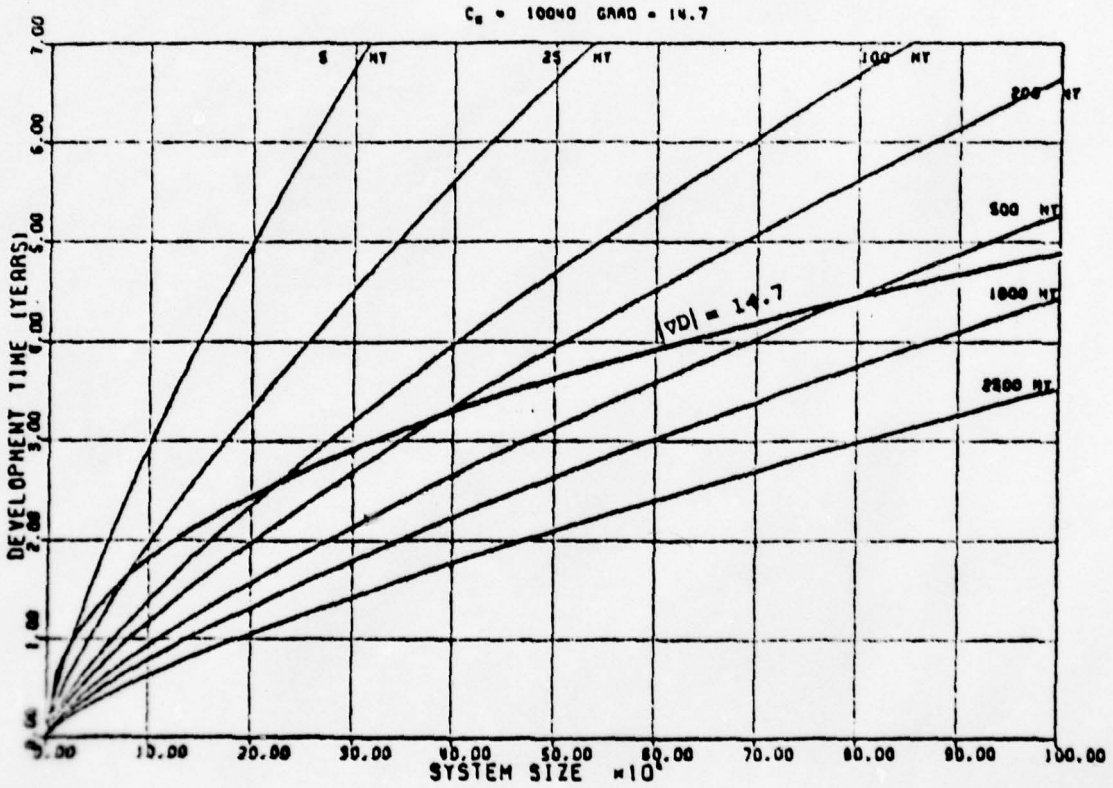


Figure VII-13: Size - Effort - Time Trade-off Chart.

$$E_i = \frac{a + 4m + b}{6}$$

where:

a = smaller possible project size estimation

m = most likely project size estimation

b = largest possible project size estimation

The overall expected value is the sum of all the individual expected values (major functions)

$$E = \sum_{i=1}^{N} E_i$$

* An estimate of the standard deviation is given by:

$$\sigma_i = \frac{b - a}{6}$$

and the overall standard deviation estimation by:

$$\sigma_{tot} = \left(\sum_{i=1}^{N} \sigma_i \right)$$

(2) Development time-effort determination
Equation VII-15 is solved for K

$$S_s = C_k K^{1/3} t_d^{1/3} = C_k (K t_d)^{1/3}$$

$$K t_d^4 = \left(\frac{S_s}{C_k}\right)^3$$

$$K = \frac{\left(\frac{S_s}{C_k}\right)^3}{t_d^4} \quad \text{in man-years (MY)}$$

In terms of development effort E = 0.4 K, so

$$DE = 0.4 \frac{\left(\frac{S_s}{C_k}\right)^3}{t_d^4} \text{ in MY}$$

The development Effort Cost (DEC) is given by:

$$\$LCC = \frac{\$}{M\ Y} K$$

or

$$\$DEC = \frac{\$}{M\ Y} 0.4K$$

(3) Manpower and Cash Flow Pattern

With the parameters for Development Effort (DE) and Development Time (t_d) known, the generation of the manloading and the cash flow pattern for the software development period can be estimated, by using the Raleigh/Norden equation which gives the instantaneous manpower, that is equation VII-16.

The cash flow is just the average dollar cost/ MY times $\dot{y}$:

$$\text{Cash Flow} = \frac{\bar{\$}}{M\ Y} \dot{y} \quad \text{in \$/YR}$$

VIII. ALTERNATIVES FOR THE HELLENIC NAVY

The alternatives for computer systems acquisition for the Hellenic Navy (HN) can be describes as follows:

* Acquire already developed systems from U.S. Navy or other NATO navies or countries.

- Description:

This acquisition is meant to include:

i) Hardware and software packages of existing systems.

ii) Tailoring of the systems to the specific needs of the HN by the vendor in consultance with the HN.

iii) Attainment of support capability by the HN.

It is understood that vendor will continue to provide certain agreed field changes in hardware and software.

- Strengths:

This alternative will give the HN the capability to operate and support selected systems, exploiting the experience of the developer navies or countries.

- Weaknesses:

This alternative has the following weaknesses:

i) It represents the most expensive approach, since all technology involved has to be transferred from the vendor each time.

ii) Restricts the active participation of the HN only to certain portions of the system application (statement of the functional needs, support).

* Acquire certain hardware and software packages and develop others to complete a system.

- Description:

This approach is meant to include acquisition of certain already developed packages of hardware (e.g., displays) and software (e.g., ballistic algorithms) and develop others required to complete a desired system.

- Strengths:

This alternative will give the HN the capability to operate and support desired systems by integrating existing technology complemented by developed technology.

It is a challenging approach, which may enhance the technological capability of the HN and reduce costs significantly.

- Weaknesses:

It seems that there are no serious weaknesses for the application of this approach. The risks involved are rather of lesser significance, especially when undertaking small scope development projects.

* Generate own capability to develop systems.

- Description:

This alternative is meant to include generation of a capability to develop desired systems from conception to operational and support. All software required is to be developed by the HN. Hardware acquisition is to be effected through competitive procedures.

- Strengths:

This approach is very dynamic and will provide the HN with the capability to operate and support systems through system development efforts carried out solely by the HN. The benefits of this approach cannot be overemphasized. The HN will be in a position to create its own desired systems.

- Weaknesses:

This alternative is desirable as a long range objective, but may not be feasible in the near future.

IX. SUMMARY AND CONCLUSIONS

From the examination of Sections III to V, it is concluded that a serious effort is continuously spent by countries of the North Atlantic Alliance towards the evolution of the Command and Control systems they have developed, and the improvement of the Communications, Command and Control Management. The Alliance itself will benefit from the individual nations advances, but the real improvement will come when the cooperation between Europe and North America in the C^3 area, is eventually facilitated through the so-called "two-way street" concept.

The examination of Section VI provides a good measure of the complexities involved in the development of a major computer system application. Systems analysis is the answer to these complexities and should be always applied when such a development is about to be undertaken.

From the cost considerations of Section VII, it is concluded that the advancing technology allows the production of very effective and reliable hardware at low cost. This will continue to be the case in the future. On the other hand, development of software is presently a rather expensive undertaking. Maintenance of the developed software is even more expensive. Software will continue to be expensive until software engineering, maturing with time, solves the existing software problems and establishes itself on a better scientific foundation.

The examination of Section VII shows also that there are methods to quantify software sizing, cost and development effort and this can provide the answers required by the management.

Section VIII provides a possible range of alternative approaches for the Hellenic Navy towards the acquisition of computer systems to fulfill her needs. All of the approaches foresee an active involvement of the HN in the acquisition/ development process. This involvement increases as we proceed from approach one to approach three.

X. GLOSSARY OF TERMINOLOGY

ALGORITHM - A prescribed set of well-defined rules or processes for the solution of a problem in a finite number of steps. In principle, the steps are sufficiently basic and definite, so that a human can conceptually carry out the prescribed steps exactly even though the time to do so is impractically long. [36]

APPLICATION PROGRAM - A computer program which directly contributes to the processing of end work, as opposed to computer systems program, language processors and other utility programs.

ARCHITECTURE OF A COMPUTER - The design personality of a computer. It can be viewed from two aspects: 1) hardware aspect and 2) software aspect.

The hardware engineer may view the computer from the hardware aspect, that is by the technology involved; LSI, CMOS, bipolar, core, pipelining, DMA, data bus, word size, etc., are all terms that may describe one processor as opposed to another.

The software engineer may describe computer architecture by the attributes of a system, as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls, the logical design and the physical implementation.

AUDIT - A formal or official examination that attests to the conformity (or non-conformity) between two supposed equivalent entities according to a predefined set of rules. [36]

AUTOMATIC DATA PROCESSING SYSTEM (ADPS) - A type of computer system used to support administrative or management systems. [18]

BATCH PROCESSING - A mode of computer processing, which is characterized by the concurrent availability to the computer of a complete set of input data for a given job to be processed, the execution of which is not controlled in real-time (i.e., on-line) by a user.

BENCHMARK - A set of computer programs and associated data tailored to represent a particular workload, which is representative of the user's application and used to test the capability of a computer system to perform that workload within a predetermined limit. Benchmarks are used for evaluating vendor computer systems.

BOTTOM-UP PRINCIPLE - A synthesis from concrete, low-level details by stepwise integration into higher-level capabilities or more abstract concepts as a method for solving a problem. [36]

CHECKOUT - Informal validation of a program or a part of program by members of the development team, once the item has been successfully compiled (or assembled), by running a series of test. When such validation is performed only visually, the activity is referred as "desk checking". [36]

CODING - The activity of expressing the steps of a given algorithm in a computer language (or, perhaps more than one language). A unit is not qualified as "coded" until compiled (or assembled) and all syntax errors removed. [36]

DATA - Representations of measurements, observations, facts, statistics, or derived quantities, either actual believed or assumed, in a form suitable for communication, reorganization, storage, retrieval, processing and dissemination. Contrast with INFORMATION. [36]

DATA BASE - A collection of mutually related data items, usually stored together, to serve one or more applications by end users. As a goal, the data base has little (or not at all) controlled redundancy, is stored so as to be independent of the usages made of its contents, yet is stored for optimal efficiency by such applications and is capable of being subjected to a common and controlled approach, when adding new data or modifying or retrieving data existing within the data base. A "data base system" is a collection of separately structured data bases united by regulated interaction to form an organized whole. [36]

DATA BASE DESIGN AIDS - Tools assisting data base designers in grouping data elements into logical record classes and in determining the relationships among logical record classes implicit in either the nature or the usage of the data.

DATA BASE MANAGEMENT SYSTEM - Allows the use of a computer system to define the contents of and the logical relationships between collection of data items that represent some useful abstraction of a real-world phenomenon (tactical command and control system, the modules and documentation of a system of

computer programs) without being concerned with the physical mechanics of storing, locating and retrieving items or groups of items.

DATA DICTIONARY SYSTEMS - Tools assisting data base designers in managing the data definition activities.

DATA MANIPULATION UTILITIES - Allow the system user to alter the form and content of data files independent of the logical significance of the data fields involved. Specific tools are Sort/Merge programs and Editors (Interactive Source Language Editors, Interactive Object Module Editors, Batch Source Language Editors and Batch Object Module Editors). See also UTILITIES.

DEBUGGING - Detection, location and repair of inconsistencies between the program response and its functional or programming specification. A program or procedure is said to be debugged if no known anomalies are present. [36]

DEBUGGING AIDS - Assist the programmer in locating the sources of program errors that have been discovered during subsystem testing, usually by giving him some control over the execution of the module under test, that is external to the normal program code. Specific tools are Interactive Symbolic Debuggers, Non-Interactive Symbolic Debuggers, Interactive Absolute Debuggers and Non-Interactive Absolute Debuggers.

DOCUMENTATION AIDS - Assist in the preparation and maintenance of documentation about the modules of a system. Specific tools are Text Processing System, Flowchart Construction Languages and Automatic Flowcharters.

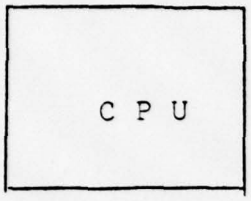
EMBEDDED COMPUTER SYSTEM - A computer system, that is "physically incorporated in a larger system, whose primary function is not computation. An electromechanical devices, a combat weapon system, a tactical system, an aircraft, a ship, a missile, a spacecraft, a command and control system, or a communication system are examples of larger systems.

Embedded computer systems also include the support software for their own design, development and maintenance. Computers used primarily for data processing, scientific, or research application are not normally included among embedded computer systems.

FIRMWARE - It is programmed software. It is software merged into hardware because it combines programming with nonalterable hardware. For example a microprogram (a program residing in ROM), because it is unalterable is called firmwave.

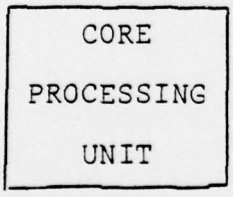
HARDWARE - Consists of the physical devices of a digital computer system, by means of the action on their circuitry on coded representations of which, the processing on information is accomplished.

These devices are: [42]



Central Processing Unit (CPU) (on mainframe)

* The engine that drives the whole system.
  Main characteristic: Fast
* Processes information. All operations fall in three simple classes:
  - move information around
  - do arithmetic
  - perform logical operations
* Controls operations of itself and all attached equipment, obeying human's computer "programs".
* Human operator runs it from "control panel" or "control console".



Core Memory

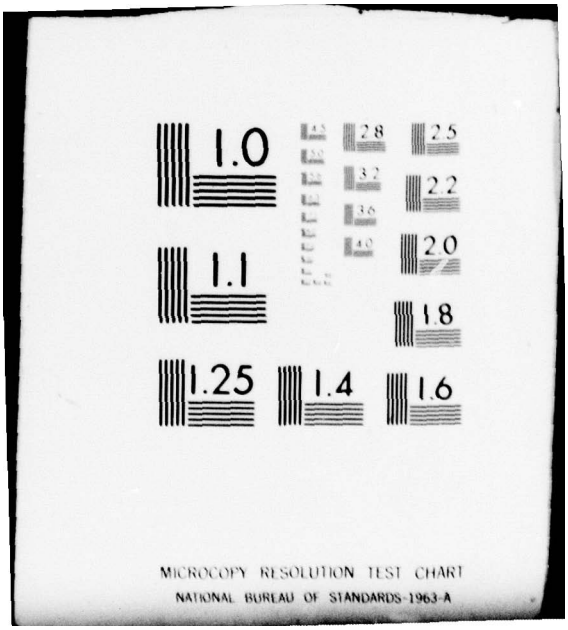
* Core Memory is the internal information storehouse of the computer.
* Known as "real" memory as opposed to "Virtual Storage" (VS).
* Core memory is fast and expensive, but is getting cheaper.
* Basic storage unit, called a "byte" (eight bits) stores two hexadecimal digits (0,1, ...9,A,B,...F) letters (A-Z) and other symbols ($,¢,*,etc.).

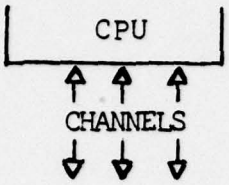
AD-A074 444 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2
A STUDY OF ALTERNATIVES FOR COMPUTER SYSTEMS ACQUISITION FOR TH--ETC(U)
JUN 79 V A NAOUM

UNCLASSIFIED NL

3 OF 3
AD A074444

END
DATE FILMED
10-79
DDC


1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



* Storage in core memory is located by "addressees".

Channels

* Paths for information going to/from CPU.
* Devices attached to CPU to communicate electronically with it via these channels.

FILE - A collection of related information units (records), which contain data in a particular state: sorted on a particular key, for example item name. Files are normally kept on secondary computer-storage devices.

INFORMATION - A representation of knowledge, intelligence, or other meaningful data in a form that can be used to cause or modify the purposeful action of humans or machines, perhaps as a result of proper organization, analysis and presentation. Contrast with DATA. [36]

INFORMATION RETRIEVAL SYSTEMS - General purpose application programs operating either on-line (interactively) or in batch, that interpret user requests to locate and display information that is stored either within a structured data base, or within separate files. Specific Tools are Query Language Systems and Report Writers.

INFORMATION STRUCTURE - A representation of the elements of a problem or of an applicable solution method for a problem, insofar as its information base is concerned.

INFORMATION SYSTEM - An assemblage of methods, techniques, procedures, programs, or devices that sense, convey, store,

process, retrieve, or disseminate information, united by regulated interaction, to accomplish an organized purposeful task.

INSTRUCTIONS - The repertoire of a language or (virtual) machine.

INTERFACE - When applied to a module, that set of assumptions made concerning the module by the remaining program or system, in which it appears. Modules have control, data and services interfaces. [36]

INTERFACE TESTING - Validation that a module or set of modules operate within agreed interface specifications to assure proper data and logical communications.

INTEROPERABILITY - It is the ability of systems, units or forces to provide services to and to accept services from other systems, units and forces and to use the services, so exchanged, to enable them to operate effectively together. It includes compatible command and control systems, procedures, weapons systems, airborne and ground-based navigational aids, electronics, communications, procedures and equipment for identifying friend from foe, and cross servicing facilities.

LEVEL OF ACCESS - A set of functions, macros, subroutines, etc., that access a particular data structure or type of data structure, through which all accesses to that structure or type, except those within the function, etc., must pass. Also called "clusters" or "Parnas modules".

LIBRARY (Benchmark Library) - A collection of synthetic programs, which have been tested and documented for general use by Government agencies in computer benchmarks.

LIBRARY (or partitioned data set) - A group of related files, logically belonging to separate projects or tasks.

MACRO - A body of text substituted directly for a statement or portion of a statement recognized to be of a proper form. Macro invocations may transmit parameters for substitution or for processing before substitution into the code body, that replaces the invocation.

MODULE - Identifiable subportions or a program composed of instructions or statements in a form acceptable to a computer prepared to achieve a certain result. They are characterized by lexical binding, identifiable, proper boundaries, named access, and named reference. The word "module" may apply to a subprogram, subroutine, routine, program, macro, or a function. A "compiled module" is a module or set of modules that are discrete and identifiable with respect to compiling, combining with other units and loading.

MONITOR - A level of access on a shared resource in a program with concurrent processes, including the means for arbitration of that resource.

MULTIPROGRAMMING - A capability of a computer system to have more than one process (program) in a state of execution at

the same time (see states of execution). It was used as a technique to increase the throughput activity. Multiprogramming does not require multiprocessing.

MULTIPROCESSING - A capability of a computer system to have more than one processors, to handle the work load.

Several multiprocessing systems have been developed, such as:

* Early systems, where the additional processors had specialized functions, e.g., I/O channels.

* Later systems, having one large CPU and several peripheral processors, performing quite sophisticated tasks, such as running a display.

* More common systems, having two or more processors, each of equal power.

* The computer networks, in which many different computers are connected often at great distances from one another.

OPERATING SYSTEM - A collection of programs within a computer system, which controls the use of available resources (processors, memory, backing store, I/O devices, files), in order to facilitate the computer system.

The following types of operating systems, with their capabilities are in use today:

* Basic Operating System (BOS)

Runs single user processes from initiation to termination. May or may not overlap I/O with execution. Provides basic I/O

support, that allows user to refer to files symbolically and to read and write them without knowing the hardware details of the I/O interface. Provides basic batch supervisor services, that control normal and abnormal job termination, job-to-job transition, and operator communication. Provides a minimum base for program development by supporting at least one language translator and/or linker/loader.

* Multiprogramming Operating System (MOS)

Provides all of the services of the BOS. Supports the concurrent execution of two or more user jobs by allowing the execution of any job to be suspended, while another is executed, without any special programming considerations in the user job. Prevents concurrently executing jobs from accidentally or intentionally destroying each other or the supervisor.

* Multiprocessor Operating Systems (MPOS)

Allows the computing load to be spread across more than one processor, based on automatic (programmed) load-leveling algorithms or operator control, but does not require special case programming in the user job. MPOS includes the shared storage, loosely coupled and networked types.

* Virtual Machine Monitor (VMM)

The operating system presents an interface to the user program, that makes it appear that the program is executing on a real computing system.

* Time-Sharing Operating System (TSOS)

This is a variant of the multiprogramming operating system, in which system resources are allocated to user jobs in

such a way, that all jobs appear to progress at the same rate. In addition, users are allowed to "interact" with, and receive output from their jobs via terminals. Such systems are optimized for response rather than throughput.

* Real-Time Operating System (RTOS)

  Allows user jobs to be executed within specified short time limits.

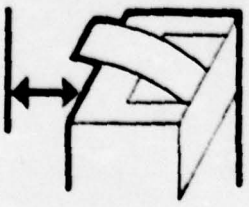
OPERATION - A well-defined finite-time execution within a program performing a time-independent function, based on its input.

PERFORMANCE MONITOR - Assists the programmer in quantifying the resource consumption characteristics of a program and in isolating performance-critical areas. Specific tools are Language Dependent Monitors and Language Independent Monitors.

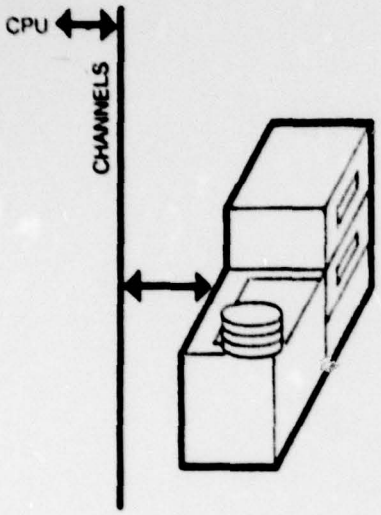
PERIPHERAL EQUIPMENT - Usually called simply "peripherals", these are external (to the CPU) devices, performing a wide variety of input, output and other tasks. "On-line" peripherals are connected electronically to the CPU. Others are "off-line" - not connected. [42]

These devices are:

Printer

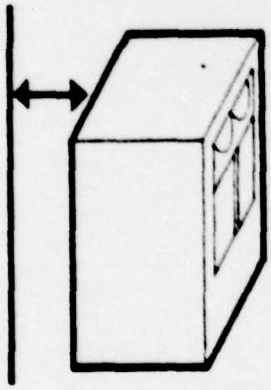


* Produces pages of printouts.
  Called "output" device.
* Very slow compared to the CPU's electronic speed.



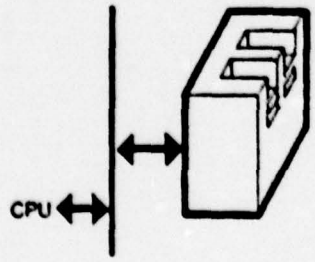
Disk Unit...Disk Drive...Disk

* Stores information by magnetic recording on continuously rotating platters.
* Handles huge amounts of storage "on-line".
* Storage is "random access", meaning the recording arms hop around fast to any "address" (location) on any "track on any disk to "read" or "write" (record) information.
* Much slower than core, but much less expensive for a given amount of information.



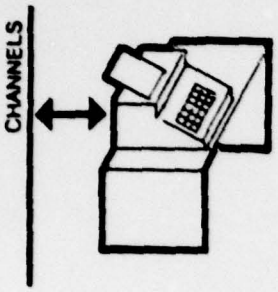
Tape Unit...Tape Drive...Tape

* Stores information on reels of magnetic tape.
* Permits plenty of storage "on-line", with huge additional amounts of reels in "library".
* Storage is "sequential", requires moving along length of tape to desired address.
* Slower than disks, less expensive too.



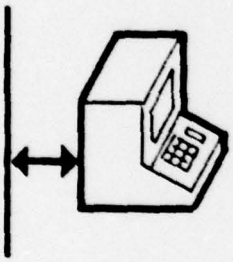
Punched Card Equipment...Unit Record Devices

* Feeds punched card information into computer (each card is a "unit record").
* Punches cards from information fed out by CPU.
* May include some sorting and collating (interleaving) ability.



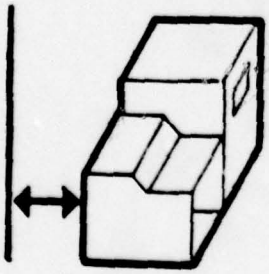
Terminal - Typewriter Type

* Keyboard like typewriter or teletype, but with extra keys and controls for communication to and from computer.
* May be "local" (near computer) or "remote" (connected to computer via cables, "data links", or telephone lines).
* Usually can perform both "input" and "output" (I/O) tasks.
* Relatively slow.



Terminal - Display Type

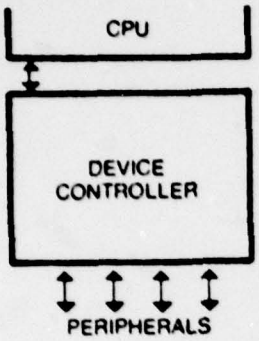
* Distinguishing feature is TV-like screen for displaying alphabetical, numerical or graphic information.
* Keyboard similar to teletypewriter plus special keys and controls.
* Can be local or remote, usually the latter.
* Relatively fast.



Intelligent Terminal

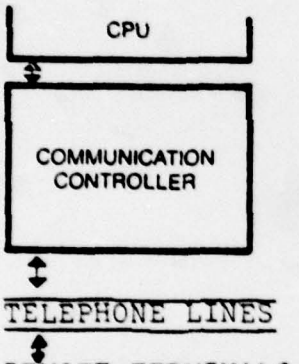
* Has some data processing ability built-in.
* Communicates with CPU as necessary to obtain information and get big jobs done.
* Generally fast devices.

Local Peripherals Device Controller



* For multiple peripherals of one type (disks, tapes, etc.) a "controller" often stands between such a group and one channel of the CPU.
* Handles information traffic between CPU and multiplicity of peripherals, although sometimes a single peripheral (e.g., a printer) has its own controller.

Communications Controller



* Handles information traffic to and from many remote terminals or computers.
* Those controllers with special processing capabilities for organizing and checking data are called "Front End Communications Processors".

Other Storage Devices and Forms:

Magnetic Drums

Store information magnetically or continuously rotating cylinder providing faster access to information, than disks do.

Floppy Disks

These small flexible disks (about the size of a 45 rpm phonograph records) are used for:

* small random access requirements in controllers and CPUs.
* as a compact substitute for punched cards.

Mass Storage

Massive amounts of storage, with slower access to it, providing lower cost for a given amount of information stored.

Other Peripheral Equipment

Computer Output Microfilm (COM)

Records information onto microfilm.

Key Data Entry Devices

The equipment used to prepare data so that the computer can accept it, including the old keypunches (card punches) plus the newer key-to-tape and key-to-disk units.

Print or Mark Readers

These data entry devices "read" digits, letters, marks, lines of text, even whole pages, for input to the computer. There are also OCR (Optical Character Recognition) devices and MICR (Magnetic Ink Character Recognition) devices.

Paper Tape Devices

Read punched paper tape, or punch it, for input and output from computer.

Plotters

Convert computer output into drawings on paper or display-type terminals. They can produce line graphs, bar charts, maps, engineering drawings, etc.

PROGRAM - A sequence of instructions. (Programs are sometimes loosely called "code"). Programs are placed into memory and interpreted (executed) by a processor. While not in use, programs are often stored in some secondary storage device, such as disk, drum, or tape.

PROGRAM MAINTENANCE - It is that process through which:

* Discrepancies are isolated and corrected (discrepancies are malfunctions within the program, or its user documentation, resulting in the failure of the program to operate as specified).

* Deficiencies are removed (deficiencies are significant program limitation. There are two sources for program deficiencies: 1) a change in other software which the given program interfaces, 2) the recognition by applications programmer groups of the need for program capabilities not yet developed). The process through which a deficiency is removed, constitutes a program "design change".

* Minor program improvements and extensions are implemented.

* Documentation is updated to reflect program changes.

* Consultation is provided to the users.

PERT/CPM SYSTEMS - Based on Program Evaluation Review Technique (PERT) and Critical Path Method (CPM), these tools assist managers in planning and controlling project activities.

PROJECT ESTIMATION SYSTEMS - Assist in the development of work breakdown structures and related performance standards for use in estimating project resource requirements.

PROTOCOL - A rule prescribing the interface discipline and correct procedures for communications with a program subroutine.

REFORMATTER - Working with a preprocessor during the introduction of structured programming construct into source program, that do not have them, assists the programmer in producing well structure programs by automatically controlling identation, the placements of comments, etc., to produce readable listings.

ROUTINE - A program or program module, that may have some general or frequent use. If a program module, then a routine always returns to the point of invocation after execution (i.e., a proper routine) or terminates abnormally.

SCOPE - The range within which an identified unit displays itself. Scope of activity refers to the boundaries within which a data structure or program element remains an integral unit. Scope of control refer to the submodules in a program that potentially may execute, if control is given to a cited module. Scope of error denotes the set of submodules, that are potentially affected by the detection of a fault in a cited module.

SIMULATOR - Software simulator is a set of programs, which, running on another host computer, simulate a computer, whose machine language is the desired high level language. Simulators are used as tools for direct support to the software development activities and can be of the following types:

* General Purpose Simulators - Allow a user to construct a computer model of a real or proposed system and to perform simulation experiments to determine the behavior of the model under various operational computer.

* Computer System Simulators - Similar in nature to the general purpose simulator except that their basic building blocks represent real computer components, whose modeled behavior approximates the throughputs, capacities and access times achievable on the modeled equipments.

SOFTWARE - It is the collection of programs and data that are used to make the hardware of a computer system run. It does encompass the following subsystems:

* Executive/Monitor or Operating System

It is the general manager of the computer system resources (memory, processors, peripheral devices, information). As a system takes up a considerable amount of core and disk storage. It performs general hardware control, input/output services, operator interfaces, job and task scheduling and a general framework in which all other programs operate.

* Program Production System

Provides environment and tools for program generation capabilities (compilers, assemblers, interpreters), program development (service functions, debug aids), program test capabilities and general communications with the hardware and/or Operating System.

* Data (File) Management System

Provides for formatted file manipulation-generation-update, retrieval, edit and report production-textual processing, man/machine interaction and overall data management.

* Utility Program System

Provides for "routine" or recurrent functions, such as Sort/Merges, Sort Generators, File Data Converters, Data Preprocessors and System Editors and Generators.

STANDARDS ENFORCER - Allows source programs to be examined automatically and checked for conformance to installation- defined standards of format, content and usage.

SUBSYSTEM - A portion of component of a system that can be designed and implemented independently of other parts of the system. This does not exclude relationships between subsystems. Subsystems may be of two general types 1) the application subsystem (e.g., material inventory) 2) support subsystems (e.g., data management, communications etc.).

SYSTEM - An integrated assemblage of components (hardware, methods, procedures, programs), designed to achieve an objective. The following are emphasized, when considering a computer system:

* the role of the component in the system rather as an individual entity.

* the synergistic effect (whole bigger than the sum of parts).

* the logical facet (content and structure of information).

* the physical facet (hardware and software).

* the fact that system maintenance is required for the life of the system.

SYSTEM DESCRIPTION LANGUAGES & ANALYZERS - Tools assisting systems analysts in describing the functional characteristics of a system and in validating the consistency and completeness of a functional decomposition.

TEST CASE DESIGN ADVISORS - Analyze programs written in a high level language and present the results of that analysis in a form suitable to assist test case designers.

TEST DATA AUDITORS - Compare data files against specification and produce reports of discrepancies and/or compliance.

TEST DATA GENERATORS - Create data files for testing and validating programs.

TEST INSTRUMENTS AND ANALYZERS - Instrument modules under test so as to collect data characterizing the behavior of the model.

TRANSLATOR - A program that takes as input a program written in one programming language (the source language) and produces as output a program in another language (the object or target language).

The following translator are used in computer systems:

* COMPILER - Translates programs written in a High Level Language (HLL), such as PL/1, FORTRAN or COBOL into either relocatable object code acceptable to a Linker or to a Low Level Language (LLL), such as assembly, acceptable to an Assembler.

* INTERPRETER - It accomplishes direct execution of a simplified language, called Intermediate Code (IC), to which

certain other translators transform a programming language (e.g., SNOO BOL is often interpreted, the IC being a language called POLISH POSTFIX NOTATION). In some cases, the source language itself can be the IC (e.g., most command language as the Job Control Language, in which one communicates directly with the Operating System, are interpreted with no translation at all).

* ASSEMBLER - Translates the Symbolic Assembly Language (SAL) to machine language, which the computer can understand, as the statements of SAL generally correspond to a single machine instruction. Specific assemblers include Basic Assemblers and Macro Assemblers.

* PREPROCESSOR - Translates a HLL (such as a "structured" version of FORTRAN) into another HLL (such as the conventional FORTRAN) and vice versa.

* CROSS-TRANSLATOR - A program running on one machine, that produces object code for another machine. The cross translator runs on a parent computer and generates code to be executed on a child computer. The parent computer typically executes an assembler or HLL compiler written in a common language like FORTRAN. The output from a parent computer is loadable object code for the child computer.

* LOADER (or Linking Loader) - Translator, whose object code is actual machine code and whose source language is almost identical, usually consisting of machine language programs in

relocatable form together with tables of data specifying points, where the relocatable code must be modified to become truly executable.

It combines the text produced by separate invocations of Compilers of Assemblers ("object modules") into executable code strings ("load modules" or "core images"), that can be loaded into the main storage of the computer system and executed without further preprocessing.

Specific loaders are Basic Linkers, Simple Overlay Linkers and Extended Overlay Linkers.

UTILITIES - Generalized software packages such as sorts, merges, data transcriptions, file maintenance, debugging aids, error handlers, checkpoints, accounting routines, dumps, labeling routines, copy routine, extracts, conversion routines, cataloging routines, programs maintenance routines, which are necessary to maintain the system and to process application date, but are insensitive to data values E.G., the transfer of data from one storage device to another.

APPENDIX A

CP-642B SYSTEM DESCRIPTION

Manufacturer	UNIVAC
Construction Standard	MIL-E-16400
Maximum Physical Dimensions	72"Hx37"Dx38"W
Maximum Weight	2,400 lbs.
Maximum Power Consumption	2,000 watts
Architecture	
Word Size	30-bits
No. of Registers	3 (accumulators)
Inst. Execution Time	
Add/Sub/Load	8-12 usec
Multiply	8-12 usec
Divide	8-12 usec
Microprogrammable	No
Technology	Discrete Components
Privileged State	No
Stack	No
Main Memory	
Maximum Size	32K to 262K-words
Speed	4 usec
Word-size	32-bit
Memory Parity Checking	No
Memory Write Protect	No
Technology	Magnetic Core
Multiported	No
Instruction Set	
Double Precision	Yes
Byte Manipulation	Yes
Bit Manipulation	No
Floating Point	Software Implemented
Math/Trig Functions	Software Implemented
Negation	One's Complement
Arithmetic Complement	One's Complement
Stack Manipulation	No
Addressing	
Direct	32K-words
Indirection	No
Indexing	7 Index Registers
Paging Hardware	No

I/O Controller	
No. of Channels	16
Types of Channels	Parallel
Maximum Data Rate	Unknown
Processor Independent	No
Maintenance/Control Panel	
Location	Front of Cabinet
Multi-register displays	Yes
Initial Program Load	Firmware
Reliability & Maintainability	
MTBF	1500 hours
MTTR	Unknown
Diagnostic Programs	Yes
Modular Building Blocks	No
Support Software	
Assemblers	Yes
Compilers	CS-1
Loader	Yes
Editor	Yes
Librarian	Yes
Debug Routines	Yes
Operating Systems	No
Real-Time OS	No
Interrupts	
Priority Structure	Yes
Nesting Capability	No

APPENDIX B

AN/UYK-7 SYSTEM DESCRIPTION

Manufacturer	UNIVAC
Construction Standard	MIL-E-16400
Maximum Physical Dimensions	41"Hx24"Dx20"W
Maximum Weight	527 to 1,139 lbs.
Maximum Power Consumption	1,720 to 6,000 watts
Architecture	
Word Size	32-bits
No. of Registers	8 or 16 (accumulators)
Inst. Execution Time	
Add/Sub/Load	6.5 usec
Multiply	10.0 usec
Divide	17.0 usec
Microprogrammable	No
Technology	Third Generation/MSI
Privileged State	Yes
Stack	No
Main Memory	
Maximum Size	256K-words (16K/module)
Speed	1.5 usec
Word-size	32-bits
Memory Parity Checking	No
Memory Write Protect	Yes
Technology	Magnetic Core
Multiported	8 ports/16K module
Instruction Set	
Double Precision	Yes
Byte Manipulation	Yes
Bit Manipulation	Yes
Floating Point	Hardware
Math/Trig Functions	Software
Negation	One's Complement
Arithmetic Complement	One's Complement
Stack Manipulation	No
Addressing	
Direct	262K-words
Indirection	Multi-level
Indexing	7 or 14 index registers
Paging Hardware	No

I/O Controller	
No. of Channels	16
Types of Channels	Serial/Parallel
Maximum Data Rate	167,000 words/sec
Processor Independent	Yes
Maintenance/Control Panel	
Location	Remote
Multi-register displays	No
Initial Program Load	Firmware
Reliability & Maintainability	
MTBF	Unknown
MTTR	15 minutes
Diagnostic Programs	Firmware/Software
Modular Building Blocks	Yes
Support Software	
Assemblers	Yes
Compilers	FORTRAN/CMS-2
Loader	Yes
Editor	Yes
Librarian	Yes
Debug Routines	Yes
Operating Systems	SHARE/7
Real-Time OS	Yes
Interrupts	
Priority Structure	Yes
Nesting Capability	No

APPENDIX C

AN/UYK-20(V) DPS DESCRIPTION

Manufacturer	UNIVAC
Construction Standard	MIL-E-16400
Maximum Physical Dimensions	18.6"Hx24.0"Dx17.5"W
Maximum Weight	185 lbs.
Maximum Power Consumption	900 watts
Architecture	
Word Size	16-bits
No. of Registers	16 or 32 (general)
Inst. Execution Time	
Add/Sub/Load	0.75 usec
Multiply	3.8 usec
Divide	6.8 usec
Microprogrammable	Yes (512 word growth)
Technology	3rd generation/MSI
Privileged State	No
Stack	No
Main Memory	
Maximum Size	65K-words
Speed	0.75 usec effective
Word-size	16-bits
Memory Parity Checking	No
Memory Write Protect	No
Technology	Magnetic Core RAM
Multiported	Two (CP/IOC and DMA)
Instruction Set	
Double Precision	Yes
Byte Manipulation	Load/Index/Store/Compare
Bit Manipulation	Limited
Floating Point	Yes
Math/Trig Functions	Yes
Negation	One's and Two's Complement
Arithmetic Complement	Two's Complement
Stack Manipulation	No
Addressing	
Direct	65K-words
Indirection	Multi-level
Indexing	All general registers
Paging Hardware	64 page address registers

APPENDIX D

BASIC AN/UYK-20 HARDWARE CONFIGURATION AND OPTIONS

Basic Configuration

* Microprogrammed Processor
* Input/Output Controller
* 16 General Registers
* Bootstrap ROM - two programs for channels and peripheral devices selected by the user
* 8K-words of Core Memory
* Power Supply as specified by the user:
  - Single phase, 115 volts, 60 or 400 hertz
  - Three phase delta, 115 volts, 60 or 400 hertz
  - Three phase wye, 208 volts, 60 or 400 hertz
* Four Input/Output Channels (one group) as specified by the user:
  - MIL-STD-188C Synchronous (0 to 9600 baud)
  - MIL-STD-188C Asynchronous (four rates of 75, 150, 300, 600, 1200, or 2400 baud)
  - RS-232C Synchronous (0 to 9600 baud)
  - RS-232C Asynchronous (four rates of 75, 150, 300, 600, 1200, or 2400 baud)
  - NTDS slow, fast, and ANEW in a normal or intercomputer mode

Options Available

* 8K-word Memory Modules (up to 65K-words)

APPENDIX E

CHARACTERISTICS OF HIS 6060 & 6080 PROCESSOR MODELS

	MODEL 6060	MODEL 6080
SYSTEM CONFIGURATION		
Number of Central Processors	1 to 4	1 to 4
Number of I/O Multiplexers	1 to 4	1 to 4
Number of System Controllers	1 or 2	1 or 2
MAIN STORAGE		
Minimum capacity (36 bit words)	98,304	131,172
Maximum capacity (36 bit words)	524,288	1,048,576
Increment Size (36 bit words)	Varies	Varies
Cycle Time (microseconds)	1.2	0.5
Words fetched per cycle unit	2	2
Storage Interleaving	2/4 Way	2/4 Way
CENTRAL PROCESSOR		
Extended Business Instruction Set	Standard	Standard
Instruction Overlap	Standard	Standard
Typical Speed (Instruction per sec)		
Single Processor	500,000	1,000,000
Dual Processor	975,000	1,950,000
Input/Output Control (Channels per I/O multiplexer)	8 to 24	8 to 24
Maximum Data Rate per I/O multiplexer (characters per sec)	3,700,000	6,000,000

APPENDIX F

DEC PDP-11/45 SYSTEM DESCRIPTION

Manufacturer	Digital Equipment Corporation
Construction Standard	Commercial
Maximum Physical Dimensions	71.5"Hx30.0"Dx21.7"W
Maximum Weight	300 lbs.
Maximum Power Consumption	2,300 watts
Architecture	
Word Size	16-bit (18-bit addresses)
No. of Registers	16 (general)
Inst. Execution Time	
Add/Sub/Load	0.3 usec
Multiply	3.3 usec
Divide	7.5 usec
Microprogrammable	No
Technology	MIS/LSI
Privileged State	Two - Kernel & Supervisor
Stack	Yes
Main Memory	
Maximum Size	128K-words
Speed	0.3 to 0.98 usec
Word-size	16-bit (18-bit for parity)
Memory Parity Checking	Yes
Memory Write Protect	Yes
Technology	Core/MOS/Bipolar
Multiported	Single - UNIBUS structure
Instruction Set	
Double Precision	Yes
Byte Manipulation	Yes
Bit Manipulation	Yes
Floating Point	Yes
Math/Trig Functions	Software
Negation	One's or Two's Complement
Arithmetic Complement	Two's Complement
Stack Manipulation	Yes
Addressing	
Direct	256K-bytes
Indirection	Yes
Indexing	All general registers
Paging Hardware	Yes

I/O Controller	
No. of Channels	14
Types of Channels	Serial/Parallel
Maximum Data Rage	1,000K-words/sec
Processor Independent	DMA only
Maintenance/Control Panel	
Location	Front of chassis
Multi-register displays	No
Initial Program Load	Firmware
Reliability & Maintainability	
MTBF	Unknown
MTTR	Unknown
Diagnostic Programs	Software
Modular Building Blocks	Yes
Support Software	
Assemblers	Yes
Compilers	FORTRAN/ALGOL/BASIC
Loader	Yes
Editor	Yes
Librarian	Yes
Debug Routines	Yes
Operating Systems	Yes
Real-Time OS	Yes
Interrupts	
Priority Structure	None
Nesting Capability	None

APPENDIX G

MILITARY COMPUTER FAMILY (MCF) BENCHMARK TESTS

Message Buffer and Transmission

Autocorrelation

Hash Table Search

Linked List Insertion

TTY Input Driver

Presort

Boolean Matrix Transpose

Target Tracking

Digital Communications Processes

Character Search

Record Unpacking

Array Manipulation

Multiple Priority Interrupt

Vector-to-Scan Line Conversion

Scale Vector Display

Virtual Memory Exchange

BIBLIOGRAPHY

1. Jacobowitz, H., and advisory editor Basford, L., Electronic Computers Made Simple, p. 313, Allen, 1967.

2. Evans, R., "The Microprocessor Revolution in Military Systems," Military Electronics, v. 2, p. 14, February 1967.

3. Hermann, R. J., "Communications, Command and Control: Changing the Face of Force Management," Defense Management Journal, v. 14, November 1978.

4. Jenista, J. F., Jr., Captain USN, "Navy Command and Control," Computers in the Navy, edited by Procop, J., Captain, SC, USN, p. 127-157, Naval Institute Press, Annapolis, Maryland, 1976.

5. Richardson, Joyce, R., LCDR, USN, History of the AN/UYK-20 (V) Data Processing System Acquisition and Its Impact on Tactical Systems Development, Master of Science Thesis, Naval Postgraduate School, Monterey, California 1976.

6. Garmon, G. S., LCDR, USN, "Restructuring the Navy Tactical Data System" Proceedings of 2nd Computer Software Applications Conference 1978, IEEE Catalog no. 78CH1338-3C, p. 504-507.

7. Missile System Subgroup, Weapons Systems and Engineering Directorate, Naval Sea Systems Command, "DDG Upgrade," Surface Warfare, January-February 1977, v.2.

8. RCA AEGIS Anniversary Issue Newsletter, Month 36, v. 2, RCA Moorestown, N. J., December 1972.

9. Kulesz, J. J., CDR, USN, "The Aegis Eye," Surface Warfare, January-February 1977, v. 2, No. 1.

10. Hessman, J.D., and Battiny, E.D., "Fiscal Year 1979 Budget Report: Good and Bad News for the Navy," Sea Power, February 1978.

11. Musgrave, A. W., Jr., CDR (sel) USN, "SIRCS and OMB Circular A-109: Changing the Major System Acquisition Process," Defense Management Journal, p. 24-28, May 1978.

12. Farinella, F. A., "A Logical Follow-On: SIRCS, Shipboard Intermediate Range Combat System," Surface Warfare, January-February 1977.

13. Hull, B. W., LCDR, USN, "Joint Tactical Information Distribution System," Surface Warfare, v. 2, No. 1, January-February 1977.

14. Bobrow, D. B., "Communications, Command and Control: The Nerves of Intervention," The limits of Military Intervention, edited by Stern, P., SAGE Publications, p. 101-117, London/Beverly Hills, 1977.

15. Gallotta, A. Jr., Captain, USN, "C^3 and EW in the Navy," Military Electronics, April 1978.

16. Lamporte, R. A., LCDR, USN, "The United States Navy and the World Wide Military Command and Control System," edited by Roland, J. R., Lt.Col. USAF, Naval Post-graduate School, November 1978.

17. Honeywell Series 6000, Summary Description, 1972.

18. Martin, C., and O'Bleness, R. O., Jr., "Life Cycle Management Concepts for Air Force Computer Resources," Proceedings of 2nd Computer Software Applications Conference 1978, IEEE Catalog No. 78CH1338-3C, p. 764-768.

19. Fisher, D. A., "DoD's Common Programming Language Effort," Computer, v. 11, No. 3, p. 24-33, March 1978.

20. Campbell, J. S., "Harnessing the Software Revolution to Meet Navy Needs," Defense Management, v. 14, No. 3, p. 16-23, May 1978.

21. Burr, W. E., and Coleman, A. A., "Overview of the Military Computer Family Architecture," AFIPS - Conference Proceedings, v. 46, p. 131-137.

22. Naval Fire Control and Action Data Automation Systems, Janes Weapons Systems, 1976.

23. NATO Facts and Figures, NATO Information Service, p. 157, 348, 143-144, January 1976.

24. Perry, W. J., "NATO, Two-Way Street Called Essential," Aviation Week & Space Technology, p. 49-51, 12 February 1979.

25. Couger, D. J., and Knapp, R., Systems Analysis Techniques, Wiley & Sons, Inc., 1974.

26. Hice, G. F., Turner, W. S., and Cashwell, L. F., System Development Methodology, North Holland Publishing Company, 1974.

27. Bogdan, W. R., "Life Cycle of Navy Airborne Antisubmarine Warfare Tactical Software," Proceedings of 2nd Computer Software Applications Conference 1978, IEEE Catalog No. 78CH1338-3C, p. 499-503.

28. Wagner, E., Lieblein, E., Rodriguez, J., and Stone, H., "Evaluation of the Software Bases of the Candidate Architectures for the Military Computer Family." AFIPS-Conference Proceedings, v.46, p. 175-183.

29. Naval Research Laboratory Report 8247, Computers/ Processors, by Lemley, L. W., 15 August 1978.

30. Cornyn, J. J., Smith, W. R., Coleman, A. H., and Svirsky, W. R., "Life Cycle Cost Models for Comparing Computer Family Architectures," AFIPS-Conference Proceedings, v. 46, p. 185-199.

31. Korn, G. A., Microprocessors and Small Digital Computers for Engineers and Scientists, McGraw Hill.

32. Boehm, B. W., "Software Engineering," IEEE Transactions on Computers, v. C-25, p. 1226-1241, December 1976.

33. Brooks, F. P., The Mythical Man-Month, Addison-Wesley, 1975.

34. TRW Systems Engineering and Integration Division Report SS-72-01, The Cost of Developing Large Scale Software, by Wolverton, R. W., March 1972.

35. Putnam, L. H., "Example of an Early Sizing, Cost and Schedule Estimate for an Application Software System," Proceedings of 2nd Computer Software Applications Conference 1978, IEEE Catalog No. 78CH1338-3C, p. 827-832.

36. Tausworthe, R. C., Standardize Development of Computer Software, Part II, Standards, Jet Propulsion Laboratory, California Institute of Technology, August 1978.

37. Pratt, T. W., Programming Languages: Design and Implementation, Prentice-Hall, Inc., 1975.

38. Joslin, E. O., Software for Computer Systems, College Readings Inc., 1970.

39. Lecht, C. P., The Management of Computer Programming.

40. Wiederhold, G., Data Base Design, McGraw-Hill, 1977.

41. Murdick, R. G., and Ross, J. E., Information Systems for Modern Management, Prentice-Hall, 1971.

42. Computer Lessors Association, Inc., The Intelligent Guide to Computer Selection, 1975.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
4.	Professor Uno Kodres, Code 52 Ko Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
5.	Hellenic Navy General Staff Stratopedon Papagou Holargos, Athens, Greece	3
6.	CDR Vassilios A. Naoum 337 Patission Street Athens (905), Greece	2
7.	LCDR Marcos G. Mastrakas Naval Postgraduate School SMC Box 2552 Monterey, California 93940	1