NASA CR-132404

FEASIBILITY STUDY OF AN INTEGRATED PROGRAM FOR AEROSPACE-VEHICLE DESIGN (IPAD) SYSTEM

by C. A. Garrocq, M. J. Hurley et al

VOLUME IV

DESIGN OF THE IPAD SYSTEM

PART I - IPAD SYSTEM DESIGN REQUIREMENTS

(PHASE I, TASK 2)

20 August 1973

Publicly Released
February 10, 1978

Prepared Under Contract No. NAS 1-11431 by

GENERAL DYNAMICS/CONVAIR AEROSPACE DIVISION
San Diego, California

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

(NASA-CR-132404) FEASIBILITY STUDY OF AN INTEGRATED PROGRAM FOR AEROSPACE-VEHICLE DESIGN (IPAD) SYSTEM. VOLUME 4: DESIGN OF THE IPAD SYSTEM. PART 1: IPAD SYSTEM DESIGN (General Dynamics/Convair) 250 p HC G3/02

N78-16007

A11/MF A01

Unclas
02561

FEASIBILITY STUDY OF AN
INTEGRATED PROGRAM FOR AEROSPACE-VEHICLE
DESIGN (IPAD) SYSTEM

VOLUME I	- SUMMARY
VOLUME II	- CHARACTERIZATION OF THE IPAD SYSTEM (PHASE I, TASK 1)
VOLUME III	- ENGINEERING CREATIVE/EVALUATION PROCESSES (PHASE I, TASK 1)
VOLUME IV	- DESIGN OF THE IPAD SYSTEM PART I - IPAD SYSTEM DESIGN REQUIREMENTS (PHASE I, TASK 2)
VOLUME V	- DESIGN OF THE IPAD SYSTEM PART II - SYSTEM DESIGN PART III - GENERAL PURPOSE UTILITIES (PHASE I, TASK 2)
VOLUME VI	- IMPLEMENTATION SCHEDULE DEVELOPMENT COSTS OPERATIONAL COSTS BENEFIT ASSESSMENT IMPACT ON COMPANY ORGANIZATION SPIN-OFF ASSESSMENT (PHASE II, TASKS 3 to 8)

FOREWORD

This investigation was conducted for the NASA Langley Research Center by the Convair Aerospace Division of General Dynamics Corporation under Contract NAS 1-11431.

The NASA Technical Monitor was Dr. R. E. Fulton, Head, IPAD Technology Section, Design Technology Branch, Structures and Dynamics Division, assisted by Dr. J. Sobieszczanski and Mrs. S. J. Voigt. The Convair Project Leader was Mr. C. A. Garrocq. This volume was prepared by M. J. Hurley, K. C. Bonine, J. C. Mallory and H. W. Buckner, assisted by D. Kurpiewski, J. Maughmer, R. A. Westerwick, J. E. Meyer, G. M. Oing and T. M. Wooster.

The Control Data Corporation participated in the performance of this study as a subcontractor to the General Dynamics Corporation. Also indirectly assisting were personnel from UNIVAC Division of Sperry Rand and International Business Machines.

The period of performance was from 15 March 1972 to 30 August 1973.

TABLE OF CONTENTS

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS, Contd

TABLE OF CONTENTS, Contd

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS, Contd

LIST OF ILLUSTRATIONS, Contd

LIST OF TABLES

LIST OF TABLES, Contd

PRECEDING PAGE BLANK NOT FILMED

SUMMARY

An IPAD system is defined herein as consisting of four major components, as shown in Figure S-1: (1) A Management Engineering Capability represented by a battery of automated Operational Modules for various management/design/engineering disciplines, (2) an IPAD Framework Software which supports and augments the Engineering Capability, (3) an Operating System Software, which features a comprehensive Data Base Management System, and (4) a Computer Complex Hardware, on which all the Engineering, IPAD, and System software will be mounted and exercised. From this statement, it can be inferred that the Management/Engineering Capability can and should be tailored to the specific needs of the management/design/engineering team (i.e., the battery of Operational Modules for aircraft design would be different than that for missiles, or navy vessels, or terrestrial vehicles, or civil engineering projects, although many common elements could be identified). On the other hand, the IPAD Framework Software, the Operating System Software, and the Computer Complex Hardware could have essentially the same basic capabilities for all users, with freedom of choice in specific software, and type and quality of equipment desired within each computer complex.

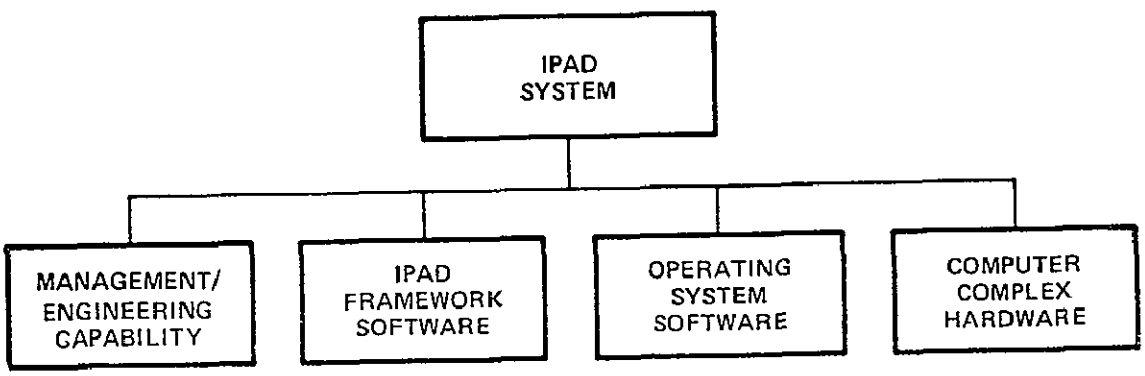


Figure S-1. Major IPAD System Components

The organization, engineering usage philosophy, and the accompanying IPAD design concept developed in this study provide the flexibility required to satisfy the project needs of any management/design/engineering team which will use and exploit the IPAD system's capability in any way it sees fit.

Objectives of this Volume

The major objective of this volume is to identify the system requirements, software elements, and hardware equipment required for an IPAD system. This objective was pursued by evolving an IPAD conceptual design, conducting a potential user survey, projecting work loads for various types of interactive terminals, comparing various features of major host computing systems, and finally selecting target systems and identifying the various elements of software required for IPAD.

IPAD in Relation to the Host Computing System

Several options were evaluated on how to incorporate IPAD into the host computer complex. System and subsystem levels of dependency were compared, resulting in the recommendation of a subsystem-level approach as shown in Figure S-2. This figure presents an overview of the host operating system, with IPAD designed subordinate to its operating subsystems; i.e., an IPAD job is executed like a standard job operating within the framework of the host computing system. This approach requires that the host operating system be highly capable, since IPAD has divested itself of all host system functions. Subordinating IPAD to the interactive communications subsystem means that there will be variations in IPAD system operation between installations of different computer manufacturers.

Figure S-2 further illustrates the possible use of a dedicated minicomputer for users of certain response-time critical code used with refreshed CRT terminals. The principal advantage accrues directly to those users operating through the minicomputer; viz., faster response time, since many interactive functions will be local to the dedicated minicomputer (perhaps shared by several interactive terminals) and hence accomplished without resorting to the host computer. Since these functions will no longer require the host computer, the host operating system will be able to service the other users more efficiently. Further, the IPAD system software can be split between that residing on the host system and that residing on the minicomputer. This will result in the least impact on the host operating system since it is only the IPAD software on the minicomputer which requires the interactive communications subsystem with very high data transfer rates (e.g., 40,800 baud). This split will enable a clean interface between the host and minicomputer IPAD system software. The disadvantages are that it requires additional hardware to utilize the highly capable refreshed CRT terminals and additional software development to provide the IPAD systems for the dedicated minicomputer. The problems of addressing several target minicomputing systems, although not as severe, parallel those for host computing systems.

However, not every institution intending to use IPAD may be willing to provide the dedicated minicomputers with their attendant identifiable costs. An alternative approach is to provide an IPAD system which optionally utilizes minicomputers for the refreshed CRTs; this necessitates that the IPAD system on the host be able to (optionally) service the refreshed CRTs thus effectively eliminating some of the advantages discussed above, which would now accrue only to those users at those institutions employing the optional minicomputers.

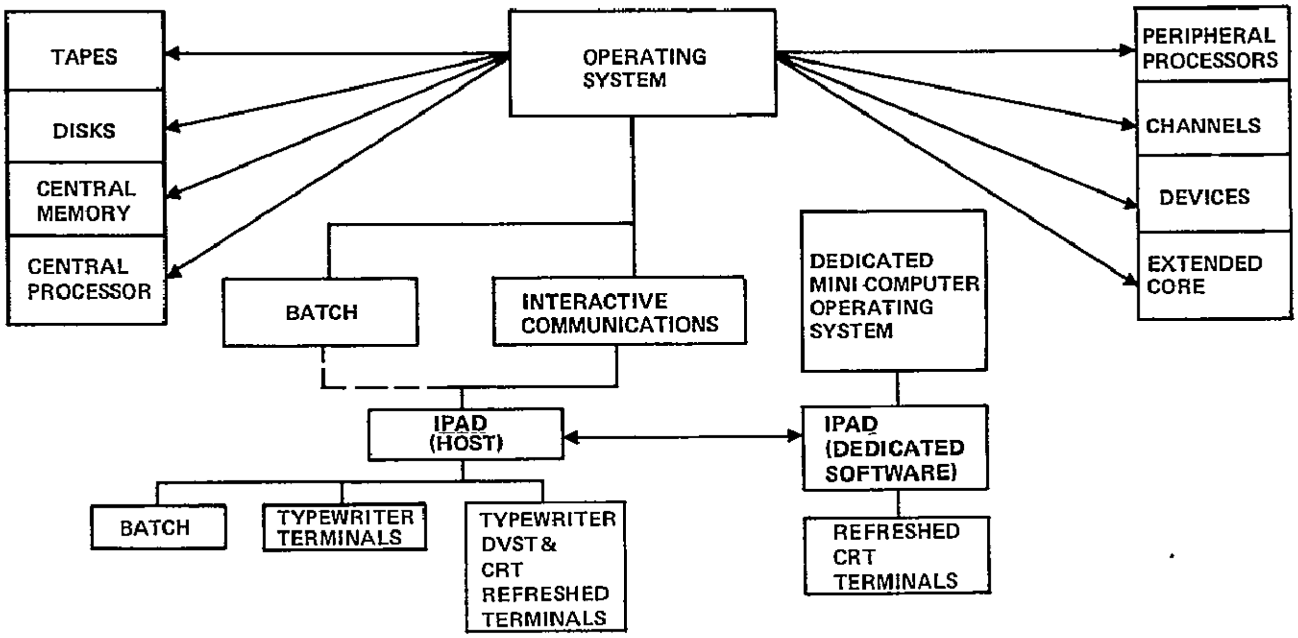


Figure S-2. IPAD in Relation to Host Operating System

The main user and computing system features that lead to the adoption of the design approach shown in Figure S-2 are:

1. Least competition for resources, since IPAD looks like a standard job to either the batch or interactive-communications subsystems.

2. Minimal software development, since existing system software is being fully exploited.

3. Least hardware/software dependence, since the bulk of IPAD is interfaced (buffered) through the host operating system.

4. Least impact on the host operating system - IPAD looks like a standard job.

5. Potentially longest life, since - being a "standard" job - IPAD will continue to be supported far into the future, possibly until standard host operating systems themselves offer all the advantages accrued through IPAD.

6. Continuous upgrading of IPAD through obtaining (practically gratis) the host system's latest features, including those of all of its subsystems.

7. Fast response time for users of the response-time critical code.

8. Cleanest interface between IPAD software elements for host and minicomputer systems.

IPAD Computer Software

The various elements of computer software associated with IPAD are illustrated in Figure S-3.

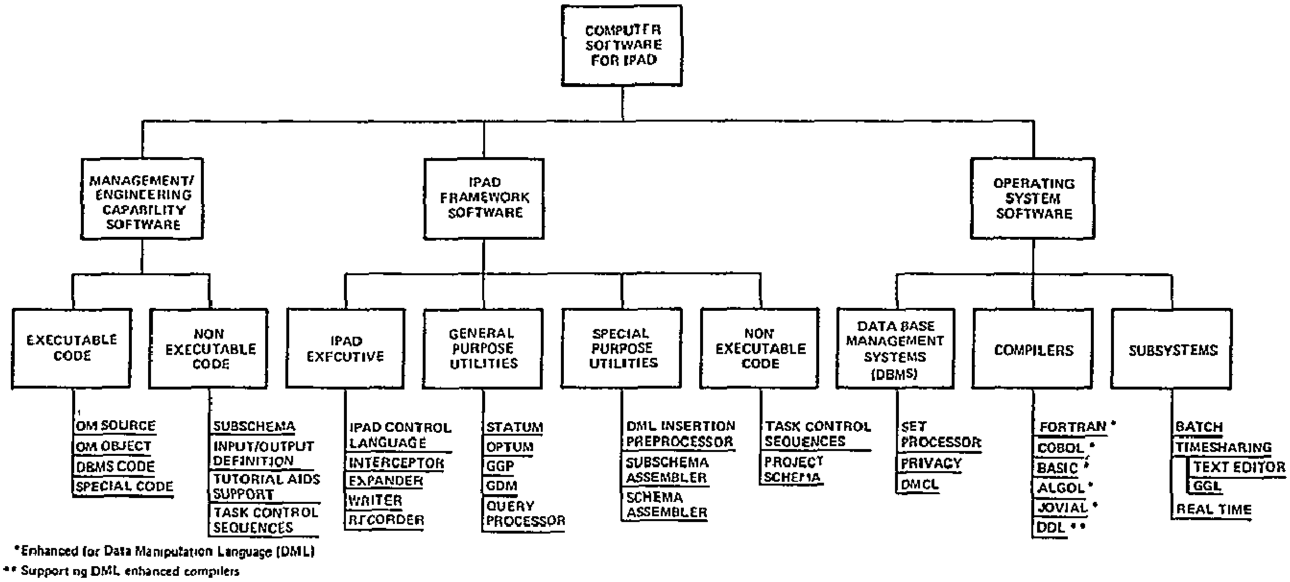


Figure S-3. Computer Software Associated with IPAD

The three major classes of software are:

1. Management/Engineering Capability Software. - The total automated capability of the management/engineering/science community is resident in a library of automated operational modules consisting of both a public domain library, accessible to all parties, and private libraries containing modules with limited or restricted availability, due to the nature of its contents being private data, classified information, or the like. From the total gamut of available modules a project team will select those which are applicable to their specific project to assemble a project library of automated operational modules that will be installed on the IPAD Computer Complex. The contents of this library are dynamic in the sense that programs are added or removed from it as the need arises, and are resident on disk or tape depending on their usage rate. All project related activities such as management, marketing, economics, technical disciplines, and design/drafting will have their respective automated capabilities installed in the system. The position of this software in relation to other computer software required for IPAD is shown in the first two columns of Figure S-3.

2. The IPAD Framework Software. - From the user's point of view, IPAD is a framework which supports and augments the capabilities of his computerized management, design/drafting, and analytical tools. From this viewpoint, the framework is composed of a number of utilities and interfacing capabilities, as shown in Figure S-3. The elements of this software are:

a. The IPAD EXECutive function, which provides control of the full capability of the host operating system/timesharing subsystem and is interfaced by: tutorial aids and the ability to code, save, and execute pre-established task sequences.

b. The General Purpose Utilities, which include:

- The Query Processor, which provides interface with a project-oriented Multidisplinary Data Bank and the Data Base Management System. To the user, the Data Base and Query Processor provide for accurate and efficient communication with respect to task assignments and task status, and efficient access to pertinent design data, design tools, and operating modules.
- A statistical utility (STATUM) and a general-purpose optimization utility (OPTUM), which provide general enegineering capability in these areas.
- A General Graphics Plotter (GGP), and a General Drafting Module (GDM), which provide multipurpose plotting and design/drafting capabilities, with access to hardcopying equipment.

The foregoing three major groups of general purpose utilities make up the basic capability. Additional utilities could be added in the future, or, conversely, some elements of this capability could be absorbed by the operating system.

c. The Special Purpose Utilities, which provide a capability to incorporate Operational Modules and to assist the user in preparing existing modules for operation within the IPAD Framework.

d. Non-excutable Code, which provides a task integration capability and permits the construction of a task oriented user file appendage to the data base, and the construction of a Data Base Management System interface to share data among the Operational Modules.

3. The Operating System Software. - This software usually resides in disk and consists of: system utilities to support the user, such as compilers, assemblers, translators, file managers, etc., the operating system library, containing system-support entities such as the resource allocator, the job scheduler, the record manager, the loader, etc.

Features of the operating system software which are considered important to IPAD include: random access files, which are deemed to be required by current and projected mass storage hardware for fast access/retrieval times; index sequential files, which combine both random and sequential features; permanent files, required for continuous availability of information contained in IPAD's data banks; an UPDATE utility, to selectively update while retaining prior data; an interactive communication subsystem, including time and memory sharing features to provide fast response times;

and an interactive graphics subsystem, to provide capability for making graphs, drawings, pictures, etc. In relation to the latter feature, it is important to point out a pressing need within IPAD for a standard graphics language.

IPAD is designed to fully exploit the host computer's operating system software. In particular, the operating system must be upgraded to contain a capable time sharing subsystem and a comprehensive Data Base Management System.

Host Computer Hardware for IPAD

Figure S-4 summarizes the minimum hardware requirements for a single project within IPAD and major computing candidates that have suitable operational equipment. These minimum requirements were derived through an evaluation of the computing requirements necessary to support typical present day computing, including interactive computing. In addition to these derived requirements, a minimal complement of magnetic tape units, card reader/punches, line printers, and recorders has been presumed. It is noted that in an IPAD environment, most data is envisioned to originate and remain within the system on mass storage devices. Data viewing, job submittal, etc. is to be accomplished via interactive devices so that the requirements for conventional job entry and I/O are minimal.

MINIMUM REQUIREMENTS

• MAIN FRAME HARDWARE	
MAJOR MEMORY CYCLE	⩽ 1 0μS
TYPICAL BINARY FLOATING ADD	⩽ 1.5μS
CENTRAL MEMORY SIZE*	⩾ 100,000 SINGLE PRECISION "WORDS"
JOB ROLLIN/ROLLOUT OR SWAPIN/SWAPOUT	"PAGING" HIGH-SPEED TRANSFER TO EXTERNAL (LOW-SPEED) CORE
• PERIPHERAL HARDWARE	
MASS STORAGE CAPACITY	> 150M SINGLE PRECISION "WORDS"
MASS STORAGE TRANSFER RATE	> 1M CHARACTERS PER SEC.
MAGNETIC TAPE UNITS	> 3
CARD READER/PUNCH	1
HIGH-SPEED PRINTERS	1
MICROFILM RECORDER	1 (CAN BE REMOTE)
TERMINALS (WITH HARDCOPIERS)	26 DVST & 10 REFRESHED CRTs
PAPER TAPE READER/PUNCH, FLAT-BED PLOTTERS	AS REQUIRED

CANDIDATES (ALL ARE LARGE-SCALE SCIENTIFIC COMPUTERS)

IBM	CDC	UNIVAC	HONEYWELL	BURROUGHS
370/145	CYBER 70 SERIES	1108	6000/6030/6040	B6500
370/155,158	EXCEPT MODEL 76	1110	6000/6050/6060	B6700
370/165,168	(VIZ, CDC 6000 SERIES)		6000/6070/6080	B7700

*IPAD WILL INCREASE CENTRAL MEMORY RESIDENCY

Figure S-4. Host Computer Hardware for IPAD, Single Project

It must be recognized that the requirements for central memory is in addition to the requirements for the host system support software. Typically, efficient host system software supporting the IPAD user is written as re-entrant code, supporting many users and residing in central memory. The data management system support to IPAD may significantly increase central memory residency. It is suspected that the 100,000 word requirement to support IPAD OMs may expand to the order of 130,000 for total system support.

The hardware candidate computing systems (bottom of Figure S-4) were obtained by liberally applying the minimum "requirements" listed in the figure; e.g., to be a candidate, the computing systems central memory had to be expandable to an excess of 100,000 "words" or the equivalent 1.0 megabytes. The resulting candidates are generally the top of the line systems of each manufacturer which are in current production. As can be seen, a reasonable number of candidates are available that can - at least minimally - support IPAD.

In consideration of the above, the three target systems selected as a basis for the subsequent design feasibility study (Volume V) were:

1. IBM 370/145, 155, 158, 165, 168 with the VM/370 operating system.

2. CDC 6000 Series with the SCOPE 3.4 operating system.

3. UNIVAC 1108, 1110 with the EXEC 8 operating system.

Interactive Terminals for IPAD

Figure S-5 summarizes the characteristics of typical direct-view storage tube (DVST) and refreshed cathode ray tube (CRT) terminals suitable for IPAD. These terminals are of the most capable type presently available in the market.

The terminal hours by device type and design phase were estimated and used to select an appropriate number of interactive terminal types as envisioned for a single project within IPAD (shown on Figure S-4). The terminal configuration is sized for one project as was the minimum required computer hardware. A company is usually engaged in a number of projects and technical activities in a more or less parallel fashion in time. Since activity in any analytical process varies with time in a project (i.e., builds up to a peak activity and tails off to a lower level), only the incremental increase in the hardware configuration is actually needed to accommodate other projects rather than a full replication of this configuration for each project. The representation of multiple projects - although not difficult to accomplish - would have necessitated statistical inference and additional computer usage (e.g., Monte Carlo) and was considered to be outside the scope of this study.

MANUFACTURER	CDC	CDC	IBM	VECTOR GENERAL	IMLAC	TEKTRONIX
MODEL	274	GPGT	2250	3D2	PDS-1	4002A
TYPE OF CRT	REFRESHED	REFRESHED	REFRESHED	REFRESHED	REFRESHED/DVST*	DVST
SCREEN SIZE (IN)	20	20	12 x 12	13 x 14	7 5 x 8 5	7 5 x 5 5
SHAPE	CIRCULAR	CIRCULAR	SQUARE	RECT	RECT	RECT
RASTER X RASTER	4,096 x 4,096	4,096 x 4,096	1,024 x 1,024	4,096 x 4,096	1,024 x 1,024	1,024 x 760
INTERACTIVE TOOLS						
A/N KEYBOARD	X	X	X	X	X	X
LIGHT PEN	X	X	X	X	X	
JOY STICK, MOUSE, ETC				X	X	X
ANALOG TABLET				X	X	X
FUNCTION KEYBOARD	X	X	X	X	X	X
MINICOMPUTER	CDC 1700	CDC SC1700	NONE	PDP-11, VARIAN 620, ETC	BUILT IN	NONE

* DVST = DIRECT VIEW STORAGE TUBE

Figure S-5. Interactive Terminals Suitable for IPAD

Answers to Key Questions

Key questions posed in the RFP in relation to Task 2 are answered in the followin paragraphs:

How should the system be organized to provide sufficient flexibility to accommodate independently developed codes, pre-existing and/or those created in the future? - Three organizational systems were evaluated: hardwired, self-organized, and user-organized. The first two are relatively inflexible to change and growth. The user-organized system is a compromise between the first two approaches, which obtains the sophistication of the self-organized system through user interaction yet has the inherent simplicity of the hardwired system. The advantages combine those of the two preceding systems, and in addition the approach is the most flexible, is highly adaptable to changing conditions, and is the most easily modified/updated. Since the individual users are responsible only for their own Operational Modules, it features the fastest incorporation of these modules by a substantial margin and the user remains an involved participant in the process. Perhaps surprising, this approach requires the least overall executive software development because the user himself performs many of the executive functions. The use of Task Control Sequences will permit the user to fabricate "execution strings" nearly as automatic as possible within a self-organized system.

What computer languages will be admissible in the pre-existing codes? - The intent of the user-organized system is to accommodate all codes that will currently execute on the host computer system. Five computer languages are considered as candidates acceptable in IPAD: FORTRAN, ALGOL, JOVIAL, PL/1 and COBOL. In addition, the various assembly languages supported by the host computer will be acceptable.

What degree of machine dependency is acceptable for IPAD? - The machine dependency of IPAD is that of its software elements, some of which are highly dependent on the operating system, while others are almost machine independent. Total machine dependency must be accepted for software such as: the host operating system, the host timesharing subsystem, compilers for programming languages (principally FORTRAN) and for the Data Manipulation Language, a Data Base Management System, and interactive Query Processor. From the executable code for IPAD the EXECutive is very dependent on the machine, while the Special Purpose Utilities are highly transferable, and the General Purpose Utilities will be transferable if written in the proposed General Graphics Language. Transferability of non-executable code can be achieved by the development and implementation of standard languages, as proposed in this study.

To what extent and how should the human element be retained in the system control in order to utilize engineering intuition, judgement, and experience? - The user-organized approach adopted for IPAD ensures user control in the application of IPAD to the design process. The interactive terminals provide for proper interface between the user and the IPAD system mounted on the computer complex. The user performs the creative jobs with the assistance of hardware and software elements provided by IPAD.

What degree of flexibility should be given to the system operator in arranging available Operational Modules into different sequences according to the needs? - In order to become a practical, useful tool, IPAD must provide unlimited flexibility for the user to solve his design/evaluation problems. This capability is provided in the proposed IPAD design by means of Task Control Sequences that the user assembles himself (and freezes for future use).

What I/O devices will best serve IPAD? - Although IPAD can be used in batch mode, it is only under an interactive environment that it can be justified and proven to be cost effective. The most advanced type of interactive terminals are considered to best serve the potentials of IPAD.

What will be the impact of next generation computers on IPAD and its applications? - It is undoubtedly not justifiable to impose on IPAD's development the requirement to accommodate the supercomputers rather than vice-versa. The impact of next-generation computers can be lessened by adopting the following recommendations:

1. Don't explicitly provide for the supercomputers in IPAD's design approach. Let "upward compatibility" of the supercomputer's system software eventually provide the framework for IPAD.

2. Until then, "front-end" the supercomputer with a more sophisticated maxi-computer that can delegate candidate tasks. Design IPAD to reside on the maxi computer.

1 INTRODUCTION

The design of a new aerospace vehicle is presently a complex, long-term process. At the onset, a set of objectives is identified in the areas of mission, weight, performance, payload, range, etc., which are specified with a fairly good knowledge of the available design technology and constraints. The designer's goal is to minimize cost, while meeting basic project objectives. The designer possesses a fund of accumulated experience and knowledge which he applies, with intuition, to the requirements and constraints he has been given. The knowledge and experience of the designer are more and more frequently being delegated to the computer; the intuition and imagination can never be. Some of the purposes of the IPAD feasibility study were to determine what sections of the design process are amenable to automation; how much monitoring must the automation have; how can the design process be effectively organized; and, most important, how can the management/designer/engineer team members retain the visibility and control necessary to exercise their intuition and imagination in the design process.

The introduction of automation is a significant change in the design process; however, the important management aspects of this change are not only related to the technical details of engineering disciplines, programming, data bases, etc., but the key to success also depends upon managing the adaptation required of the people involved in the use of the automated process.

Automation of any process requires not only a thorough knowledge of the process, but of the pivotal factors that drive and control it. When the process involves the myriad details of project team data flow and communications, many programs and subroutines, thousands of variables, and the ramifications of computer operating system characteristics, it is easy to lose sight of the fact that it is still the designer - the engineer - who is the key driver and decision-maker in the process.

Although the various volumes of this report describe some of the considerations necessary for the technical basis needed to successfully automate the design process, the underlying, guiding philosophy has been that of providing a tool adapted to the needs of the members of a management/designer/engineer team--the ultimate users--and that is a truly useful tool. The acknowledged principle has been that the engineer and his management are generally more interested in solving the design problem than in becoming a better communicator with the computer.

The scope of the total IPAD feasibility study is illustrated in Figure 1-1. The study was divided into the following eight tasks within two study phases:

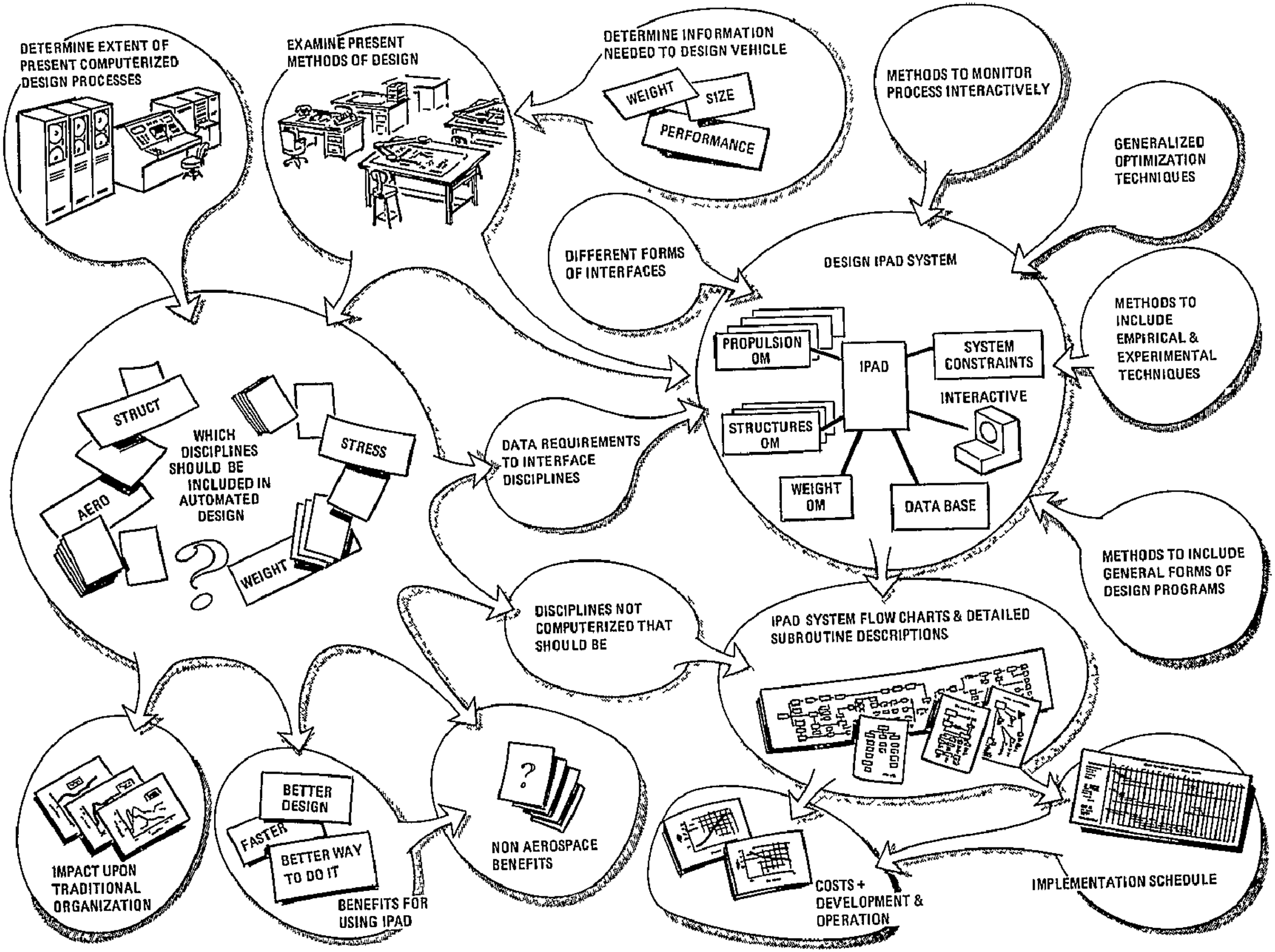


Figure 1-1. IPAD Study Flow Chart

PHASE I

STUDY PLAN COORDINATION

TASK 1 - CHARACTERIZATION OF IPAD SYSTEM

- Define an IPAD Engineering Usage Philosophy
- Identify Feasible Automated Design Procedures
- Evaluate Adequacy of Existing Computer Programs
- Recommend Areas for Further Development
- Determine IPAD Feasibility and Applicability
- Recommend IPAD's First Release Engineering Capability

TASK 2 - DESIGN OF IPAD SYSTEM

- Define a Systems Operating Philosophy
- Evaluate System Design Options
- Identify Elements of IPAD's Utility Library
- Investigate Organization and Management of Data Bank
- Determine Number and Type of Input/Output Terminals
- Determine Host Computer Complex Configurations Adequate for IPAD
- Recommend IPAD's First Release Computer System Capability

PHASE II

TASK 3 - IPAD IMPLEMENTATION SCHEDULE

TASK 4 - IPAD SYSTEM DEVELOPMENT COST

TASK 5 - IPAD SYSTEM OPERATIONAL COST

TASK 6 - IPAD SYSTEM BENEFIT ASSESSMENT

TASK 7 - IPAD IMPACT ON COMPANY ORGANIZATION

TASK 8 - IPAD SPIN-OFF ASSESSMENT

Figure 1-2 summarizes the main features of an IPAD system as presently conceived and described elsewhere in this report.

IPAD IS:

- AN INTEGRATED SYSTEM OF AUTOMATED MODULES.
  EACH DISCIPLINE IS RESPONSIBLE FOR ITS OWN CAPABILITY DEVELOPMENT, UPDATE & GROWTH
- A USER-ORIENTED & DIRECTED MODULAR SYSTEM WITH FLEXIBILITY FOR CHANGE, ADAPTATION & EXPANSION
- A HARDWARE/SOFTWARE COMPUTER SYSTEM DESIGN APPROACH
  TO PERFORM ENGINEERING DESIGN PROCESSES MORE EFFECTIVELY, ECONOMICALLY & SWIFTLY
- A COMPUTER SYSTEM STRUCTURE
  USABLE IN MANY ENGINEERING & SCIENTIFIC FIELDS
- ITS DATA BANK IS THE REPOSITORY FOR ALL DESCRIPTIVE & INFORMATIVE DATA GENERATED BY THE ENGINEERING/SCIENTIFIC TEAM FOR A SPECIFIC PROJECT
- A MANAGEMENT TOOL
  TO PROVIDE IMMEDIATE VISIBILITY INTO PRODUCT STATUS & PROGRESS
- INITIALLY, A REASONABLE ENGINEERING CAPABILITY (SET OF AUTOMATED MODULES) MOUNTED ON A STATE OF THE ART HARDWARE/SOFTWARE STRUCTURE THAT CAN BE READILY IMPLEMENTED
- ULTIMATELY, A COMPREHENSIVE, DYNAMIC ENGINEERING TOOL SUPPORTED BY EFFICIENT, COST-EFFECTIVE HARDWARE/SOFTWARE CAPABILITY
- AN EDUCATIONAL AID FOR TRAINING NEW ENGINEERS IN THE USE OF VARIOUS DESIGN PROCESSES

IPAD IS NOT

- A SINGLE, HARDWIRED COMPUTER PROGRAM
- AN AUTOMATED, SINGLE-PURPOSE PROCEDURE
- A DISLOCATED ARRAY OF RANDOMLY COLLECTED COMPUTER PROGRAMS
- A SYSTEM OF PROGRAMS TO BE RUN BY A SINGLE DISCIPLINE
- A SYSTEM OF PROGRAMS IMPOSED BY AN AGENCY (OR COMPANY) ON THE AEROSPACE INDUSTRY COMMUNITY

Figure 1-2. Major IPAD Features

2 THE CONCEPTUAL DESIGN

This section discusses the eventual role of digital computing leading up to the need for IPAD, the present day computing environment in which IPAD must function, and a conceptual design of IPAD. This section concludes with a preview of the selection studies required to develop the design requirements.

2.1 Eventual Role of Digital Computing

Engineers are responsible for defining the overall problem and solution concept, automating on computers where possible, instigating the programming or the choosing of a suitable simulation, organizing the compilation of input, suggesting or selecting an output format, providing adequate checks of the solution, and for distribution of the results. Typical examples of automated approaches depicting various degrees of engineering involvement are discussed in References 1, 2 and 3. Providing input formats and output displays, application of advanced diagnostics and cybernetics, research in the field of general computer techniques, and the compiling of computer manuals and teaching of computer orientation courses are the responsibility of the software engineers. A graphical representation of this scheme is presented in Figure 2-1.

Engineers, whether they eventually program or not, must become familiar with the overall concepts of digital computing and its flexibility. In the past, minicomputers have provided an excellent tool in instigating computer mechanization (Reference 4). In so doing, however, it was found that many, though realizing the greater potentials of the large scale scientific (maxi) computers, chose to remain with the mini. This can be attributed to a few fundamental reasons. For some, the mini is an engineering tool. Others desired the speed with which their results were obtained. Still others appreciated the control they achieve over their own programs on an "open shop" basis. In this way the individual can meet his due date without reliance upon another programmer. The utilization of the mini had been increasing at times faster that that of the maxi. It is obviously essential, therefore, that the basic philosophy of the mini mode of operation be instilled into a similar operating mode on the maxi.

Furthermore, for many people at present the computer in an interactive mode is a design tool rather than a production machine. An engineer is usually mechanically inclined and, as such, it is convenient, and usually faster, for him to visualize the approach and solutions to his design/engineering problems and the engineer obtains a deep insight into the fundamentals of the problem. The output is displayed in a form which is easy to utilize for most engineers and scientifically inclined personnel. In

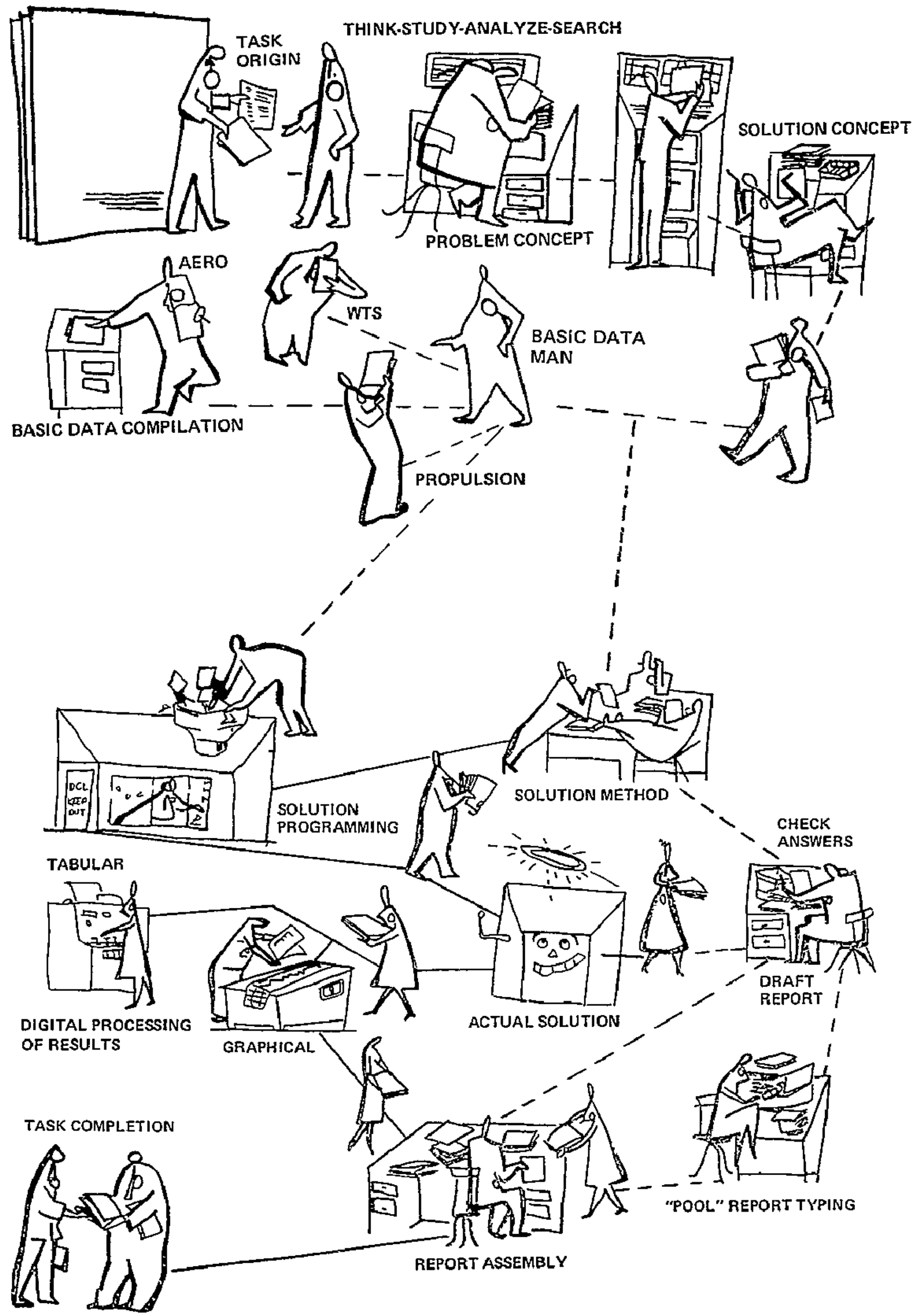


Figure 2-1. Evolution of an Engineering Task

short, the engineer establishes an affinity for the interactive approach to the solution of his problems. The equipment and the output display has many built-in features cybernetically designed for the engineer. Owing to these features, interactive design/ computing is an optimum engineering design tool.

The application of the digital computer to problem-solving has no real insurmountable limitations other than those of hardware and of approach philosophy. The problem solution concept can be thought of as a continuum, as suggested by Figure 2-2. This procedure can be made to encompass most of the problem-solving task.

INPUT TO ROUTINE	EVOLVING CONCEPTUAL INPUT TO A LOGIC FLOW ("MAPPING") TRANSLATING ENGINEERING "LANGUAGE" TO COMPUTER "LANGUAGE" CHECKING AND RECTIFYING LOGICAL ERRORS IN INPUT FORMATS REFLECTING MINOR LOGICAL DECISIONS IN INPUT TRANSMITTING PLOTTED DATA ONTO INPUT SHEETS ARRANGING PUNCH INPUT IN SEQUENTIAL FORM TRANSCRIBING DATA OUT OF FORM ONTO INPUT SHEETS PERFORMING MINOR CALCULATION CONCERNING INPUT PARAMETERS	[IPAD?]
DIGITAL ROUTINE (OM)	HI-SPEED ARITHMETIC CALCULATIONS & DATA MANIPULATIONS	
OUTPUT FROM ROUTINE	PERFORMING MINOR CALCULATIONS ON OUTPUT PARAMETERS SPOTCHECKING OPTIMUM CRITICAL OUTPUT PARAMETERS REARRANGING OUTPUT FORMAT TO SUIT PARTICULAR PURPOSE PLOTTING PORTIONS OF OUTPUT TRANSPORTING INFORMATION TO OTHER GROUPS TABULATING OUTPUT ON REPORT MASTERS SORTING, COUNTING & PERFORMING OTHER LOGICAL MANIPULATIONS CHECKING FOR LOGICAL ERRORS IN OUTPUT FORMAT TRANSCRIBING DATA INTO "N" DIFFERENT FORMS FOR SPECIFIC USES	

Figure 2-2. Evolution of the Problem Solution Continuum

Although current methods exist to solve these tasks an effort must be put forth to more efficiently utilize the computing equipment to simplify the design task and to reduce the time to perform it. Within the foreseeable future, engineering will be compelled to think in motion, and new equipment should be designed and built to achieve this end where possible.

2.2 The Computing Environment

To the system designer the computing environment consists of all of those factors which influence the design of IPAD. Paramount among these is the collection of individual users of IPAD (i.e., of the "system"). The characteristics of computer equipment and computer's operating system represent constraints in the design of IPAD. However, exploitation of an advanced operating system's features can significantly reduce the complexity of IPAD as will be seen.

The subsections that follow present the environment as it evolves from the available computer hardware, through the available computer's operating systems software and finally discusses some of the attributes of the most complex system element, the user.

2.2.1 The host computing system hardware. - Figure 2-3 illustrates a typical scientif computer installation, viz a CDC CYBER 70 series computing system. Typical input/o put peripherals consist of card readers/punches, line printers (listers), digital magnetic tapes, removable disk packs, and microfilm recorders and/or online plotters. Magnetic tape output typically feeds remote CRT (microfilm) plotter-recorders (e.g. an S-C 4020 which can mix vector-drawn images with extruded-character alphanumerics), paper-ink recorders (e.g. CALCOMP drum plotters and GERBER large flatbed drawing plotters), and numerically controlled machines (e.g. mills).

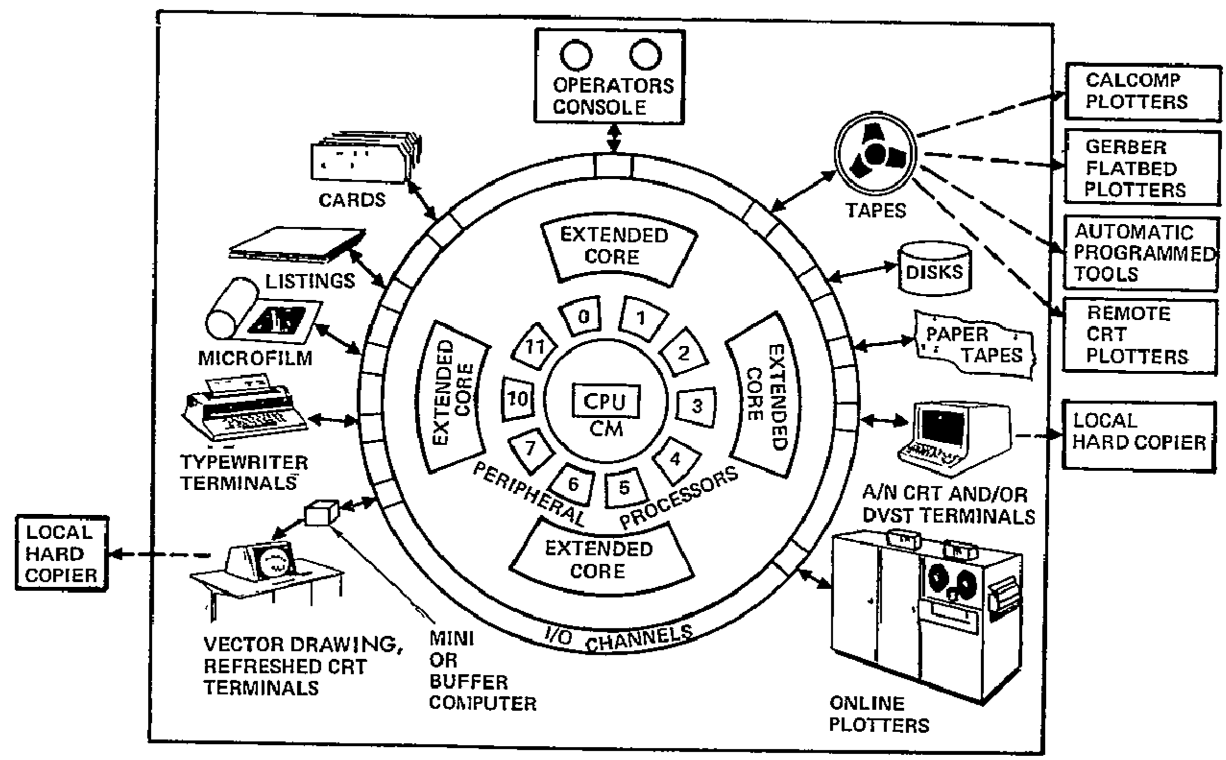


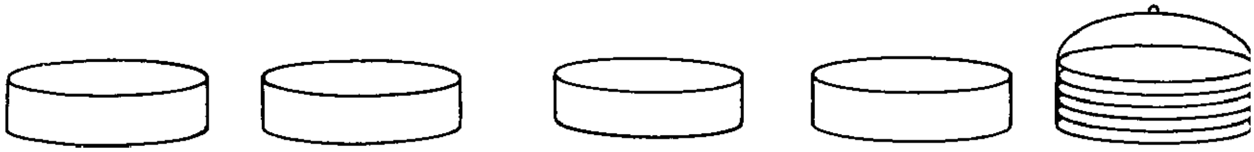
Figure 2-3. Host Computer and Peripherals (CDC's CYBER 70 Hardware)

Interactive computing remote terminals typically range from the simplest electromechanical typewriter I/O devices operating at about 10 alphnumeric characters per second (i.e. the speed of a skilled typist on an electric typewriter) to the most sophisticated large-screen, "vector-drawing" (can make graphic displays), refreshed CRT terminals, usually operated through a minicomputer (or simpler buffer computer) at up to 4000 characters per second. The refreshed CRT constantly refreshes the beam-deflected CRT image at a refresh rate above the flicker threshold and is usually equipped with both keyboard alphanumeric input as well as a "topological" I/O device such as a light-sensitive pen. Midway between these two terminals is the alphanumeric CRT and Direct View Storage Tube (DVST) terminals, both equipped with alphanumeric keyboard for input and operating at higher communication rates than the typewriter terminals (e.g. 30 to 600 characters per second). The DVST terminals are vector-drawing but need not be refreshed; they employ a special retentive phosphor and special circuitry which retains the image with very slow decay (until repainted). The CRT devices typically are equipped with local hardcopy devices or utilize the recorders/plotters peripheral to the host computer.

The schematic of the host computer (Figure 2-3) is idealized to better convey the design's intent. Surrounding the central processor (arithmetic) unit (CPU) is the central memory (CM), usually a fast access core memory. The CPU communicates with the CM in processing almost every instruction, consequently the packaging of the CM and the lead lengths between CM and CPU are the limiting speed factor. Communicating with the CPU through CM - a CDC design feature - are up to twenty (ten shown) peripheral processors (PPs) which are small self-contained computers (with their own small memories) which handle tasks peripheral to scientific computing (e.g. input/output with peripheral devices). Optional is extended core storage (ECS) which can operate as an extension to CM and/or as a high-speed I/O buffer device. ECS is typically slower than CM and of lower cost; it is expandable in modules up to several million storage locations. The peripheral processors (PPs) communicate with the peripheral devices through the I/O channels. One special peripheral is the operators' console which is serviced by one of the PPs and provides the interactive interface with host computer's operating system. The operating system's monitor is resident in one PP.

The operating system and support utilities generally reside on disk. Figure 2-4 illustrates a typical disk allocation in a large computing complex. The operating system library contains the various software entities which support the system, e.g. PP programs such as the Resource Allocator (1RA) which assigns resources (e.g. a magnetic tape unit) to an incoming job or the disk Stack Processor (1SP) which processes word strings from (to) CM (or ECS) to (from) disk. Also supporting the user are various system utilities such as compilers, assemblers, translators, etc.

Each job (run unit) is assigned disk residency prior to execution. This job residency is associated with the various job queues as well as the job's files (e.g. input, output, scratch). The user (in advanced installations) can elect to retain (allocate) permanent disk residency to certain files; the accounting files associated with the various jobs are one such example. Special files (usually large data files, e.g.



USER UTILITIES

- COMPILERS
  - FORTRAN
  - ALGOL
  - BASIC
  - JOVIAL
  - SPL
  - COBOL
- ASSEMBLERS
  - MACHINE LEVEL
- TRANSLATORS
  - APT
  - MIMIC
  - CSMP
  - GPSS
  - SIMSCRIPT
  - SCEPTRE
- UTILITIES
- FILE MANAGERS
- TEXT EDITORS

OPERATING SYSTEM LIBRARY

- RESOURCE ALLOCATOR
- JOB SCHEDULER
- RECORD MANAGER
- JOB PROCESSOR
- INTERACTIVE MONITOR
- FILE AUDITOR
- JOB SWAPPER
- LOADER
- REAL-TIME MONITOR
- TAPE SCHEDULER
- JOB TERMINATOR

JOB RESIDENCY

- INPUT QUEUE
- OUTPUT QUEUE
- LOCAL FILES
- SCRATCH FILES

USER RESIDENCY

- PERMANENT FILES
- ACCOUNTING FILES
- SYSTEM FILES

USER DISK PACKS

- SPECIAL DATA
- PREVIOUS RESULTS

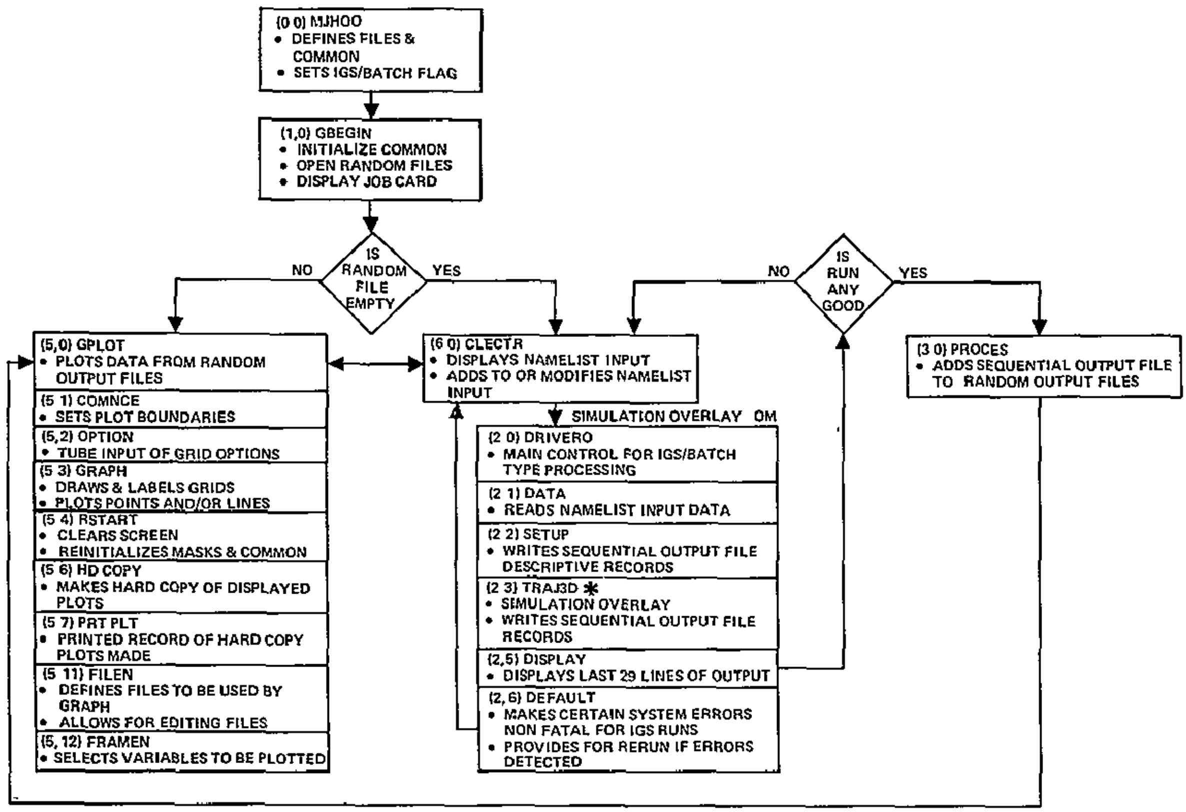
Figure 2-4. Disk Allocation, a Typical Installation

wind tunnel data) may be allocated to removable disk packs. Thus many users may utilize a single disk unit which contains different - yet permanent - information at different times throughout the computing day.

Note that all the files typified in Figure 2-4 have concurrent disk residency requirements.

An unusually large amount of information may reside "permanently" on disk - at least permanent for the duration of a comprehensive engineering study (some several weeks) - usually requiring a large number of disk-pack units and sophisticated user scheduling. NAMESIM (originally designed for simulations employing the FORTRAN IV NAMELIST input feature) is an example of such a computer system - actually a microcosm of IPAD - requiring unusually large disk residency and will be discussed briefly to illustrate these mass-storage demands.

Figure 2-5 illustrates the program structure of NAMESIM. Although programmed as a conventional three-level overlay structure, it is not represented as such in t figure since it operates from a random task file whereby any overlay level may be call into central memory from any point within the program. (Caution must obviously be exercised in programming such a structure.) The program structure shown is representative of its operational structure.



* Currently one of three OMs Implemented
P4322 Re entry Vehicle/Aircraft Re-entry Flyback & Approach Simulation
P5255 Clustered Lifting/Entry Vehicle Staging Simulation
P5458 Clustered Lifting/Entry Vehicle Ascent Simulation (Including Real Winds Aloft)

Figure 2-5. NAMESIM Structure, An Operational Microcosm of IPAD

The computation core of NAMESIM is the (2, 0) overlay (center of figure) which currently can consist of any of three continuous (in contrast to discrete) variable simulations:

1. P4322, a re-entry vehicle or aircraft re-entry and flyback/approach simulation.

2. P5255, a clustered lifting-entry vehicle (e.g. Space Shuttle) staging simulation.

3. P5458, a clustered, lifting-entry vehicle (e.g. Space Shuttle) ascent simulation, including the ability to simulate real winds aloft.

All simulations are programmed substantially the same except for the (2,3) overlay (see Figure 2-5) which is the application simulation itself. Overlay (2, 1) reads in NAMELIST input data, (2,2) sets up a sequential output (disk) file by writing descriptive output records for subsequent processing, (2,6) defaults certain system errors and provides for error recovery if the program is being run interactively, and (2,5) displays the last 29 lines of (tabular) output information printed by (2,3) and allows the user to "page through" this output file at the interactive CRT terminal.

If the user interactively decides - by "paging through" the output listing at the CRT - that the simulation is not worth processing, he can elect to return to overlay (6,0) to display the NAMELIST* input and provide modification and rerun of the simulation. A simulation or system error will return him to the same point. Thus the user can make repeated runs (in a short period of time) in an attempt to obtain a satisfactory simulation for more comprehensive analyses.

When an output listing is deemed ostensibly satisfactory, the user elects to continue to overlay (3,0) to process the sequential output file - set up by (2,2) and written by (2,3) - to a random-access file for ease of plotting and for permanent disk storage.

Overlay (5,0) allows interactive plotting (graphing) of results with the ability to select variables to be plotted, set plot boundaries (automatically labeling graphs), obtain hardcopy (via a remote microfilm recorder), etc. The user then has the option to return to overlay (6,0) to rerun the simulation or to terminate the interactive job; before either decision he must decide whether to retain the previous run as permanent (which is done immediately to avoid "catastrophic" failures which had lost several hours of good work prior to this NAMESIM modification). Upon initiation, NAMESIM will automatically enter the interactive plotting module unless the random output file is empty.

Figure 2-6 presents disk usage statistics as sampled on March 22, 1972. The total disk usage is composed of the NAMESIM family of programs (4 percent of total), the microfilm recorder (SC 4020) hardcopy queue (1 percent of total) which is periodically dumped to magnetic tape (as a cost saving feature in that the cost per graph drops substantially as the number of graphs to be processed increases), the random winds file to support simulation P5458 (12 percent of total, representing only the windiest month of March), and the output random files (83 percent of total) at some point within an engineering study.

---

* Subsequent modification to NAMESIM have removed this restriction by providing for either free-form (NAMELIST) or fixed-form (conventional) input.

• TOTAL DISK REQUIREMENTS, 48383 SECTORS (968 HALF TRACKS)			
• NAMESIM SYSTEM (ABSOLUTE OVERLAYS),			1979 SECTORS
	. WITHOUT OM'S	755	
	. P4322 ONLY	405	
	. P5255 ONLY	443	
	. P5458 ONLY	376	
• OUTPUT RANDOM FILE SIZE,			40226 SECTORS
	. P4322	8888	
	. P5255	13463	
	. P5458	17875	
• SC4020 HARDCOPY MICROFILM QUEUE,			474 SECTORS
	. P4322	174	
	. P5255	65	
	. P5458	235	
• P5458 RANDOM WINDS FILE (JYMSPHERE WINDS, ONE MONTH),			5704 SECTORS

Figure 2-6. NAMESIM Disk Usage Statistics
(Typical: as of 03/22/72)

The total disk usage of 48,383 sectors represents approximately 24 million characters of mass storage in support of just three engineering simulations! Note that these disk requirements do not include NAMESIM job residency requirement (e.g. input, output and scratch files) other than the permanent I/O files listed.

Figure 2-7 completes the picture of NAMESIM by illustrating typical NAMESIM response times for various task segments. The large delays encountered between ideal conditions (one interactive user and sparse batch background) and poor conditions (two interactive graphics users with heavy disk I/O plus several conventional interactiv terminals with moderate batch background), points out that excessive demands can easi be imposed on the computing system. These large delays are principally the result of:

1. Job swapping (rollin/rollout) speed (here disk access).
2. Scheduling algorithm used (Convair's algorithm does not interrupt disk I/O which is in process).

3. Tasks delegated to the interactive graphics minicomputer. Our CDC 274 Interactive Graphics System utilizes the CDC 1700 minicomputer principall as a buffer computer, i.e. each "button pick" is a host computer interrupt.

Task	Typical (One User) SEC.	Slowest (Two Users) SEC.
• REACTION TO SPECIFIC REQUEST (BUTTON PICK)	<1	60
• SETUP OF INITIAL DUAL PLOT REQUEST (FILES, VARIABLES)	30	150
• REPLOT WITH CHANGE OF VARIABLES	~10	>100
• GENERATION OF PLOT HARD COPY (3 FILES, DUAL PLOTS)	~3	>30
• SETUP OF NEW RUN THROUGH NAMELIST	120	300
• EXECUTION OF SIMULATION OVERLAY	~60	~600
• PROCESSING SEQUENTIAL FILE TO RANDOM FILE	~120	~600

Figure 2-7. NAMESIM Response Time Statistics (Typical)

NAMESIM operation has additionally established a user preference for a 45-minute average console residency with approximately five cases investigated per console sitting (following the initial debug phase which typically averages a case per minute).

2.2.2 <u>The host computing system software (i.e., the operating system software).</u> - The IPAD design objectives as they relate to the host computer operating system are:

1. The IPAD system design shall be open-ended; limitations shall arise only through the host computer's hardware/software constraints rather than IPAD's design approach.

2. The developed IPAD software shall be as transferable to other computer installations as is practicable.

3. Maintenance and modifications required by IPAD to achieve increased capability or to retain an acquired capability during a computing system upgrade shall be minimized.

4. Since the host or main computer hardware configuration and operating system can have a significant effect on the partitioning of computing functions (computation, I/O, display generation, and data management), any IPAD design must take into account the host computer system for which it is intended.

Several IPAD design approach options can immediately be envisioned and examined on their own merits. First, an examination of available operating system features is given.

2.2.2.1 Review of operating system features: Figure 2-8* summarizes the host computer's operating system features deemed important to IPAD together with the availability of these operating system features on three major, large scale scientific computing systems:

1. Control Data Corporation's (CDC's) 6000 series.
2. International Business Machine's (IBM's) 370 series.
3. UNIVAC's (a division of Sperry Rand) 1100 series.

There is little question that access/retrieval times for current and projected mass storage hardware are going to require random access file structures. An advanced file structure, index sequential, combines the features of both random and sequential files e.g., allowing the random positioning of the file for speed and the subsequent retrieval of information sequentially for retrieval speed and convenience. As noted in Figure 2-8, all three computing systems provide these file structures.

A necessary requirement of IPAD's data banks is that these must reside on permanent (or at least semipermanent) storage. This is necessary so that the original data for a given design can be available in an unaltered form throughout the several weeks when the design is being evolved and evaluated. As alternate designs and design improvements become defined, these too must be stored - preferably as modifications to prior data - to reflect the current design alternatives in a fully updated form. It follows that card or magnetic tape mass storage of computerized data is not sufficient. Card storage of data is of low density, awkward to handle (spillage) and difficult to store (bulk, weight and environmental requirements) in large quantities. Magnetic tapes are ideal in contrast to cards, however they are only suitable for storage of large, sequential files. The physical structure of magnetic tape necessitates sequential access and thus does not support the random access requirement stated above. For storage of large amounts of data, the data content must be large since small minireels defeat many of the advantages of magnetic tape (e.g., they are effectively of low density due to the large reel-to-tape ratio and hence larger bulk). Further the access time of either cards or tape must of necessity be large due to the re-

*Note that the figure and discussion were concluded prior to the announcement of the VM/370 operating system by IBM (i.e., November 1972).

SOFTWARE FEATURE	AVAILABLE UNDER CDC SCOPE? (VERSION?)	AVAILABLE UNDER IBM OS?	AVAILABLE UNDER UNIVAC EXEC8?
• RANDOM ACCESS FILES (FORTRAN)	YES(3 1 2)	YES	YES
• INDEX SEQUENTIAL FILES (FORTRAN)	YES(3 3)	YES	YES***
• PERMANENT FILES	YES(3 1 6)	YES	YES
• UPDATE UTILITY	YES(3 2)	YES	YES
• INTERACTIVE COMMUNICATIONS	YES	YES	YES
• INTERCOM 2 0	YES(3 3*)	–	–
• INTERCOM 3.0	YES(3 3)	–	–
• INTERCOM 4 1	YES(3 4)	–	–
• TIME SHARING?	YES	YES	YES
• MEMORY SHARING?	YES	YES	YES
• INTERACTIVE GRAPHICS SYSTEM	YES**	YES**	YES**
• IGS VERSION 1	YES(3 1 2)	–	–
• IGS VERSION 2	YES(3 3)	–	–
• TIME SHARING?	YES	NO	YES
• MEMORY SHARING?	YES	NO	YES

*We Retrofit INTERCOM 2 0 Into SCOPE 3 2
**Recognizing That Graphics Language & Architectures Differ
***But Difficult With Fortran (Best Use Assembly Language)

Figure 2-8. Summary of Currently Available Software Features Deemed Important to IPAD

quired physical placement and retrieval of the store by human operators. This limits the applicability of these storage devices. Until advanced mass storage techniques become practical (e.g. laser memories), reliance remains on magnetic disk with removable disk packs. The operating system must then provide for permanent files on disk. Further, utility software must be available to selectively update prior data; it is preferred that this UPDATE UTILITY provide the ability to retain data prior to the modification to make both unmodified and modified data available simultaneously. Although this can be done by retaining two complete copies (the unmodified and modified copies), storage implications make such an approach prohibitive.

All of the three major computing systems provide an interactive communication subsystem as a part of their operating system. As noted for CDC, these systems are continuously upgraded; INTERCOM 4.1 (of latest operating system version, SCOPE 3. provides an expanded, reentrant TEXT editor as one of its improvements. These interactive communications subsystems currently provide both memory and time sharin Time sharing is the ability to work on another (perhaps also interactive) job (multiprogramming) and to share the available time among the interactive jobs in such a way as to provide "prompt" response to user requests. If the computer's central memory were large enough to contain all (at least all interactive) jobs simultaneously, there would be no need (efficiency aside) to provide a memory sharing capability; memory sharing is the ability to remove a temporarily inactive job from central memory (usually to disk but sometimes to remote core storage) and replace it with a job currently requiring service (e.g., a previously temporarily-inactive job requesting reactivation). With a large number of interactive jobs in the system at the same time, the demands placed on conventional disk for memory sharing (job swapping) can significantly degrade the response time for every interactive user utilizing the system at any one time.

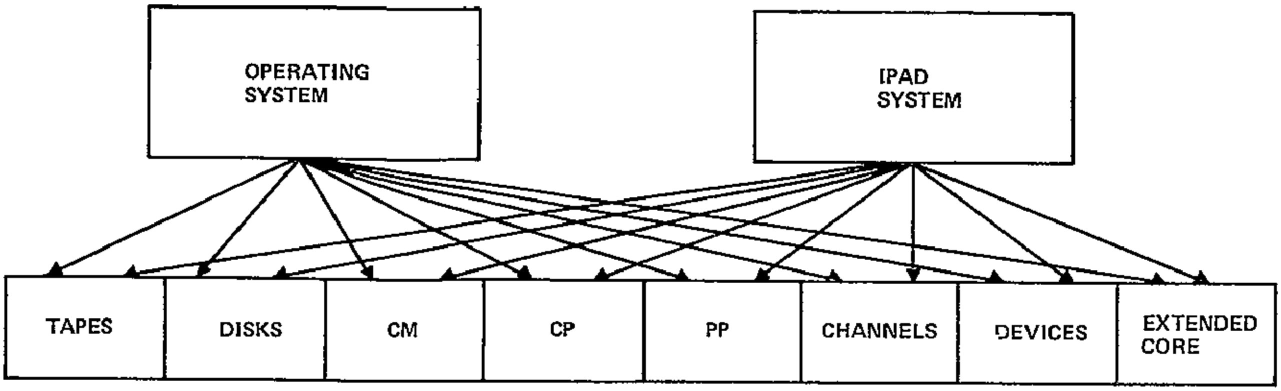
Historically, the development of interactive graphics subsystems has parallelled that for interactive communications subsystems. Interactive graphics provides the capability to draw vectors (e.g. graphs, drawings, pictures) on a CRT device in response to interactive commands. The earlier subsystems have evolved around moderate to large screen CRTs and special purpose buffer computers and other related hardware. As noted in Figure 2-8, IBM's OS* operating system on their current 370 line cannot time-share or memory-share interactive graphics jobs since these jobs are not interactive communications jobs as viewed by this computer's operating system (CMS on the Model 67). Development is underway (e.g., CDC's General Purpose Graphics Terminal, GPGT) to establish interactive graphics subsystems as a part of the interactive communications subsystem thus permitting enjoyment of those features normally a part of this latter subsystem (e.g., request to do some conventional processing before, after and even during interactive graphics jobs); currently none of the three major systems denoted on the figure provide this capability.

Although all three computing systems provide interactive graphics subsystems, it should be noted that the graphics language - and even its architecture - differ among these systems. A graphics job operational on one computing system will not run (without extensive modification) on a different system. This is in contrast to a conventional FORTRAN IV jobs which (with some notable exceptions) run adequately on any of these computing systems (even the interactive communications subsystems) without (or with extremely slight) modifications.

We are now in a position to examine the various design approach options to incorporate IPAD into the computational complex.

2.2.2.2 System level design of IPAD: Figure 2-9 presents an overview of IPAD designed at the system level. This design is the most comprehensive because IPAD acts as its own operating system; hence it requires the least capable host computer operating system. It has potentially the fastest response time - for IPAD users - since the operating system portion of IPAD is dedicated to IPAD.

*TSO — which supports the IBM 2250 graphics console — only time-shares and memory-shares jobs within a dedicated fixed partition.



ADVANTAGES
- REQUIRES LEAST CAPABLE OPERATING SYSTEM
- FAST RESPONSE TIME

DISADVANTAGES
- SYSTEMS COMPETE FOR RESOURCES
- DUPLICATION OF SOFTWARE
- SYSTEM LOCKUPS MORE POSSIBLE
- MOST DEPENDENT ON COMPUTER HARDWARE/SOFTWARE
- LARGEST INVESTMENT
- SHORTEST LIFE

Figure 2-9. System Level Software Design

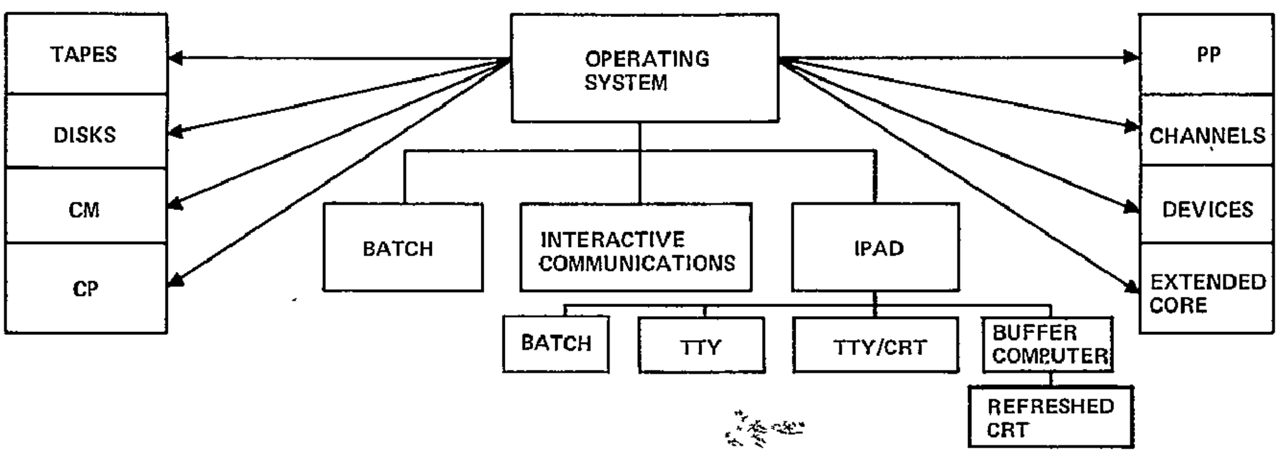
The disadvantages however are numerous and devastating. Principal among these is the heavy competition for resources between IPAD and the host operating system which produces extensive conflict and inefficiencies. These, in turn, make irresolvable conflicts (system lockup) a distinct possibility and will probably require the computing system being dedicated to IPAD in order to achieve reliable operation.

A perhaps surprising consequence is that this IPAD design will be the most dependent on the host computer's hardware/software architecture. This follows directly from the fact that IPAD must act (principally) as its own operating system in scheduling interactive jobs, allocating resources, etc. To do so requires dealing directly with the hardware architecture and software-primitives and thus becoming dependent on this non-transferable code. It follows also that there will be an unavoidable functional duplication of software as IPAD accomplishes the same functions as typically accomplished by host operating systems (e.g. interactive job scheduling); this in turn requires the largest investment to achieve an operational IPAD.

The most damaging disadvantage from the system designer's viewpoint is un-undoubtedly the short life enjoyed by the resulting system. Being tied to specific computer system architectures, IPAD will be affected each and every time these systems are altered. As the systems' architectures evolve (as they inevitably do) these required modifications become more extensive until it becomes clear that reprogramming IPAD - even in the moderately short run - might be less costly than continuing piecemeal modification. Further, many host operating system features which subsequently become available will not become available to IPAD due to the modifications involved. This will accelerate the aging process until IPAD offers few advantages over the then current operating systems. This, in turn, leads to IPAD's premature demise and substantial waste of its original investment.

It is clear that IPAD must exploit - to the maximum extent possible - the host computer through the host computer's operating system.

2.2.2.3 Subsystem level design of IPAD: Figure 2-10 presents an overview of IPAD designed at the subsystem level, i.e. subordinate to the operating system but at the same level as the operating system's subsystems (e.g., the level of the interactive communication subsystem). The immediate advantages are apparent in the substantially reduced competition for resources; the host system resolves conflict decisively (even if arbitrarily) thereby eliminating the potential for system lockup. Further, most of the duplication of software has been avoided thus requiring less software development and making IPAD less dependent on specific computing system hardware/software architecture.



ADVANTAGES

- LESS COMPETITION FOR RESOURCES
- MUCH LESS SOFTWARE DEVELOPMENT
- LESS HARDWARE/SOFTWARE DEPENDENT
- OBTAINS UPDATED OPERATING SYSTEM CAPABILITY (PRACTICALLY) GRATIS

DISADVANTAGES

- REQUIRES MODERATELY CAPABLE SYSTEM (MUCH OF IPAD CAPABILITY SUBSET OF SYSTEM CAPABILITY)
- SLOWER RESPONSE TIME
- STILL COMPETITION BUT ONLY AMONG INTERACTIVE DEVICES

Figure 2-10. Subsystem Level Software Design

The most significant advantage is the updated capability continuously afforded IPAD through inevitable extensions to the host operating systems' capabilities (providing of course that these don't conflict with IPAD). These upgrades come to IPAD practically gratis since they are obtained through the host operating systems. They also are available to IPAD when first released, viz. without undue delay.

The principal disadvantage is obvious - IPAD now requires a moderately capable operating system since most of IPAD's capability is in actuality a subset of the operating system's capability. This disadvantage is unavoidable and was dictated by the desire to exploit the host operating systems.

A subordinate disadvantage is potentially* slower response times since the host system is attempting to provide adequate service to all its subsystems, including IPAD. Since these subsystems accomplish somewhat the same functions, albeit in different ways, there still exists competition for resources; however the scheduling algorithm will insure that this competition is limited to the interactive devices competing for the attention of the operating system.

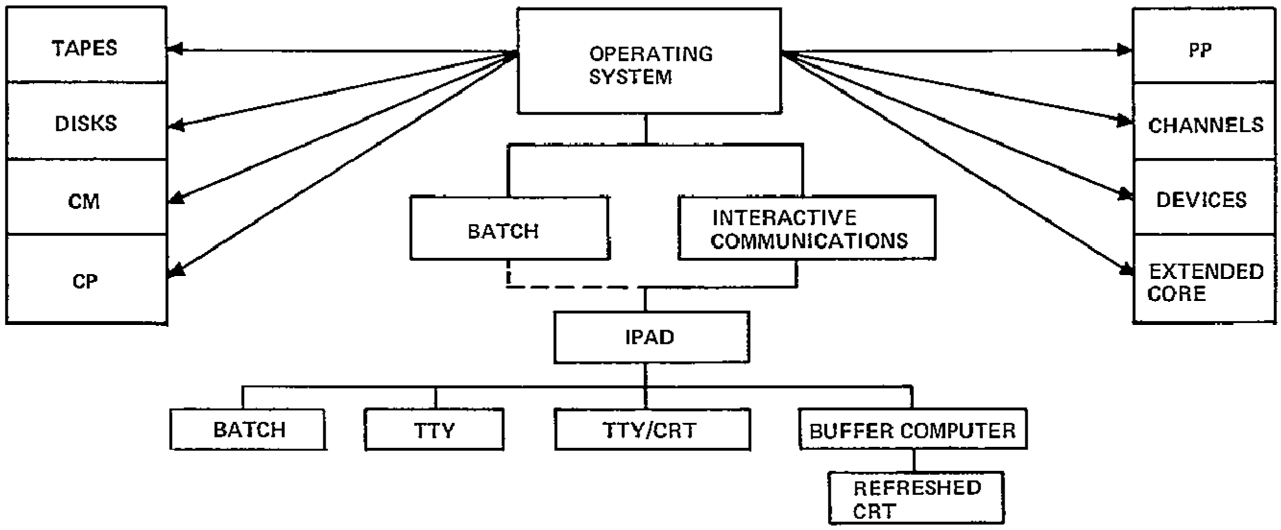
Much was gained by subordinating IPAD to a moderately capable host operating system without, however, exploiting the host system to the maximum extent possible.

2.2.2.4 Sub-subsystem level design of IPAD: Figure 2-11 presents an overview of IPAD designed subordinate to the host operating system's subsystems, i.e., an IPAD job looks like a standard job operating within the framework of the host system. Figure 2-12 presents an optional version which differs only in that the buffer compute for the most capable refreshed CRT interactive graphics terminal has been replaced by a reasonably capable, dedicated minicomputer. Figure 2-11 will be discussed firs

The advantages are obvious from the preceeding discussions and contain the appropriate superlatives:

1. Least competition for resources since IPAD looks like a standard job to either the batch or interactive-communications subsystems.
2. Minimal software development since existing system software is being fully exploited.
3. Least hardware/software dependence since the bulk of IPAD's dependence is interfaced (buffered) through the host operating system.
4. Least system impact - IPAD looks like standard job.
5. Potentially longest life since, being a "standard" job, IPAD will continue to be supported far into the future, possibly until standard host operating

* Potentially since these can be controlled by adjusting the host system's scheduling algorithm.



ADVANTAGES

- LEAST COMPETITION FOR RESOURCES
- OBTAINS UPDATED OPERATING SYSTEM AND INTERACTIVE SUBSYSTEM FEATURES (PRACTICALLY) GRATIS
- MINIMAL SOFTWARE DEVELOPMENT
- LEAST HARDWARE/SOFTWARE DEPENDENT
- LEAST SYSTEM IMPACT
- POTENTIALLY LONGEST LIFE

DISADVANTAGES

- REQUIRES HIGHLY CAPABLE SYSTEM SOFTWARE (IPAD IS A SUBSET OF INTERACTIVE COMMUNICATIONS SUBSYSTEM)
- POTENTIALLY SLOWEST RESPONSE TIME
- POTENTIALLY LARGE VARIATIONS IN IPAD SYSTEM OPERATION BETWEEN INSTALLATIONS

Figure 2-11. Sub-subsystem Level Software Design

systems themselves offer all the advantages accrued through IPAD (obviating the need for IPAD).

Again the most significant advantage is the continuous upgrading of IPAD through obtaining (practically) gratis the host system's latest features, including those of all of its subsystems.

The disadvantages are that the host operating system is now required to be highly capable since IPAD has divested itself of all host system functions. The response time (in practice) will be potentially the slowest - at least initially - since the computer system managers will attempt to balance the requirements of all users; it would be difficult (e.g., requiring host operating system coding changes) to differentiate between an IPAD interactive user and a non-IPAD interactive user, nor would such overt differentiation likely be acceptable to non-IPAD users.

However we have now acquired a new disadvantage arising principally from subordinating IPAD to the interactive communications subsystem; it is now likely that there will be large variations in IPAD system operation between installations of different computer manufactures. This follows since much of IPAD's command language need actually only be the command language of the host computer's interactive subsystem for which there is no standardization. Further there will be operating system features which differ between installations. These will be exploited by IPAD and non-IPAD users alike unless standardization is achieved by withholding this capability from IPAD (making IPAD's capability the set-intersection of the capabilities of the target computing systems - an approach of dubious merit).

Figure 2-12 illustrates an optional version of an IPAD design at the sub-subsystem level, viz. the use of a dedicated minicomputer for refreshed CRT users. The principal advantage accrues only to those users operating through the minicomputer, viz. faster response time since many interactive functions will be local to the dedicated minicomputer (perhaps shared by several interactive terminals) and hence accomplished without resorting to the host computer. Since these functions will no longer require the host computer, the host operating system will be able to service the other users more efficiently.

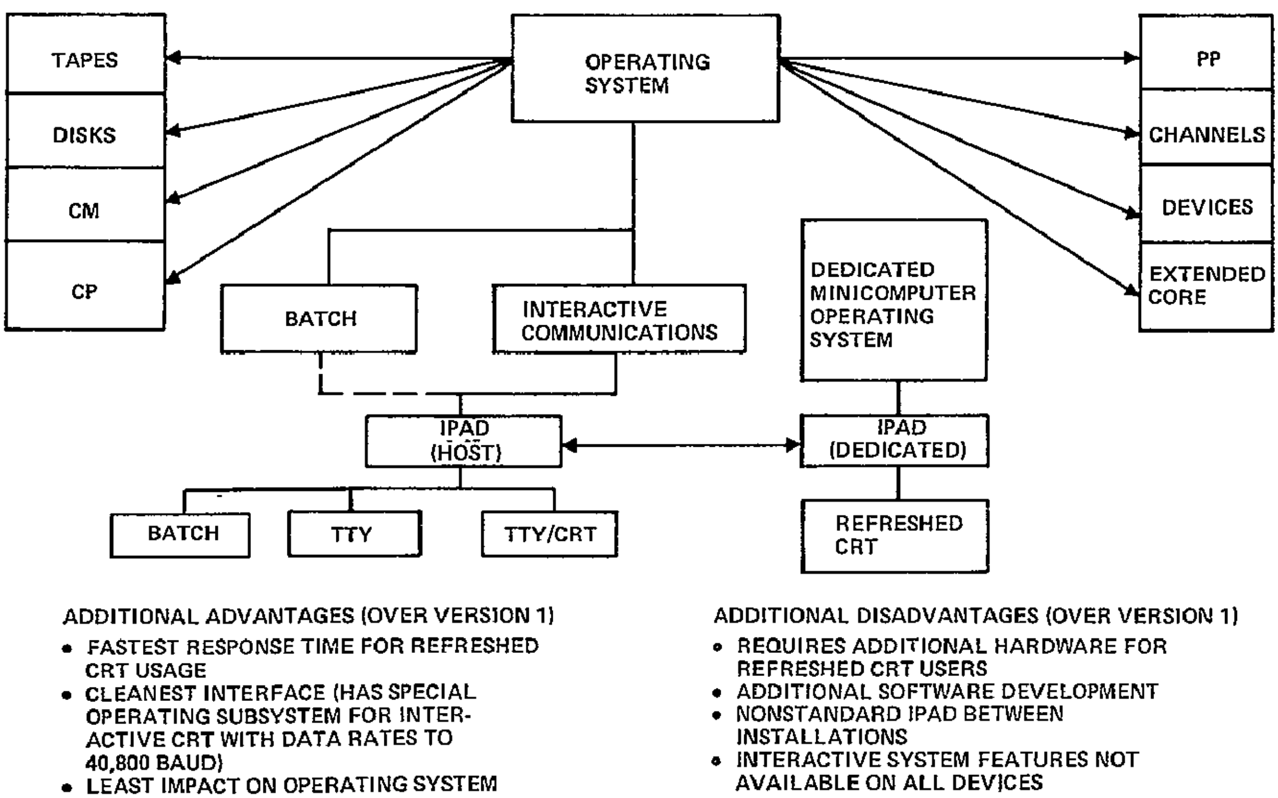


Figure 2-12. Sub-subsystem Level Software Design With Optional Dedicated Hardware

Further, the IPAD system software can be split between that residing on the host system and that residing on the minicomputer. This will result in the least impact on the host operating system since it is only the IPAD system on the mini which requires the interactive communications subsystem with very high data transfer rates (e.g., 40.8Kbaud). This split will enable a clean interface between the host and mini IPAD systems.

The disadvantages are principally that it requires additional hardware to utilize the highly capable refreshed CRT terminals and additional software development to provide the IPAD systems for the dedicated mini. The problems of addressing several target minicomputing systems, although not as severe, parallel those discussed for host computing systems. The dedicated IPAD software development could become significant if the target minicomputers differed much in architecture (hardware or software). Further, the minicomputer dedicated IPAD capability would likely be nonstandard between installations employing different minicomputers; interactive features available through one mini's operating system may not be available on another's system.

However, not every institution intending to use IPAD may be willing to provide the dedicated minicomputers with their attendant identifiable costs. The only realistic approach is to provide an IPAD system which optionally utilizes minicomputers for the refreshed CRTs; this necessitates that the IPAD system on the host be able to (optionally) service the refreshed CRTs thus effectively eliminating all but the first advantage listed in the figure, which would now accrue only to those users at those institutions employing the optional minicomputers. Since the IPAD system split is not possible in this case, the additional software would be just that required for the mini with minor tie in to the host IPAD system; i.e., the host system becomes essentially the one required under the first version (Figure 2-11).

It is not clear whether the advantages accrued to the refreshed CRT users through the use of minicomputers justify a parallel development of yet another dedicated IPAD system, especially since many of the large, refreshed CRT terminals are vended with appropriate minicomputers as part of their system. An alternate approach is to allow the vendors of such systems to repackage/reprogram available IPAD system software (following IPAD's release) for their systems and to sell/lease these systems based strictly upon economic incentives.

It is premature to resolve this problem, since it depends on the complexity of the IPAD suggested design and the host operating system capabilities to be exploited. This question must be answered after the recommended IPAD system design has been established. (The reader is deferred to the conclusion section of Volume V, Part III of this report).

2.2.3 The user. - The most essential part of the environment that the system designer must consider when evolving a system design is the potential users of such a system. Only the user exhibits such diversity, operational complexity, variability, sensitivity (e.g. to response time) and frustratability. The scarcest resource in the aerospace field in undoubtedly the creativity of the human mind. The development and use of this resource is inhibited by the tedious, rote work which requires full or near-full concentration to use the computer in today's batch-mode environment. Online interaction with programs, data bases and symbol manipulating through interactive terminals affords the opportunity to put man back in the driver's seat in the design process thereby opening up new dimensions of creativity which he previously could never attain. These human factors considerations must be reflected in the IPAD design.

The following subsections discuss some of the objectives of any system which must interface with human operators.

2.2.3.1 Applicability: IPAD must be applicable to users of highly diverse backgrounds with only the command terminology changed to suit the users. In particular, the system design approach should make provisions for the non-engineering user.

Figure 2-13 illustrates some of the engineering disciplines that must be represented within IPAD. These range from Advanced (Preliminary) Design through detailed design Drafting and Tooling Engineering. In addition to the traditional engineering disciplines, Management, Market Analysis, Scheduling, and Manufacturing can benefit significantly from an IPAD system. Non-engineering aerospace disciplines such as Accounting, Budgets, and Personnel should not be excluded from using IPAD due to inadvertent restrictions brought about through an excessive engineering predisposition in IPAD's design approach.

ADVANCED DESIGN	MISSION REQUIREMENTS	PERFORMANCE	AERO-DYNAMICS	PROPULSION
SPECS & CRITERIA	CONTROL DYNAMICS	WEIGHTS	PRELIMINARY DESIGN	COSTS
LOADS	STRUCTURAL ANALYSIS	STRUCTURAL DESIGN	STRUCTURAL DYNAMICS	THERMO-DYNAMICS
SYSTEMS	DRAFTING	TOOLING	DATA MANAGEMENT	MANAGEMENT
MARKET ANALYSIS	SCHEDULING	RISK ASSESSMENT	DRAWING RELEASE	MANUFACT'G

Figure 2-13. Typical Engineering Disciplines to be Represented Within IPAD

2.2.3.2 The role of interactive computing: The IPAD system must allow a user either to monitor an OM during execution or spinoff the task to the background job stream. The use can then proceed directly to other tasks which can be operated in parallel. Response time is an extremely important human factors consideration.

The question, "Why is interactive computing the key to a successful IPAD implementation?" can be answered simply:

1. Without the reduction in both calendar time and manhours that result from the use of interactive computing techniques, IPAD cannot be a cost effective tool. (This is from the standpoint of both amortizing the implementation cost and particularly the operational use costs.)

2. Without the visibility and insight available through interactive (particularly graphic) displays and the rapid access provided by interactive-on-line-response, the user simply cannot adequately monitor, much less control, the complex design process in a fully computerized environment. In fact, it is not handled adequately in the present "compartmentalized" approach using batch techniques with some interactive procedures.

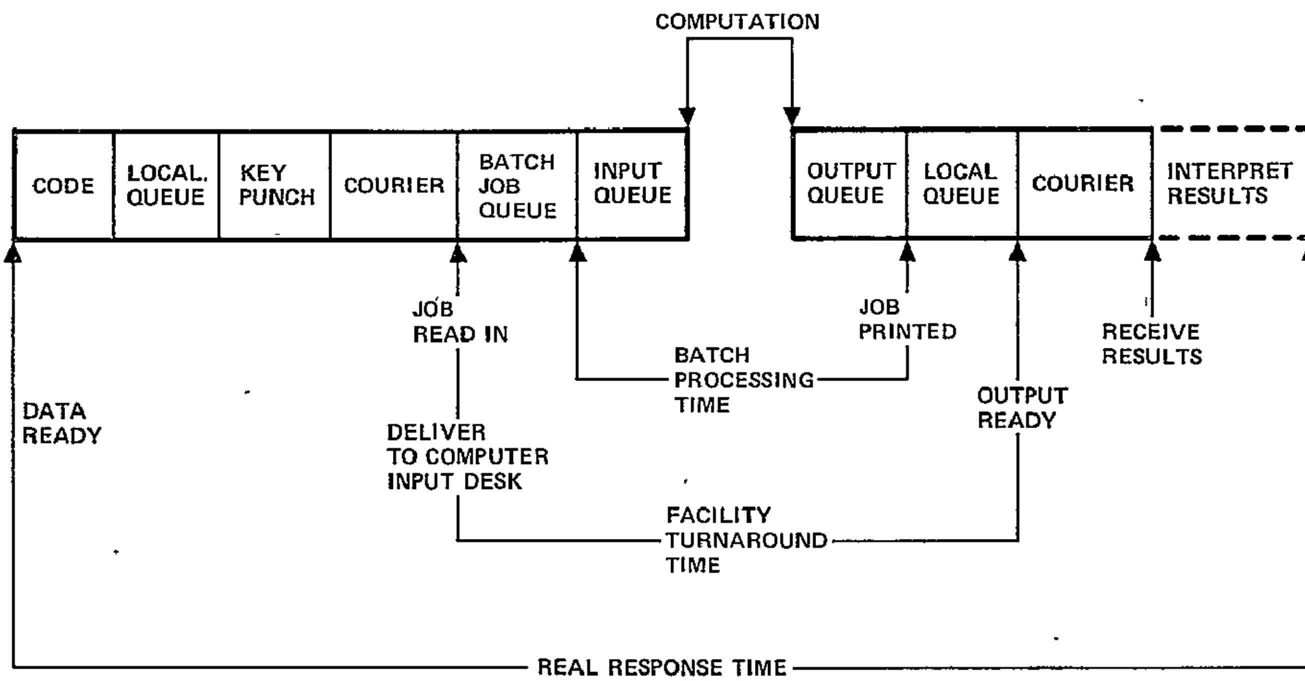
IPAD to be an effective, viable design tool must be truly a Computer Aided Design (CAD) system as defined by the DOD/Industry Symposium on CAD and CAM/NC (Computer Aided Manufacturing/Numerical Control) held at Davenport, Iowa in October 1969 (Reference 6):

> "CAD is the application of computers to design where the designer converses directly with the computer by using a graphic or non-graphic console in such a manner that his problem-solving processes are highly responsive and essentially uninterrupted."

The interactive, on-line aspects of the system are the key elements in reducing the response time for a specific solution (problem step) and shortening the overall product design time. There is more to this time saving than the actual time spent in waiting for results and hunting for errors. Inherent in the batch mode of using the computer is the need for the engineer to accomplish other tasks while waiting for results from a run in order to proceed on a given job. In a number of cases this approach works out very well, but there are many instances when schedules and problem complexity make it difficult to switch to another task. At any rate an engineer, analyst, or designer just does not stop or start work on a task instantaneously; when transfering from one task to another - as required in the batch mode of operation - there is additional time needed to "ramp up" (get back up to speed) on the next job, job step, or task. The saving or elimination of this kind of time can be a twofold advantage;

not only is the time itself a quantitative factor but the human creativity can be enhanced and improved by the timely reinforcement available through the immediate interactive feedback of results in response to problem input. The conversational mode of problem solving is a powerful way to obtain better (e.g., high quality, longer life, lower manufacturing cost, simpler construction, lower weight, less maintenance) solutions to design problems.

The key to uninterrupted problem-solving is turnaround or response time. The term "turnaround time" is ambiguous in that it has been applied to all four of the explicit time spans shown in the diagram below, and to many other ill-defined time spans as well.



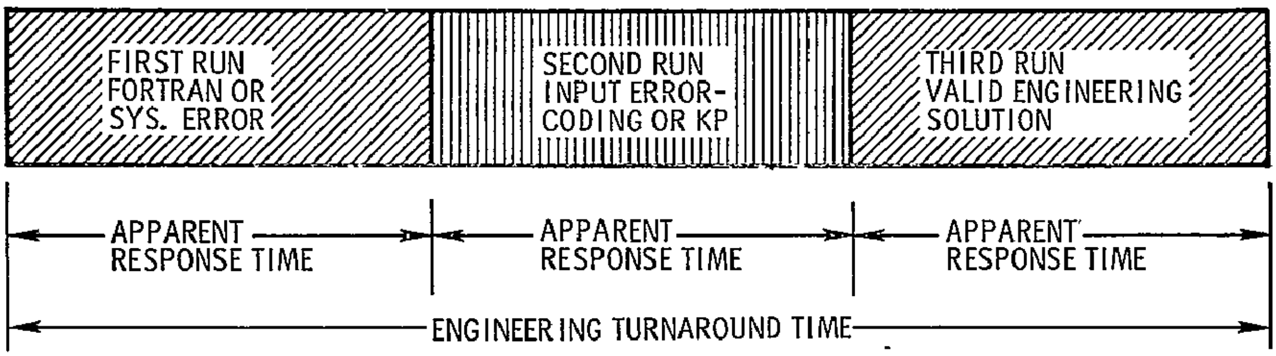
Engineering turnaround time involves the real response time plus the following additional delays:

1. Data translation from engineering symbols, names, and units to the form required by the computer program; this is a task which precedes encoding of data specifically for the program (OM).

2. Output must be valid engineering results, i.e., not data coding errors, operating system/program, or machine errors. Note that, for example, an unstable solution because of a control system gain parameter is a

valid engineering answer; it may not be an acceptable solution but it provides the engineer with insight on how to modify the problem parameters for the next run.

Included in the output consideration is the time to determine the significance of the output. (A stack of printed output may take much longer to be assimilated than a plot or graphical presentation of the output.) Until an engineer gets valid results from his program he cannot make engineering progress on that phase of his work. Therefore, engineering turnaround is measured from the time the data is ready for coding until valid engineering results are obtained. If a program were submitted with one program source language coding error and one data coding input error, the turnaround time might look like that diagrammed below. As can be seen, the real response per valid run is often a multiplicative function of the real response time to get a single



run through the computer (previous diagram). The iterative nature of the design process introduces yet another multiplicative factor resulting from many of these engineering turnaround times caused by the interdisciplinary interactions in the design process. This interdisciplinary involvement superimposed on the design process creates more problems in communication and time delays. Careful and timely coordination is required to meet any predetermined schedule or even to prepare a realistic schedule; unpredictable turnaround time makes this planning difficult. The use of interactive computing not only reduces turnaround time but makes it more predictable and improves communication when common data and computer program interfaces are used. It can be seen that a considerable amount of lost calendar time and engineering manhours are involved in turnaround time in the design process and may be available for reduction.

The desire to reduce the manhours spent and calendar time has resulted in a partnership between the designer and the computer in which each partner contributes his best skills to the design process. This partnership was described by Kopp and Wertz of Battelle Memorial Institute as (Reference 7), "The designer is responsible for the thought and heuristic aspects; the computer is used for the most effective and efficient handling of the computational and algorithmic aspects of the design process."

The discussed advantages of interactive systems can be qualitatively summarized as follows:

1. Provides user control of OM execution:
   a. Input/output validation:
      - Data entry/keypunch errors.
      - Input errors producing erronous results.
   b. Monitoring the solution process.
   c. Interruption of the solution process.
   d. Output selection and review.
2. Increases insight into OM solution:
   a. Observe effects of:
      - Input variations.
      - Solution process modifications.
   b. Immediate information in response to parametric studies.
3. Stimulates creativity by maintaining designer's "mental momentum".
4. Results in better selection of design alternatives and quicker elimination of "dead end" approaches.

Taken all together, the proper use of interactive computing techniques can give the user better visibility of and a superior ability to control any or all of the design process resulting in higher productivity for the individual.

Experience with conventional interactive communications systems at Convair is summarized in Figure 2-14. These systems typically use the interactive communications subsystem of the host operating system (Convair/SDO retrofit CDC's INTERCOM 2.0 into SCOPE 3.2) and electromechanical typewriter terminals (Convair/SDO used NCR 260's) or CRT display, keyboard input terminals (Convair/SDO used a TEKTRON 4010) acoustically coupled to the computer via the inplant telephone system. Experien with this type of system has demonstrated the applicability of typewriter-type interactive terminals for those tasks listed in the figure. The time and cost savings increase with the number of separate job-steps; this is particularly true if the results of the pr ceeding job-step must be evaluated before the next step is initiated.

2.2.3.3 The role of interactive graphics: Although interactive graphics is actually a part of interactive computing, there are several considerations which make its contribution a major step forward in interactive computing.

	TIME SAVINGS	COST SAVINGS
• PROGRAM DESIGN & DEBUGGING		
SMALL PROGRAMS & SUBROUTINES	10 TO 1	4 TO 1
LARGE PROGRAMS DEVELOPED IN SEGMENTS	4 TO 1	3 TO 1
• DATA BASE		
BUILD, EDIT & MANAGE	10 TO 1	10 TO 1
INTERROGATE/MODIFY	20 TO 1	3 TO 1
• SYSTEM PROGRAMMING		
DEVELOP & EXECUTE REQUIRED MODEL FROM SUBROUTINE & SUBMODULES	20 TO 1	5 TO 1

Figure 2-14. Typical Teleprocessing Mode Vs. Batch Mode Savings (Convair Experience)

Consider the design process itself:

1. Nature of the design process:
   a. Designers are not computer oriented; they use computers as tools to produce designs.

   b. Geometrical models are used by designers; numerical models are used by computers.

   c. End results of design processes are three-dimensional (physical) products; final computer results must usually be transformed and presented in geometrical or graphical form.

2. Graphics allows both designer and computer to perform effectively:
   a. Designers retain the geometric model while the computer manipulates the numerical model.

   b. Validity of computer's numerical model can best be demonstrated graphically.

   c. Geometrical properties of numerical results can be displayed and readily understood.

d. Graphical presentation of functional behavior is easily accomplished.

e. Evaluation of numerical results is greatly speeded and enhanced.

Figure 2-15 is a readily apparent example of benefits to be derived through interactive graphics.

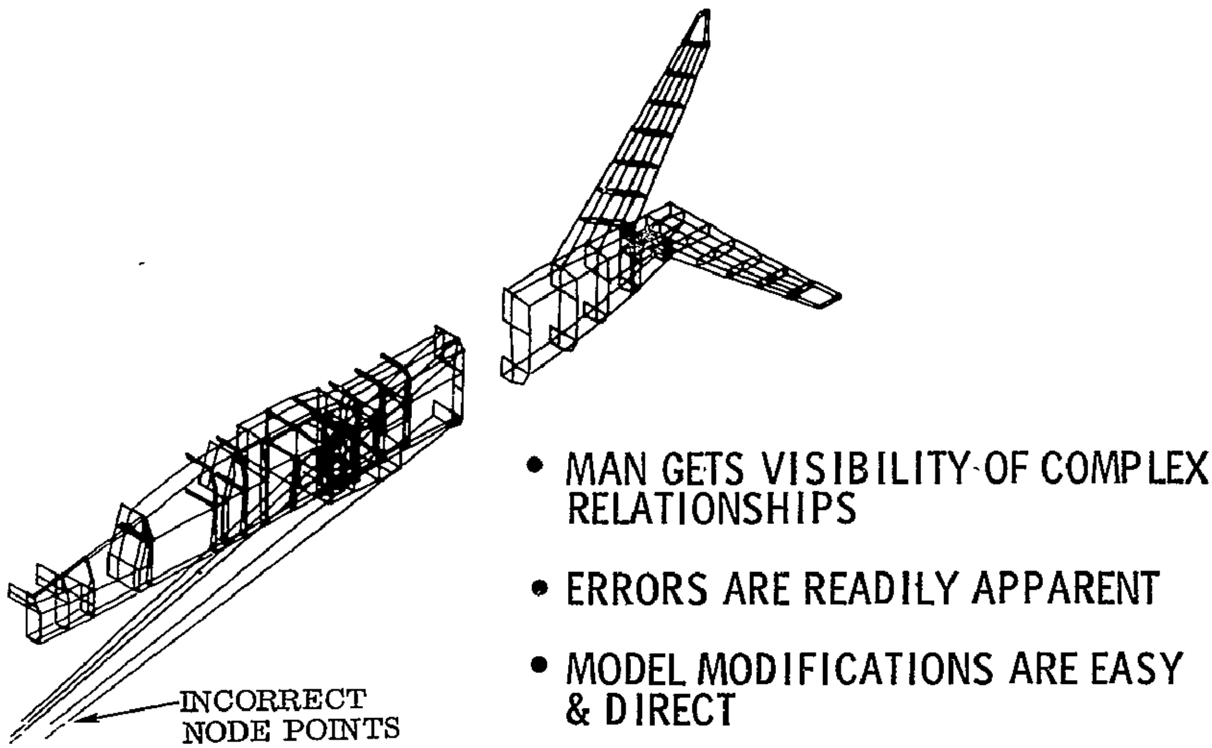


Figure 2-15. Example of Some Benefits of Interactive Graphics

Consider the problems related to the data base:

1. The data base contains multidimensional items:
   a. Model representations, e.g., 3D bar-panel structure.
   b. Descriptions of data in functional form e.g., Weight per inch vs Station.
   c. Data tables of values, e.g., stability derivatives as a function of Mach number, incidence angles, etc.
2. Retrieval of numerical values from data base is not enough; these are often difficult to interpret when obtained. Graphics provides the medium in which designer is most comfortable:

a. Geometric meaning can be given to numerical values in the data base.

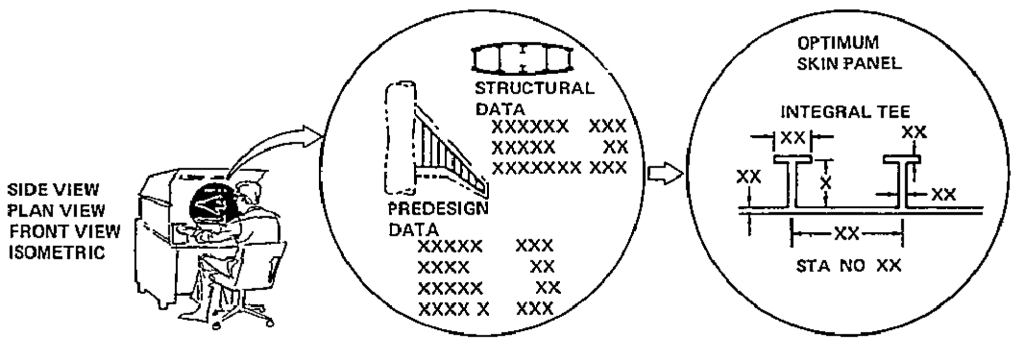
b. Geometric models can be viewed in a variety of orientations, enhancing interpretation.

c. Functional data forms are easily plotted and interpreted.

d. Data tables can be compactly presented as graphs.

Without graphics, retrieval of information from data base is severely hampered and the designer is forced to operate without his usual geometric aids.

Figure 2-16 summarizes the features of some existing interactive graphics system together with some industry examples.



- FEATURES
  - DIRECT TRANSFER TO DETAIL DRAWINGS
  - VISUAL CHECK FOR COMPONENT PLACEMENT
  - STRUCTURAL PACKAGING WITHIN A CONFIGURATION ENVELOPE
  - CUTTER PATH CHECK FOR N/C MACHINING
  - INDUSTRY EXAMPLES
    - LOCKHEED-BURBANK REPORTS 6 TO 1 TIME SAVINGS ON FLOOR BEAM PRODUCTION DRAWINGS BY GRAPHICS & $500,000 SAVINGS IN DEVELOPMENT OF 1,300 WIRING DIAGRAMS VIA GRAPHICS FOR L-1011
    - MOTOROLA, LOCKHEED, IBM & OTHERS USE GRAPHICS FOR PRINTED CIRCUITBOARD DESIGN & PACKAGING GD ELECTRONICS REPORTS SAVINGS OF 3 TO 1 IN COST & 10 TO 1 IN TIME
    - McDONNELL DOUGLAS USES INTERACTIVE GRAPHICS TERMINALS WITH DIMENSIONAL DATA BASE FOR DETAIL ENGINEERING DRAWINGS COMPANY REPORTS TIME SAVINGS OF BETWEEN 5 TO 1 & 15 TO 1 DEPENDING ON COMPLEXITY OF THE DRAWING
    - SIKORSKY DIVISION OF UNITED AIRCRAFT REPORTS USE OF INTERACTIVE GRAPHICS FOR GEAR BOX DESIGN & INTERFERENCE PROBLEMS THERE WERE NO QUANTITATIVE REPORTS ON SAVINGS

Figure 2-16. Graphics Compatibility with Existing Techniques

Convair's experience with the CDC 274 Digagraphics (large screen) terminal is summarized in Figure 2-17. It is worth noting that none of these applications could be accomplished on non-graphic terminals.

PRODUCT DEVELOPMENT TIME (CALENDAR):	UP TO 20:1
MANHOUR SAVINGS:	UP TO 10:1
COMPUTER COST SAVINGS:	UP TO 10:1
NONQUANTIFIABLE	MET SCHEDULES PREVIOUSLY TOO SHORT

Figure 2-17. Interactive Graphics Experience at Convair Aerospace

Industry - particularily the aerospace industry - has made very large investments in interactive graphics application (Figure 2-18) attesting to the economic advantages of interactive graphics as an engineering tool. The reader is referred to Appendix B for details and examples substantiating Figure 2-18.

○ CURRENTLY IN USE/DEVELOPMENT IN INDUSTRY ● CURRENTLY IN USE/DEVELOPMENT AT GENERAL DYNAMICS

	AEROSPACE	SHIPBUILDING	AUTOMOTIVE	ELECTRONICS	CIVIL ENGR	BUILDING & CONSTRUCTION	MILITARY
SURFACE DEFINITION	○● LOFTING MASTER DIMENSIONING	○ LINES FAIRING HULL DESIGN	○ BODY STYLING DIE DESIGN	ANTENNA BEAM PATTERNS	○ TOPOGRAPHY CONTOUR MAPPING	○ SHELLS	BLAST EFFECT ENVELOPES
STRUCTURES ANALYSIS	○● FINITE ELEMENT ANALYSIS FRAME ANALYSIS	○ STRUCTURAL DETAILING	○ SUSPENSION SYSTEM DESIGN	ANTENNA STRUCTURES	STRUDEL STRESS	MEMBER SELECTION FRAME DESIGN	○ ROTOR ANALYSIS
CONFIGURATION LAYOUT	○● PACKAGING INTERFERENCE ANALYSIS WT & BALANCE	○ ARRANGEMENTS SPACE RESERVATION PIPING CABLING WIRING ROUTING		○● PC BOARD LAYOUT IC CHIP DESIGN	ROADS ALIGNMENT SHIPYARD CRANE TRACKS	○ OFFICE LAYOUT PIPING DESIGN	
DRAFTING & N/C	○● PART DESIGN WIRING ENGR DWG MAINTENANCE PART PROG	○ STRUCTURAL DETAILING FLAME CUTTER PROGRAM	○ PART DESIGN TOOL & DIE DESIGN	● BACK PLANE WIRING PC BOARD ASSEMBLY FROM TO LISTS	○ HIGHWAY PROFILE CROSS SECTIONS PLAT LAYOUT		
MECHANISMS	○● LINKAGES FOR DOORS LANDING GEAR ETC KINEMATICS	PROPULSION PIPE SIZING PIPE STRESS	○ WINDSHIELD WIPERS DESIGN	ANTENNA VIBRATION RF INTERFERENCE		HEATING VENTILATION, AIR CONDITIONING SYSTEMS	RECOIL ANALYSIS
PRELIMINARY DESIGN	○● A/C PERFORMANCE AIRCRAFT DESIGN SPACE VEHICLES WING FLAP DESIGN	ISDS COGAP		LOGICAL DESIGN		○ ARCHITECTURAL LAYOUT PROSPECTIVE VIEW GENERATION	WEAPON SYS ANALYSIS
ENGINEERING ANALYSIS	○● CONTROL SYSTEMS ANALYSIS DATA REDUCTION OPTIMIZATION CURVE FITTING TRAJECTORY ANLY AERODYNAMIC ANLY	○ HULL CHARACTERISTICS HEAT BALANCE	○ SAFETY FACTOR PLANNING STRESS ANALYSIS	○● CIRCUIT ANLY ECAP ETC			○ DAMAGE ANALYSIS DETONATOR ANALYSIS STRUCTURAL ANALYSIS
MANAGEMENT	○● DATA BASE INQUIRY FORWARD PRICING PROPOSAL MONITORING EDITING & PREPARATION			LEARNING CURVE ANALYSIS FUNCTIONAL ANALYSIS & FLOW DIAGRAMMING PERT CHARTING			
PROGRAMMING	○● ON LINE DEBUGGING ON LINE PROGRAMMING			TEXT EDITING INSTRUCTION			

Figure 2-18. Interactive Computer Graphics Applications

2.2.3.4 Tutorial aids and recovery: The interactive portion of IPAD should forgive user errors at the interactive terminal; if the user makes a mistake in selection or input, the system should not abort his job but provide messages and an opportunity to recover from the error and proceed. In particular, a tutorial capability must exist which permits the user to learn as he uses the system.

There are two components of tutorial aides:

1. Tutorial data
2. Executable code to display the data.

There is no explicit requirement to insert tutorial capabilities within an installations OMs, but all IPAD system and utility software should contain executable code providing a tutorial capability.

Tutorial capability applicable to OMs (but not necessarily within the OMs) could be provided separately, e.g. by a general display utility which displays data provided at OM incorporation by the cognizant programmer. The form and content of this data could be determined by the needs of the user group and provided in a data file for access by the utility. Typically such data might include:

1. A description of the capability provided by the OM.
2. A description (including definition, units and coordinate systems) of inputs required for and outputs provided by the OM.
3. Explanation of any diagnostic messages which the OM generates.
4. Suggested user procedures for any known abort conditions.

The same general utility could also be used to provide tutorials for IPAD utilities. Beyond this there are two additional requirements of utility software which are not imposed on OMs:

1. The command language for a given utility must include a request for tutorial assistance (e.g. "HELP"). The utilities response to this request might be some display indicating options logically available to the user in view of previous action.
2. User supplied input must be capable of being edited and opportunities provided to:
   a. Correct the input.
   b. Review material equivalent to a conventional users' guide.
   c. Issue the "HELP" command.

2.2.3.5 The role of standards: IPAD must make provisions for certain engineering standards (e.g. , coordinate systems, units, etc.) but should not pre-suppose nor require these. (Note that strict standards can throttle the user's creativity; they are useful principally as a means of inter-human communication and have limited use within IPAD.) IPAD shall delegate to the user the standard (e.g., coordinate) he prefers to use. However, the user should be aware of designated standards that will reduce system overhead.

In general, it is best to let the user continue to work in his own (individualistic) mode of operation, i.e., use his own coordinate systems, his own units, his own formulas and methods of approach. Man does his best creative thinking when operating in an environment free from distractions and where he feels comfortable (confident) In creating, he associates facts and ideas residing in his own memory; if these facts and ideas must be recast into some new mold, they usually become strangers. His tools are familiar items e.g., he knows their limitations; if some are taken away from him and others are replaced by new tools, he begins to lose confidence and must start a relearning cycle. It must be understood that within his present mode of operation most of the surprises have been encountered and resolved; the user can presently work with familiar patterns and can focus on the problem at hand without being irritated or distracted by little things going wrong.

On the other hand, there has to be some standardization within the IPAD system to keep its size within reasonable bounds and to offer direction to users not having a preference. (There isn't infinite storage capacity available to handle the special needs of every engineer.) The compromise solution for introducing engineering standards within IPAD must strive for making any standardization transparent to the engineer. The user will not use IPAD unless it is easy for him; he absolutely will not use IPAD if he is forced to create in a way which is foreign. Note however, that the user is usually willing to learn almost anything if he can delegate the drudgery work to the computer while retaining the ability to create within his own regimen.

The four areas in which standardization is envisioned to benefit the engineer are data base organization, vehicle and subsystem design/drafting, systems of units, and coordinate-system transformations. Each of these subjects is briefly treated in the paragraphs that follow.

<u>Data base organization and assembling.</u> - Project teams within engineering and scientific communities use many different terminologies, symbols, and glossaries as the predominant means of communication. These terminologies and glossaries are often used as mnemonic aids in naming variables for computer programs; this practice is expected to continue in the naming and defining of data base quantities within IPAD. To adopt a stereotyped set of variable names and glossaries for IPAD's data base is neither required nor warranted. The choice of size, structure, contents, variable

names, management, and security of the data base depends upon the user who selects those which best fit his needs. IPAD must assist the engineering/scientific communities by providing freedom and flexibility for organizing and assembling the data base. This flexibility shall not prevent a standardized data base - with specific size, contents, and glossary - from being used in common by interested parties.

Vehicle and subsystem design/drafting. - Imbedded in the creative aspects of the design process are a multitude of mechanical, repetitive tasks which culminate in the visualization of ideas and products by sketches, drawings, perspectives, plots, etc. Many of these mechanical tasks can and should be standardized. The tasks range from the drawing of a line or circle, to the repetitive drawing of parts, to the complex composition of a vehicle's inboard profile. The more complex tasks are usually broken down into many simpler ones which assemble into the total. Standardization is needed for the basic drafting operations (such as routines for line and curve generation). Standardizations is also needed for data banks containing standard parts for libraries of normalized dimensional data (scalable), and for components/subcomponents (such as landing gears of various types, brakes, wheels, tires, etc.). Some of the activities ancillary to the design process itself, such as applicability of specifications, tolerances, computing mass properties, etc., should also be standardized. IPAD should accommodate all applicable standards of the design/drafting activities.

Systems of units. - Although the use of the English system of units is still predominant in English-speaking countries, the use of the International System of Units has been increasing in the past several years. The reporting of study results in both systems has been on many occasions a contractual requirement which has typically led to performing analyses and evaluations using the English system and then converting the final result to the International system. Although most of the scientific, engineering, and manufacturing communities may continue to operate with the English system there will be an increasing need to operate with both. This duality must be dealt with for some years and IPAD must make provisions to supply standardized conversion capability to relieve the users from the tedium of making these conversions. Furthermore, IPAD should contemplate the use of data and OMs in a mixed mode, and have the option to produce final results in either system.

Coordinate systems and transformations. - Several reference coordinate systems are typically used within a project-oriented team involving different engineering/scientific disciplines. Some of these disciplines have adopted one or more coordinate systems as standard for their studies. In order to preserve the disciplines' modus operandi and also to provide for interdisciplinary exchange of data and freedom of operation within a team, the IPAD system should standardize a number of specific coordinate system transformations. These transformations should be transparent to the user. A general purpose utility consisting of a set of transformation matrices, with appropriate tutorial aids, could provide this standardization so the engineer would not be required to transform data external to the system. He should be able to use the data directly in its available form and call the standard transformation utility at the needed points to make the coordinate changes. By using the same standard utility, the user can present his results expressed in other coordinate systems as required.

2.3 IPAD, A Conceptual Design

A conceptual design of IPAD arose as an outgrowth of the aforesaid considerations and an attempt to facilitate the utilization of present (and envisioned) computing systems by potential users of these systems. This section presents that design as it evolved.

2.3.1 Overview and operating philosophy. - Figure 2-19 illustrates an overview of the IPAD conceptual design. This figure is - as the IPAD system it depicts - centered around the interactive graphics terminal. The data base is envisioned as a collection of subbases (shown in the dotted box at bottom of figure) consisting of:

1. Engineering Review Board (ERB) Action File, a communications file summarizing the action requests placed on the various engineering disciplines by the ERB as the principal representative of engineering management.

2. Task Status File, a communications file summarizing the current status of action requests placed on the various engineering disciplines by the ERB. The entries in the Task Status File are correlated with entries (action requests) in the ERB Action File and are linked to these.

3. Multidisciplinary Data Bank (MDB), that portion of the IPAD data base reserved for project approved data. This data bank is under the supervision and control of the Data Base Administrator (see figure) who ensures that the project's data is reviewed and approved before being inserted into the MDB. This is intended to prevent an "epidemic" resulting from erroneous data invalidating a sequence of studies drawing upon this data, and the subsequent "chain reaction" of erroneous data being fed back into the MDB.

4. MDB Data Update File, the input queue for data to be inserted into the MDB. The Data Base Administrator reviews and approves this data before actual insertion. An illustration of this process is presented in Figure 2-20.

5. Utility Library, the collection of IPAD system code support IPAD users in general. Utilities are to be distinguished from specific OMs supporting individual users.

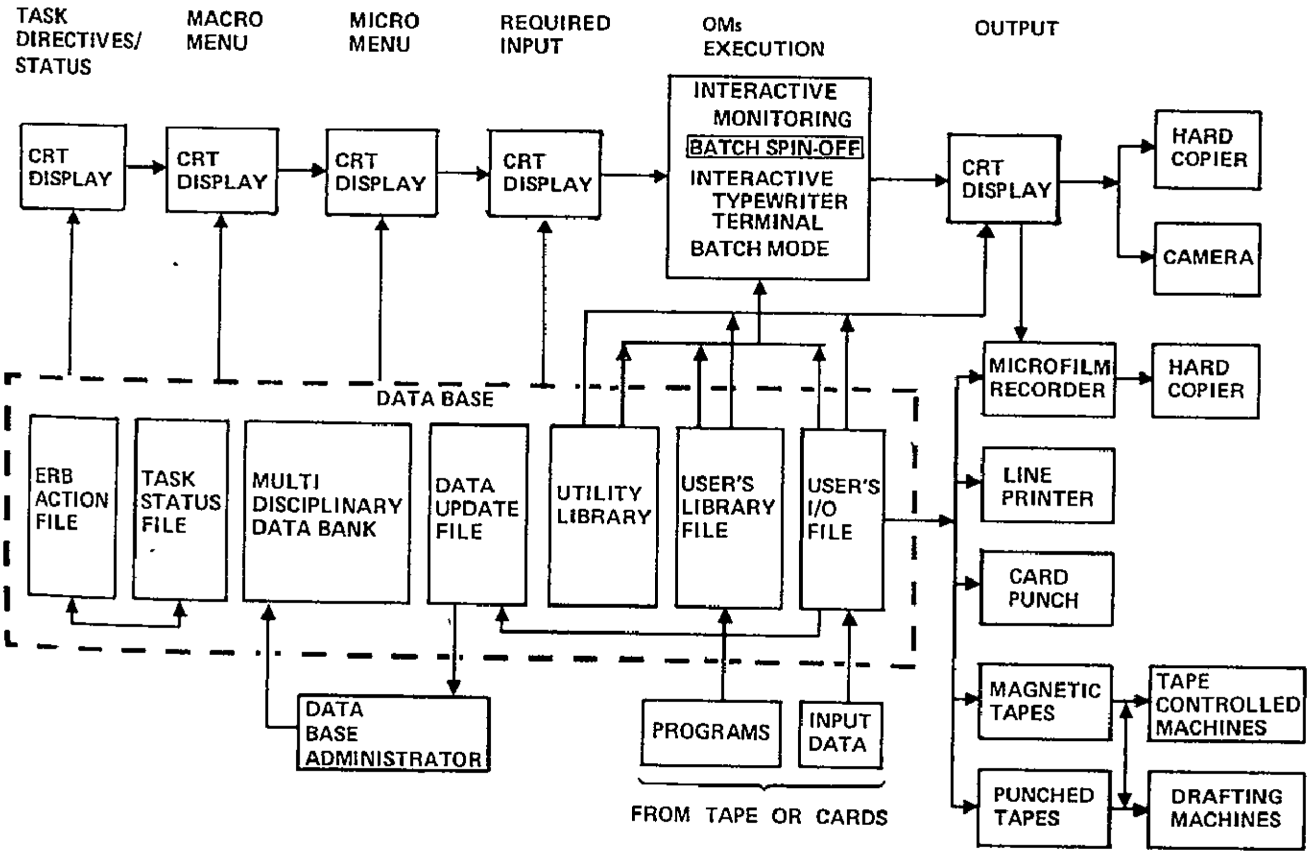


Figure 2-19. IPAD System Operation, Batch Spin-off Mode

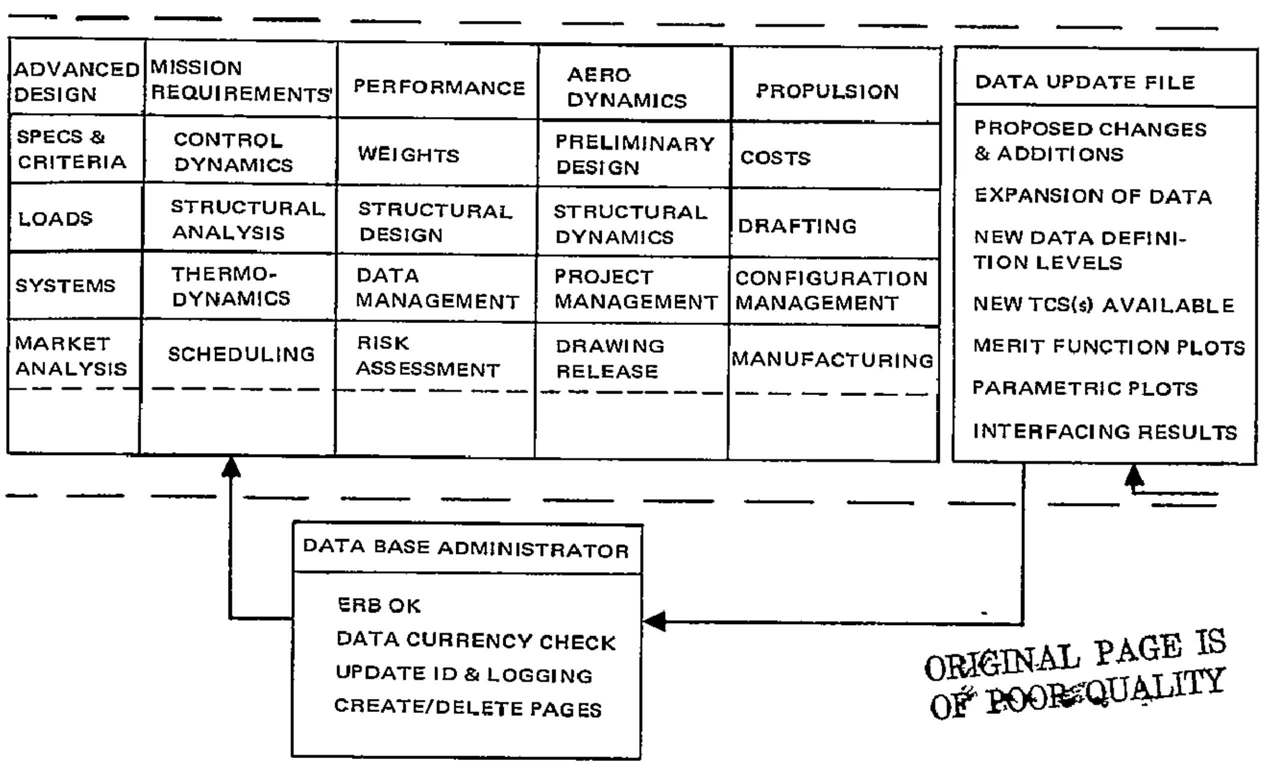


Figure 2-20. Updating the Multidisciplinary Data Bank (MDB)

6. User's Library File, the collection of OMs and special utilities (code) supporting the individual users. It also can contain special data supporting several users. It is envisioned that this file's organization will be at the discipline level (lumping the OMs or special data utilized by users within a given engineering discipline). Additional infrequently-used programs (OMs) may only temporarily reside in the User's Library File, being read into the system (when needed) from cards or magnetic tape.

7. User's Input/Output (I/O) File, that collection (for a given user) of intermediate results, partially constructed inputs, partially processed outputs, and related data that the user requires for the purpose of conducting his task. The User's I/O File can be considered as the user's "scratch" area in the data base. Infrequently-used data may also come from (or go to) cards or tape and only temporarily reside in the User's I/O File.

The last three files are illustrated in more detail in Figure 2-21.*

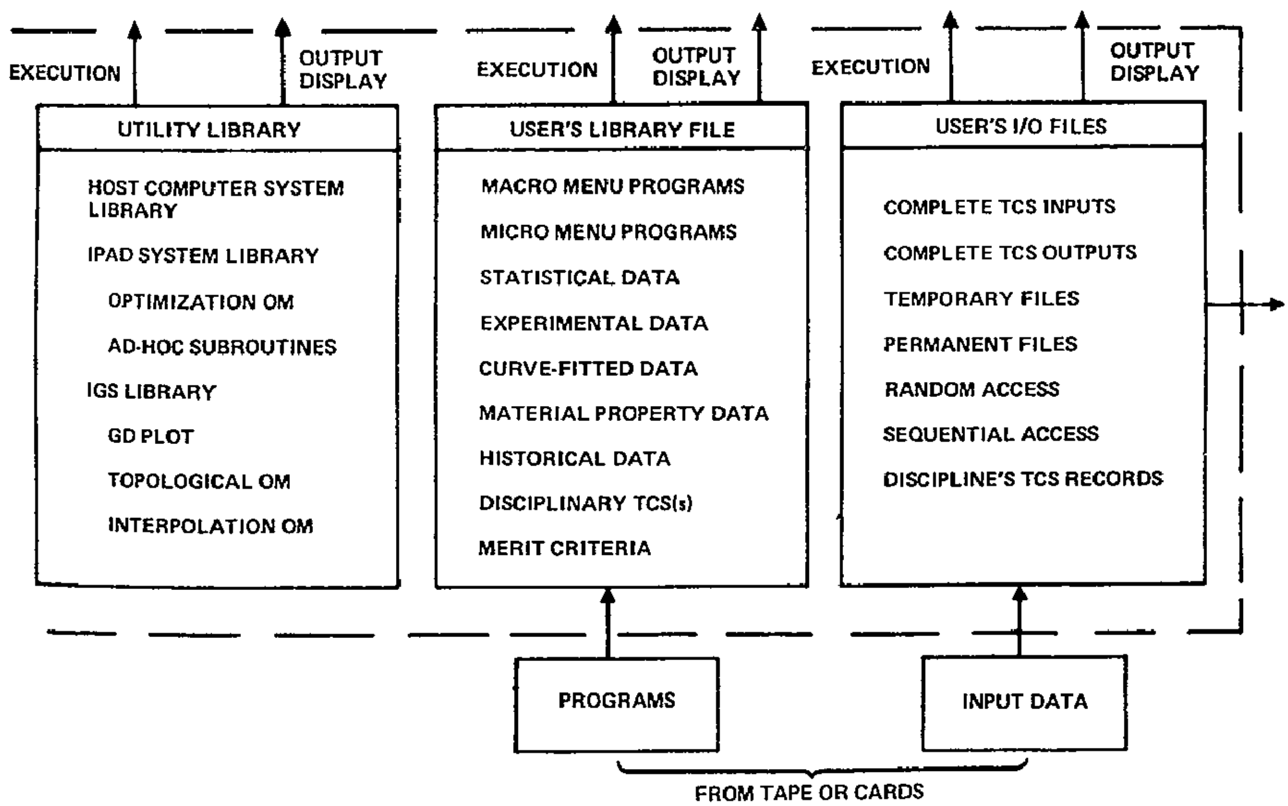


Figure 2-21. Users' Utility, Library and I/O Files (Typical)

---

* The reader is deferred to Subsection 2.3.3 for an explanation of TCSs.

As envisioned, the user would "sign on" the IPAD system and (Figure 2-19):

1. Examine any new task directives in the ERB Action File and the current task status summarized in the Task Directives/Status File. This process is personalized to the user or his disciplinary area (through his sign-on identification) in order to eliminate unrelated information.

2. Select a task to be worked on during the current interactive session. The user acquires an individual OM through the Macro/Micro Menu selection process. The Macro and Micro Menus are display data supporting the OMs (and IPAD utilities) contained in the User's Library File. (These menus will be discussed later on.) The form of the menu is a logic tree (or more complex) structure which stepwise refines the selection process until the actual OM to be used has been selected.

3. The required input for the selected OM is then examined from display data (IDEF, see Subsection 2.3.4) stored with that OM in the User's Library File. From that user-oriented display information, the user directs the system to configure the required input data for that OM, selecting data from the Multidisciplinary Data Bank (MDB), the User's Library File and/or his User's I/O File. The selection process for input data is similar to that for OMs.

4. The user then selects the type of OM execution desired (Figure 2-19):

   a. Interactive monitoring if the OM selected is an interactive OM with large amounts (or graphical) output to be monitored (either requires use of CRT terminals).

   b. Batch spin-off, which is a request for immediate (high-priority) batch processing while the user performs other, perhaps related, tasks.

   c. Interactive typewriter mode if the OM selected is an interactive OM with minimal I/O data requirements.

   d. Batch mode, which is a request for deferred batch processing (typically overnight).

5. Following OM execution (Figure 2-19 illustrates the case of batch spinoff) the resulting output from the OM is examined via display data (ODEF, see Subsection 2.3.4) stored with that OM in the User's Library Files. From that user-oriented display information, the user directs the system to configure and present the required display - usually for viewing if on a CRT terminal - and perhaps recording on any of the devices indicated in the figure. Figure 2-22 illustrates currently available devices in more detail.

Having completed the output of that OM (or perhaps leaving a partially completed task in his User's I/O File), the user has the option of signing off or returning to the Macro or Micro Menus for another OM or returning to the Task Action or Status Files for another task. Prior to signoff, however, it is envisioned that the user will be given (by the IPAD system) the Task Status File to update, based on the tasks (subtasks) performed.

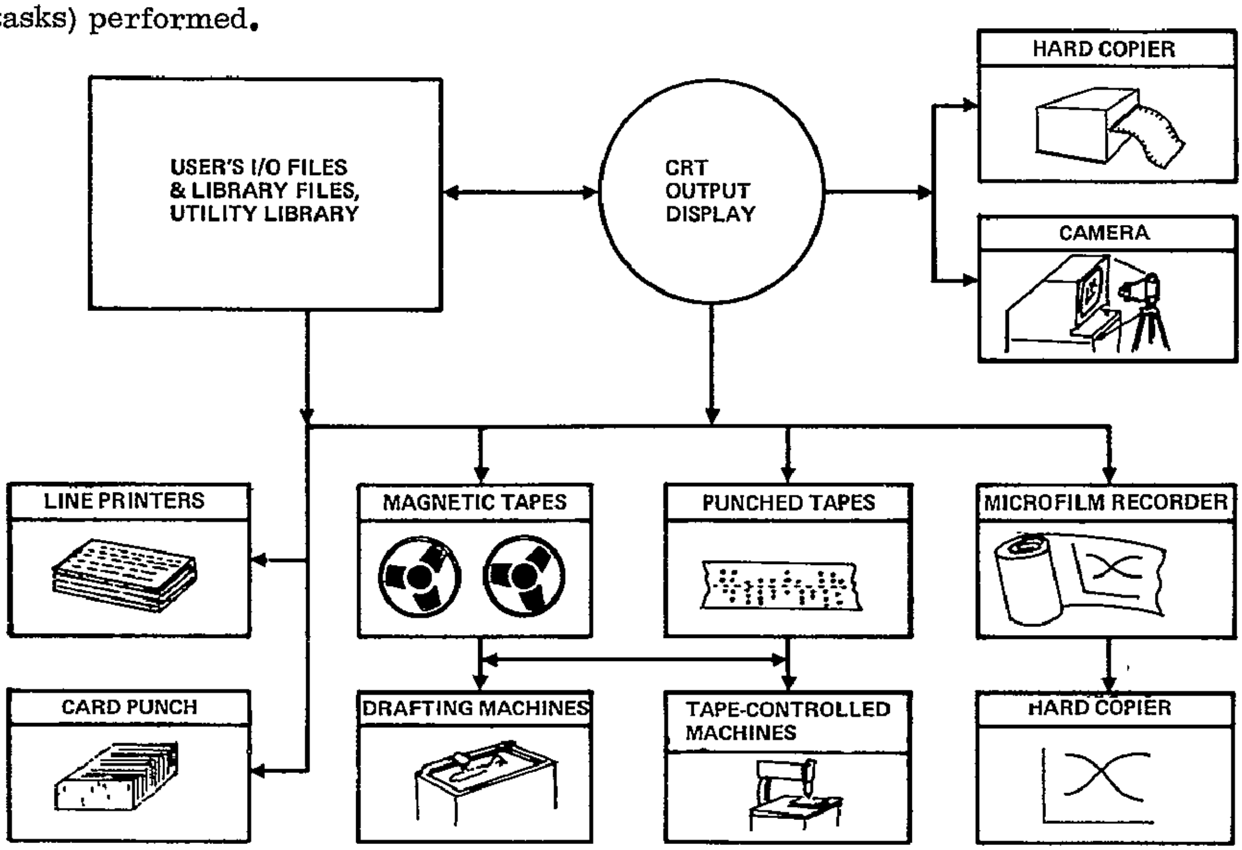


Figure 2-22. Available System Output Options

An example of the displays that a Structural Engineer might see as he conducts a wing definition study is presented in Figures 2-23 through 2-26. (Note that these figures represent details of corresponding blocks in Figure 2-19). In response to an ERB request for "Proposed Wing Changes", the related tasks in his Task Status File are displayed and the user selects "Wing Definition Study" (Figure 2-23). (Note that the asterisked tasks in the figure are those completed and await ERB review before being removed from the Task Status File.) In response to the "Wing Definition Study" selection, a Macro Menu is presented to the user to select a category in which an OM (which he is envisioning using) resides (Figure 2-24); he selects the category "Wing." In response to the selection of Wing, the user is presented a Micro Menu of the cross-reference category type (Figure 2-25); here he selects "Multiple Station," "Synthesis" and "2nd Level" resulting in the selection of a specific interactive OM to be used. With that OM comes a summary of the required input data with which the

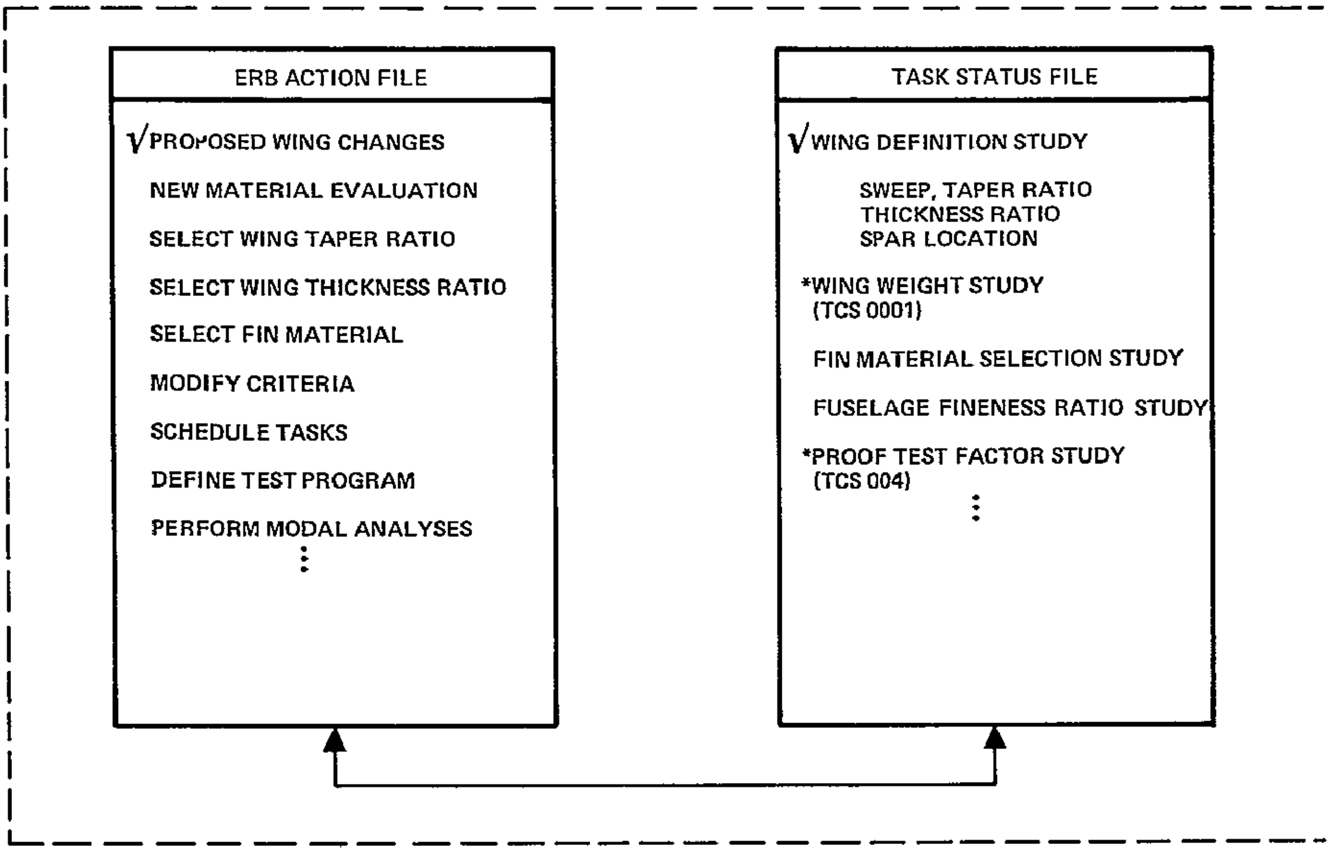


Figure 2-23. ERB Action File and Task Status File (Structures)

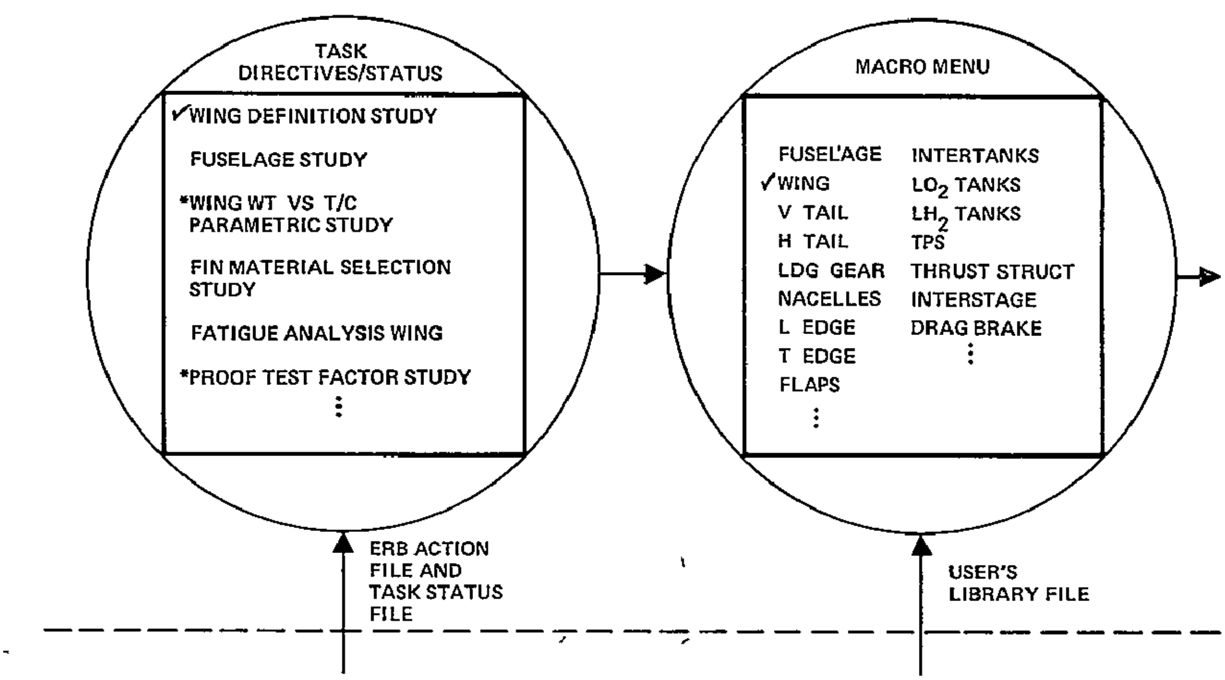


Figure 2-24. Task Directives/Status and Structures Discipline's Macro Menu

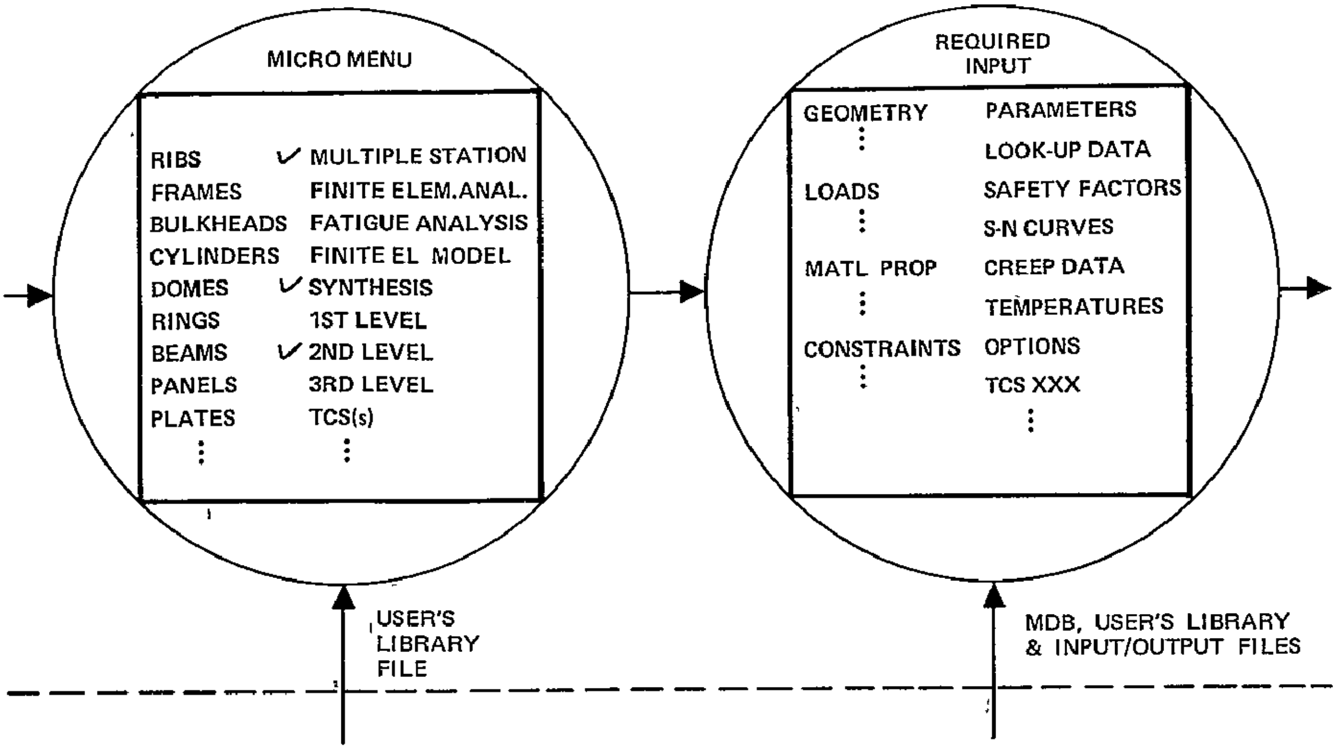


Figure 2-25. Structures Micro Menus and Required Inputs

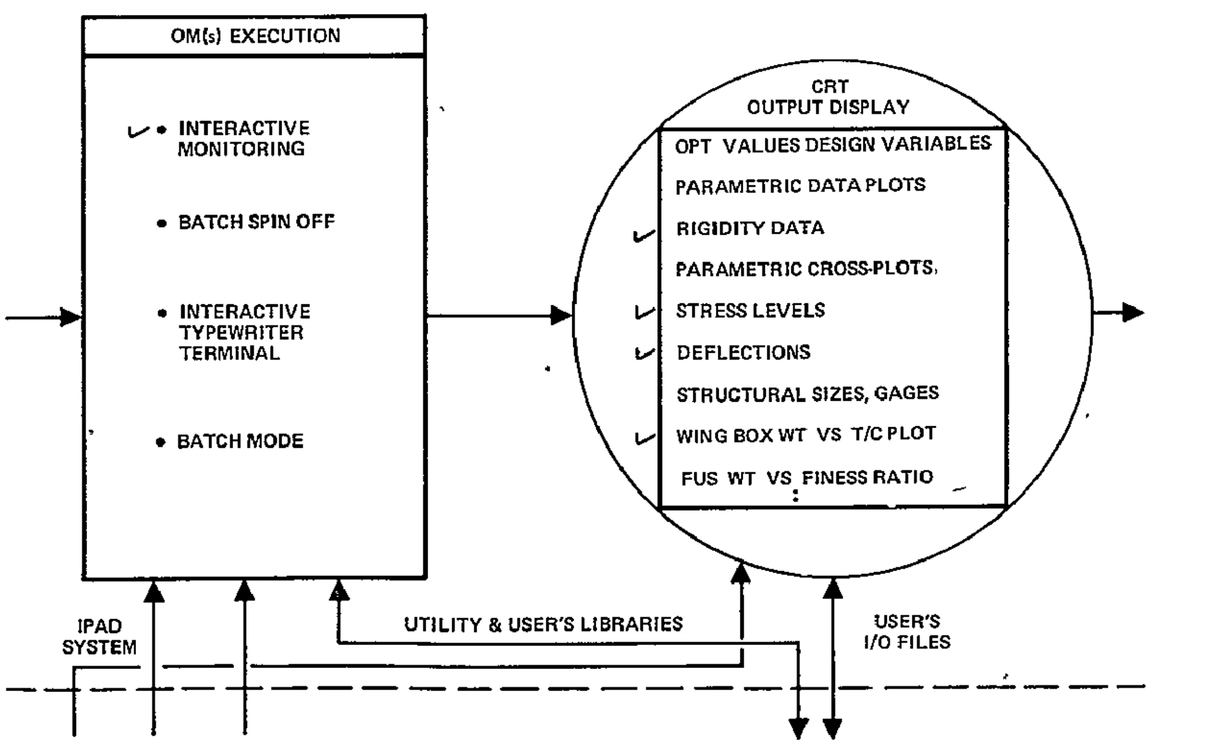


Figure 2-26. Disciplinary OM Execution and Structures Output Menu

user interacts in constructing an input file for that OM (Figure 2-25). Selecting "Interactive Monitoring" (Figure 2-26), the user is presented a menu of available output parameters to view while monitoring and selects "Rigidity Data", "Stress Levels", "Deflections" and "Wing Box Weight vs T/C. Plots" for graphical presentation. The user then sets up the plot grids and arrangements, and calls for OM execution which is monitored interactively.

Figure 2-27 presents a near identical operation using interactive typewriter type terminals. The hardcopy in this case is the typed sheet produced (compare with Figure 2-19). Figure 2-28 presents the batch mode for which no interactive display is required; note however that the hardcopy capabilities (summarized in Figure 2-22), except for those available only through the interactive terminal, are still available (through IPAD) in a batch mode (providing these have been requested).

The remainder of this section presents the details of this conceptual design of IPAD.

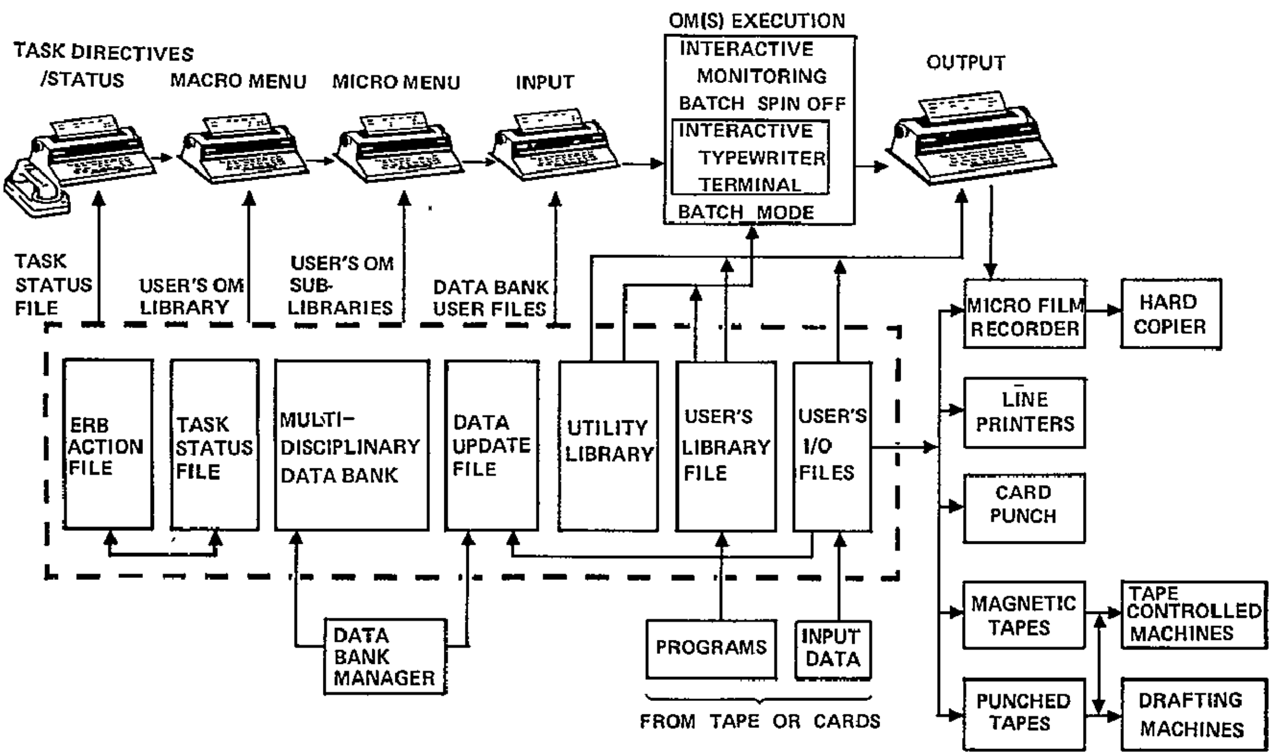


Figure 2-27. IPAD System Operation, Typewriter Mode

2.3.2 System baseline. - The IPAD system design approach assumed for this conceptual design is the "sub-subsystem" approach discussed in Subsection 2.2.2.4 (Figure 2-11) and is illustrated in Figure 2-29 in terms of CDC hardware/software.

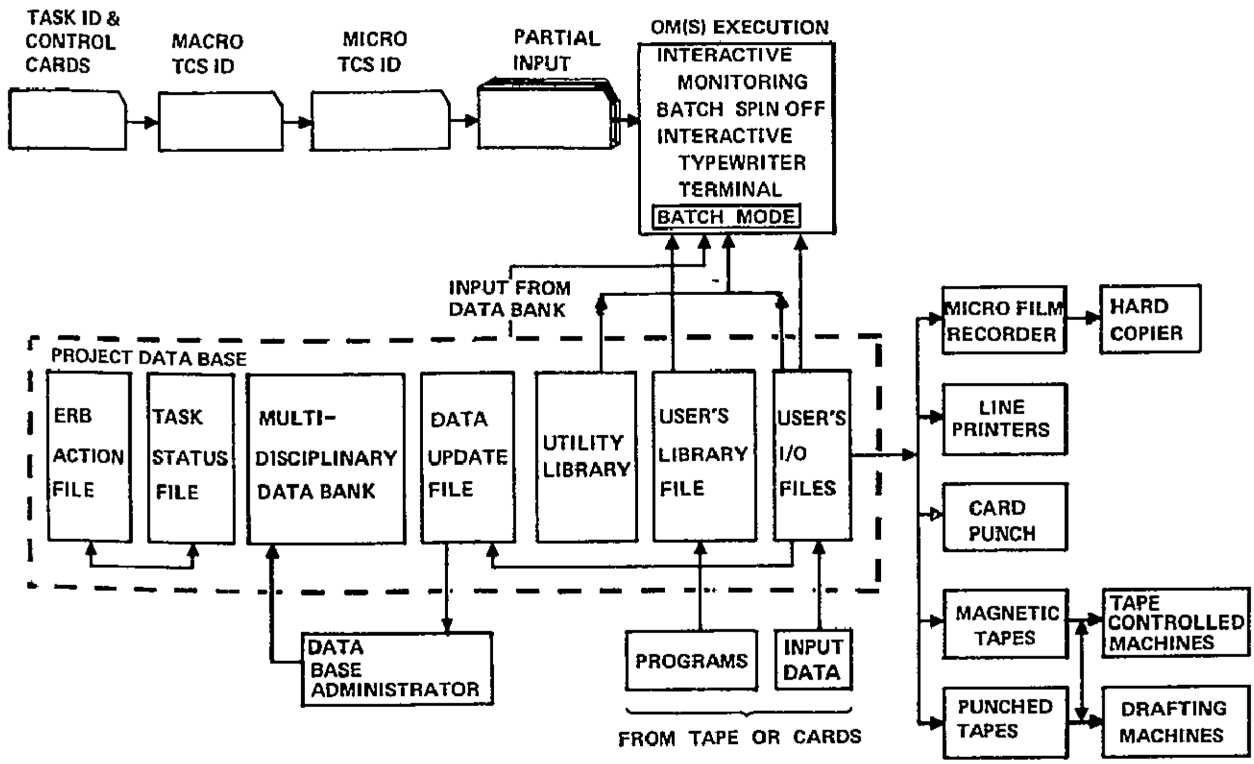


Figure 2-28. IPAD System Operation, Batch Mode

The use of the CDC CYBER 70 system as a baseline system to evolve the IPAD conceptual design arose through intimate familarity with CDC hardware/software rather than an intent to restrict IPAD to CDC systems in general. It is felt that - at this level of analysis - CDC's system architecture will serve to typify large scale scientific computer system of the type required by IPAD (Section 2.2) and hence test the viability of the conceptual design.

The CDC CYBER 70 series hardware was assumed as the host computer with the SCOPE 3.4 operating system software including the INTERCOM 4.1 interactive communications subsystem. IPAD is then envisioned as a subsystem to INTERCOM 4.1 (principally) supporting a variety of interactive communication devices such as:

1. Texas Instrument's TI-725 portable "typewriter" terminal.
2. National Cash Register's NCR C-260 small "typewriter" terminal.
3. Tektronix's 4010 alphanumeric/graphics CRT terminal with keyboard input.
4. CDC's 274 large screen graphics CRT terminal buffer-controlled by the CDC 1700 minicomputer.

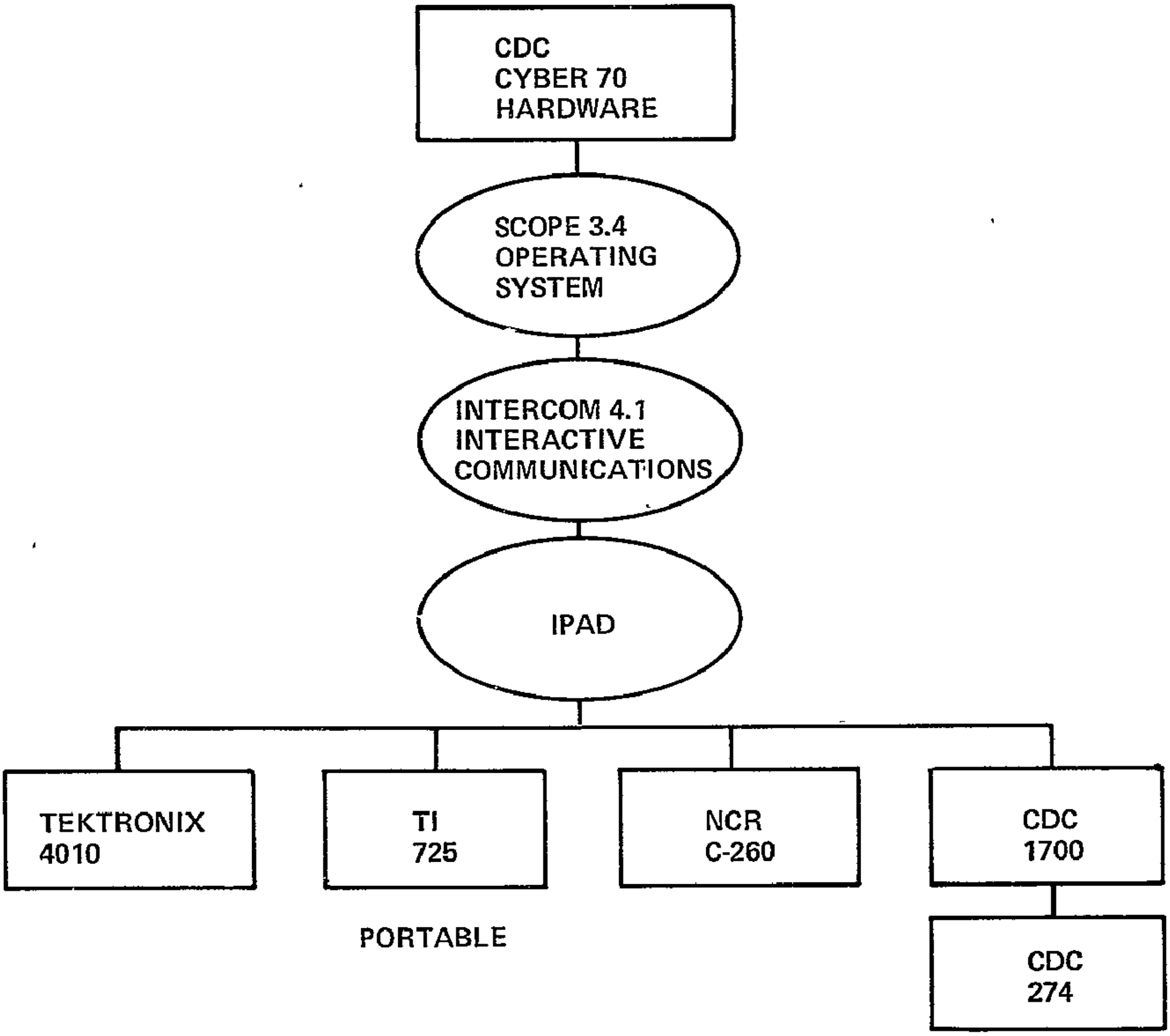


Figure 2-29. A Baseline Computer Configuration

These terminal types typify the range from the smallest (and least expensive) electromechanical typewriter I/O terminal to the large, sophisticated interactive graphics terminals.

2.3.3 The TCS, a command structure. - The objectives which give rise to a Task Control Sequence (TCS) - a command structure - are:

1. The system shall be able to execute complicated task steps automatically with the user in a monitoring role.

2. Unlimited flexibility in the arrangement of the OM execution sequence should be provided.

3. The OM's shall execute in the batch or interactive mode under the IPAD system with the same command structure. Any or all of the capabilities of a given OM must be accomplished (up to the limits of the I/O device) with the same software. (The IPAD system will ignore commands not applicable to the attached device.)

4. The user shall have full control over the design process.

5. IPAD shall be readily adaptable to change, thus improving or extending its useful life.

The design of the command/control system (which IPAD is in an interactive mode) depends principally on the technique of incorporating existing engineering capability (the Operational Modules or OMs) into the system. In this context an OM is defined as:

> A fully functional piece of code which can run stand-alone in batch mode or is a fully checked-out interactive program.

An OM is usually a fully operational existing FORTRAN batch program which represents a portion of an engineering dicipline's computational capability.

Three approach options for incorporating OMs into the IPAD system are briefly investigated in the following subsections.

2.3.3.1 OM-organized system: An OM-organized system is one in which the data paths and error recovery procedures are hardwired - either into the OM or as separate but directly related code - at OM insertion into IPAD. These data paths and procedures typically remain fixed although they can be altered by reprogramming. A typical such system which has been in continuous operation since 1964 is DYNamic Engineering System (DYNES), illustrated in Figure 2-30 and documented in Reference 8 . A brief description of DYNES will serve to illustrate such systems.

DYNES was developed as a means of organizing and automating the preparation and handling of data involved in serial batch execution of various programs. This non-interactive program has four major roles (Figure 2-30). An input section executive analyzes the instructions on punched cards supplied by the user and determines what course of action is required. DYNES then calls the required programs from the DYNES program library tape and prepares them for serial execution. The data-gathering section prepares the input data tape for the programs sequenced for execution. Separate small formatters in DYNES manipulate the data (from the merged data tape supplied) into the exact format required for the specific programs. The DYNES program library tape contains the programs currently being run in DYNES.

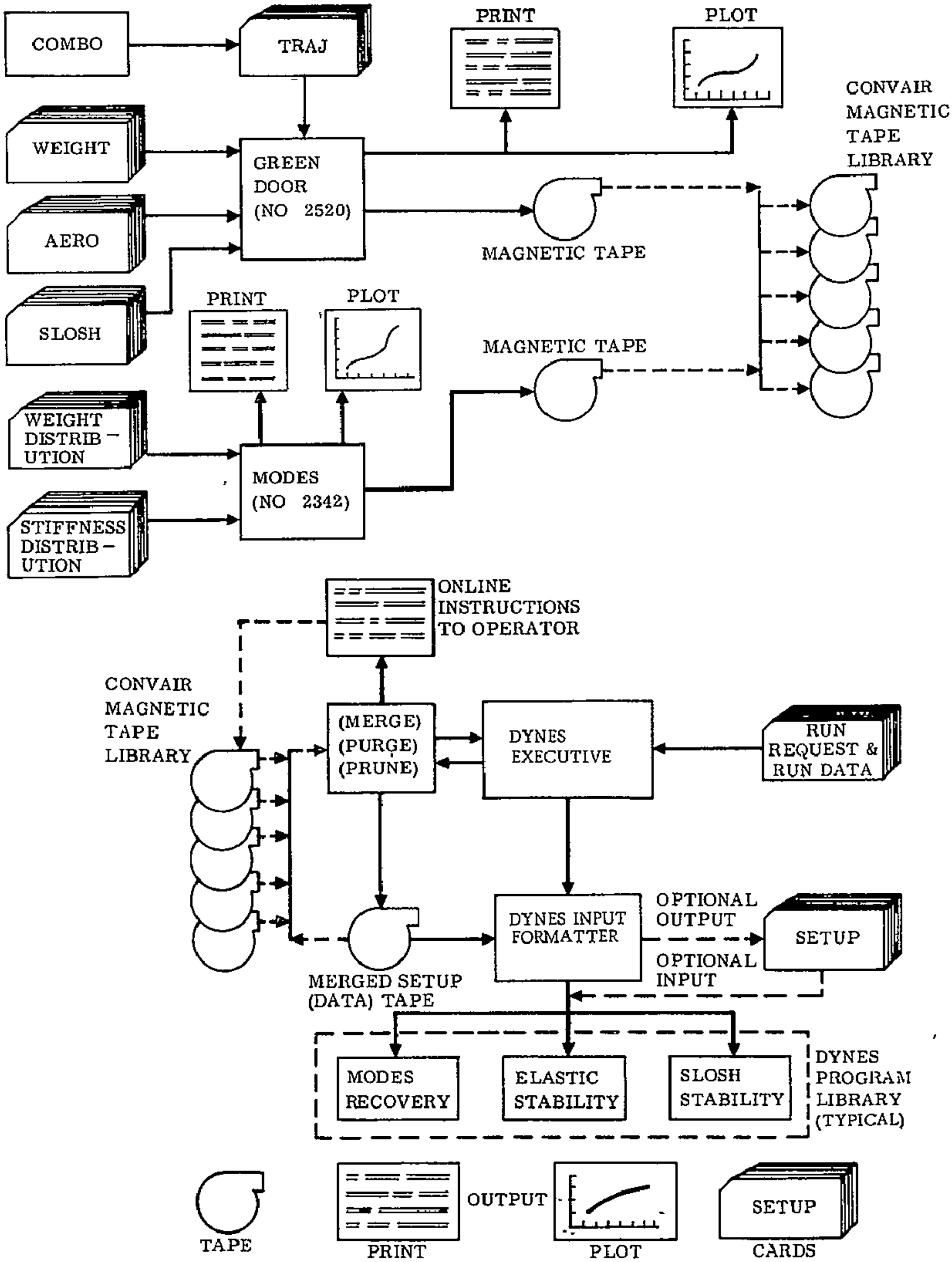


Figure 2-30. Data Flow Diagram for DYNES

DYNES also contains a diagnostic section capable of error detection and diagnostic message printout. The magnitude of the error determines the resulting action. In some situations no corrective action will be attempted, while in others certain assumptions will be made (and reported) so that the run may proceed. Some predetermined obvious errors can be compensated. When no such action can be taken, the case will be bypassed, and the run will continue. Only in extreme circumstances will the system immediately terminate the run.

The most important feature of the DYNES system is its versatility. Each of the programs under DYNES can be run as a "stand-alone" as well as in DYNES. Any phase of the operation can be run as desired, or the entire operation can be serially executed. The output from any link may be requested before proceeding. As an example, the setup deck option may be requested so that DYNES may be run as a stand-alone from card input, thus bypassing the MERGE setup tape.

Another such frequently used system supporting structural dynamics' analyses and in operation since 1968 is DYNAMO (Reference 9).

The advantages of this approach (Figure 2-31) evolve primarily around the cost structure of IPAD system implementation:

1. The technique is a proven technique which has been operational for at least a decade; system development costs and risks are minimal.
2. The costs of incorporating each OM into the IPAD system is borne by that O when programmed into the system. Full implementation cost for such systems are often (erroneously) reported as being the least (compared with othe systems) due to insertion costs being "lost" within the OM development.
3. It has potentially the highest computer efficiency since the data paths and error recovery - if any - are usually not sophisticated and are either programmed into the OM or as separate (usually single) coded modules.

The advantages of this approach are completely overturned by the disadvantages, particularly those restrictions associated with data required/supplied by the OM:

1. There is no ability to modify data paths without reprogramming.
2. It places extensive restraints on the data base structure to the point that the base tends to become fully constrained as the number of OMs in the system become large. (Consider the OM-to-OM interconnection as the number of OMs becomes large.)

FEATURES

- DATA PATHS FOR OM'S HARDWIRED AT OM INSERTION AND REMAIN FIXED
- ERROR RECOVERY HARDWIRED WITH OM'S

ADVANTAGES

- COST FOR OM INSERTION BORNE BY INDIVIDUAL OM'S
- HIGHEST CPU EFFICIENCY
- LEAST INITIAL SOFTWARE COSTS

DISADVANTAGES:

- HIGHEST OVERALL SOFTWARE COSTS
- MOST INFLEXIBLE
- LONG DELAYS FOR OM INCORPORATION
- MUCH MORE CODE ASSOCIATED WITH EACH OM (SUBSTANTIAL DUPLICATION OF CODE)
- NO ABILITY TO MODIFY DATA PATH
- PLACES CONSTRAINTS ON DATA BASE STRUCTURE.

Figure 2-31. OM-Organized System Attributes

3. The programming queue to incorporate OMs into the system due to the limited availability of programmers becomes unsurmountably long, leading to long delays for OM incorporation (and hence poor first impressions).

4. The required programming associated with each OM leads to much duplication of code and hence increased costs. (Consider the required reformatting function of data into and out of the data base.)

5. The most inflexible (hardwire system with extensive data base restrictions).

6. Highest overall cost (all of the above).

The most significant operational disadvantage is the data base restriction (2 above).

2.3.3.2 Self-organized system: At the other extreme is a self-organized system (Figure 2-32) for which the data requirements for OMs are deciphered during execution with errors detected automatically, resulting in alternative approaches (branch paths). User interaction is required in such systems only when the IPAD Executive is stumped and requests additional information.

FEATURES:	
• DATA PATHS FOR OM'S DECIPHERED DURING EXECUTION • ERRORS DETECTED AUTOMATICALLY RESULT IN BRANCH PATHS • USER INTERACTION REQUIRED ONLY WHEN EXECUTIVE STUMPED	
ADVANTAGES:	DISADVANTAGES:
• NEAREST TO FULLY AUTOMATIC • READILY ADAPTS TO DATA BASE CHANGE • BASICALLY INSENSITIVE TO SKILL OF USER	• USER CAN FEEL ALIENATED THROUGH NON-PARTICIPATION (THUS PROPAGATING ERRORS) • EXTREMELY COMPLEX LOGIC PROGRAMMING IN SOPHISTICATED EXECUTIVE • FAIRLY INFLEXIBLE • LARGEST SOFTWARE DEVELOPMENT • DIFFICULT TO MODIFY • LOWEST CPU EFFICIENCY

Figure 2-32. Self-Organized System Attributes

The advantages are particularly attractive to the user and are a direct result of the exceptionally "smart" executive. If designed correctly it places essentially no restrictions on the data base.

The disadvantages arise principally out of the extremely complex logic programming required for the sophisticated executive:

1. Largest software development costs and risks by far.
2. Fairly inflexible - once programmed - since it is difficult to modify due to the built-in software sophistication.
3. Lowest computer efficiency due to the required built-in logic check/recovery procedures.

A perhaps unusual (but real) problem is that such systems tend to be so automatic (once programmed correctly they make few errors) that the user can become alienated through non-participation. When this happens apathy sets in and the user errors increase radically.

Due to the sophistication involved, there are few if any such systems in daily operation with the possible exception of advanced computer operating systems.

2.3.3.3 User-organized system: A compromise between the two preceding approaches is a user-organized system which obtains the sophistication of the self-organized system through user interaction yet has the inherent simplicity of the OM-organized system. The data paths (Figure 2-33) are "softwired" (i.e., interactively constructed by the user) during checkout following initial OM incorporation into the system and interactively modified during use as required.

The advantages combine those of the two proceding systems and in addition the approach is most flexible, highly adaptable to changing conditions, and most easily modified/updated. Since the individual users are only responsible for their own OMs, it features the fastest incorporation of OMs by a substantial margin. Further the user is an involved participant in the process (involvement conquers boredom). Perhaps surprising, this approach requires the least overall software development because of the primitive executive required - the user himself functions as the executive.

FEATURES:

- DATA PATHS FOR OM'S SOFTWIRED (CONSTRUCTED BY USER) DURING CHECKOUT FOLLOWING INITIAL OM INCORPORATION
- DATA PATHS FOR OM'S MODIFIED AS REQUIRED BY USER DURING USE

ADVANTAGES

- FASTEST INCORPORATION OF OM BY SUBSTANTIAL MARGIN
- MOST FLEXIBLE
- MOST ADAPTABLE TO CHANGE
- EASILY MODIFIED / UPDATED
- USER IS MORE INVOLVED PARTICIPANT
- LEAST OVERALL SOFTWARE DEVELOPMENT (USER IS THE EXECUTIVE)

DISADVANTAGES:

- HIGH USER SKILL REQUIRED DURING INITIAL OM INCORPORATION (ERRORS MADE DURING INCORPORATION MAY PASS UNNOTICED FOR SOME TIME.) CAN BE SUBSEQUENTLY FOULED UP BY SLOVEN USER.
- SLIGHTLY LOWER CPU EFFICIENCY
- USER MUST ADAPT TO DATA BASE CHANGES AS THESE OCCUR

Figure 2-33. User-Organized System Attributes

The disadvantages are a direct consequence of the advantages gained from user interaction/direction: a higher skill level is required by the user who inserts his own OMs, slightly lower net computer efficiency (compared to the OM-organized system), and it is the user who must adapt to change (e.g., changes in the data base structure). The most significant disadvantage is that any "softwiring" errors made during initial OM incorporation may pass unnoticed for some time invalidating those results; linkage can also be incorrectly modified (fouled up) by a user possessing modification authorization.

The user-organized system approach is adopted for IPAD as being by far the most advantageous.

2.3.3.4 An example of a TCS: We can now describe a Task Control Sequence (TCS) as follows:

1. What are they? TCSs are command or control sequences generated during IPAD program initialization. They perform the same type of function as the CDC "control cards" or the IBM Job Control Language (JCLs). They are stored on files for subsequent use of modification and reuse.

2. How are they generated? TCSs are written with the help of IPAD utilities using the interactive capabilities of IPAD. The most efficient method of generation appears to be in using interactive graphics and menus of operations embodied in TCS strings.

3. How are they used? The IPAD Executive program manipulates (through the host computer system's interactive communications or batch subsystems) the utilities designated by the TCS commands.

4. What are their limitations? The TCSs can only use the IPAD utility or OM programs known to the IPAD EXEC which processes it. Note however that IPAD will include the standard systems utilities as a subset.

Figure 2-34 is offered as an example of a segment (string) of a Task Control Sequence (TCS), a sequence of user control commands to the primitive IPAD executive. The TCS in a batch mode of operation differs little from the Operating System Control Language (OSCL) for control of the job process (CDC's Control Cards)

Careful reading through Figure 2-34 will suggest the intended structure of a TCS The meaning of "substructure" (middle left of figure) implies the ability to step forward or backward in the control sequence in an arbitrary course (large step) to fine (single step) fashion. The ability to backspace a TCS (to reverse the result of that execution) and interactively modify the TCS (by recording the job steps while accomplishing same) is envisioned.

It is estimated that Figure 2-34 (including deleted steps) represents approximate two hours concentrated work at a CRT interactive terminal. The functions denoted

- SIGN ON
- ACCESS TASK STATUS FILE
- DISPLAY (LIST) FILE
- ACCESS USER MACRO MENU FILE
- DISPLAY (LIST) MENU
- ACCESS MICRO MENU
- DISPLAY (LIST) MENU
- SELECT (ACCESS) OM
- ACCESS TCS FILE
- DISPLAY (LIST) FILE
- SELECT (ACCESS) TCS
- SELECT TCS SUBSTRUCTURE
- STEP TCS

(additional access steps omitted for brevity)

- BACKSPACE TCS
- SET TCS TO MODIFY
- ACCESS WEIGHTS DATA FILE
- DISPLAY (LIST) CONTENTS
- SELECT (ACCESS) DENSITY
- FETCH DENSITY SUBSET
- SELECT MODEL FILE
- DISPLAY (LIST) FILE CONTENTS
- SELECT MODAL MODEL
- FETCH MODEL
- SELECT TOPOLOGICAL I/O
- SUPERIMPOSE MODEL AND DENSITY
- ENCIRCLE NODE WITH BOUNDARY
- LUMP MASS AT NODE

(additional topological I/O steps omitted for brevity)

- ACCESS AERODATA
- DISPLAY (LIST) CONTENTS
- SELECT AERODATA SUBSET
- DISPLAY (LIST) DESCRIPTORS
- SELECT UPDATE MODIFIERS
- FETCH AERODATA SUBSET

(additional coordinate access steps omitted for brevity)

- EXECUTE TRANSFORMATION UTILITY

(additional data linkage steps omitted for brevity)

- EXECUTE DATA LINKER
- SKIP FORWARD TCS
- EFFECT TCS MODIFICATION
- SET TCS SUBSTRUCTURE
- STEP TCS

(additional OM execution steps omitted for brevity)

- TERMINATE TCS
- DISPLAY (LIST) LOCAL FILES
- SELECT (ACCESS) FILE
- DISPLAY FILE CONTENTS

(additional file editing steps omitted for brevity)

- EDIT FILE COPY
- DISPOSITION FILE COPY (EG TO UPDATE FILE)
- DISPOSITION FILE

(remaining file disposition omitted for brevity)

- ACCESS TASK STATUS FILE
- ACCESS TEXT EDITOR

(text and editing steps omitted for brevity)

- EFFECT TASK STATUS FILE MODIFICATION
- SIGN OFF

Figure 2-34. Typical IPAD Operation From An Interactive Console With a TCS String

(e.g., ACCESS, DISPLAY, SELECT, FETCH) are to be supplied by appropriate utilities.

2.3.4 Incorporation of the OMs. - The process of incorporation of OMs into IPAD should meet the following objectives:

1. The intended user shall be able to incorporate existing OMs (which are numerous) without requiring modifications (or with only simple mods, core size limitations permitting).

2. If usage indicates sufficient improvement can be achieved through OM modifications, provisions shall be made to replace the existing OM with a modified version at a convenient time. (Usage of the replacement should not be apparent to the user except through faster response).

3. IPAD system operation shall not be bound to any particular level of sophistication of the process. The system should be amenable to the level of need of the particular user at any given time.

4. IPAD shall be able to absorb new developments in design, analytical techniques, and system technology as they occur with little or no modification.

The technique envisioned is illustrated in Figure 2-35. To the existing Operational Module (OM) is added:

1. An IDEF (Input DEFinition) which describes (or otherwise defines) all the input required by the OM.

2. An ODEF (Output DEFinition) which describes (or otherwise defines) all the output produced by the OM.

As noted in Figure 2-35 the resulting "system OM" is unmodified at initial incorporation for speed of implementation. If indicated, the OM may be reprogrammed (by stripping out the unnecessary input/output formats and output report-type headers) and the IDEF/ODEF correspondingly reduced. It is presumed that the justification would be that the resulting improvement in disc/computational/response efficiency would cover the reprogramming costs. Note that the unmodified version of the "system OM" is operational until replaced by the improved version. This provides unlimited time to program/checkout the improved version.

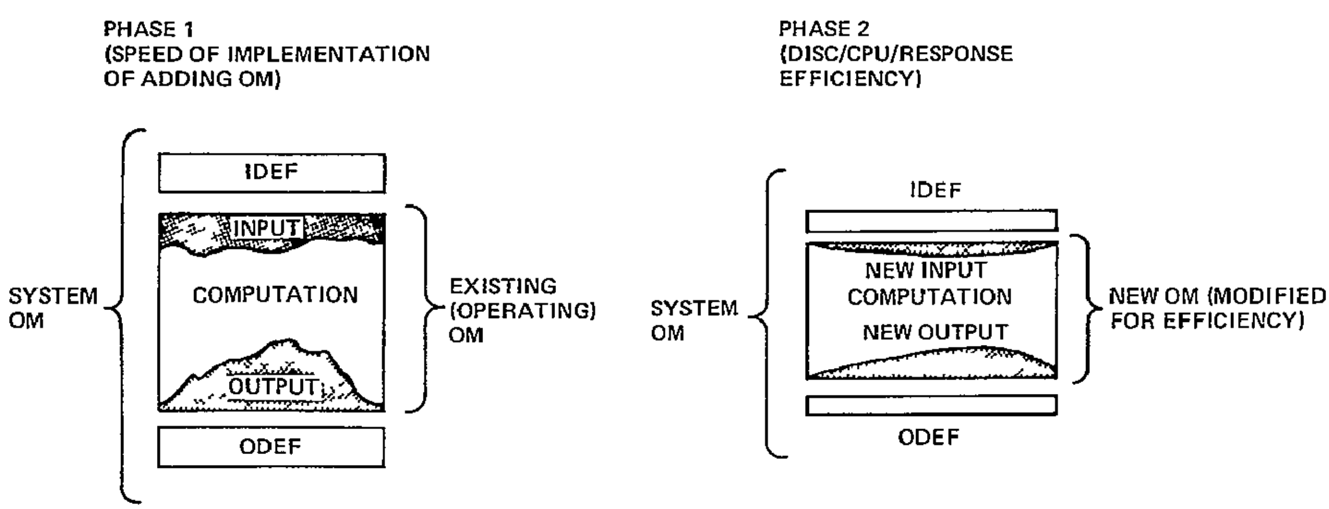


Figure 2-35. Incorporating Existing OMs Into IPAD

The IDEF and ODEF are further illustrated in Figure 2-36. Note that they are identical in content except for the input defaults (values assumed in lieu of values input) which are meaningful only for the IDEF. Further, most of the information is provided for viewing and interpretation by the user who provides user-direction to the system for control of formatting that information required for input and resulting from output. Not required by the user (and hence invisible to the user) are the locations and format of each input/output variable in their corresponding files.

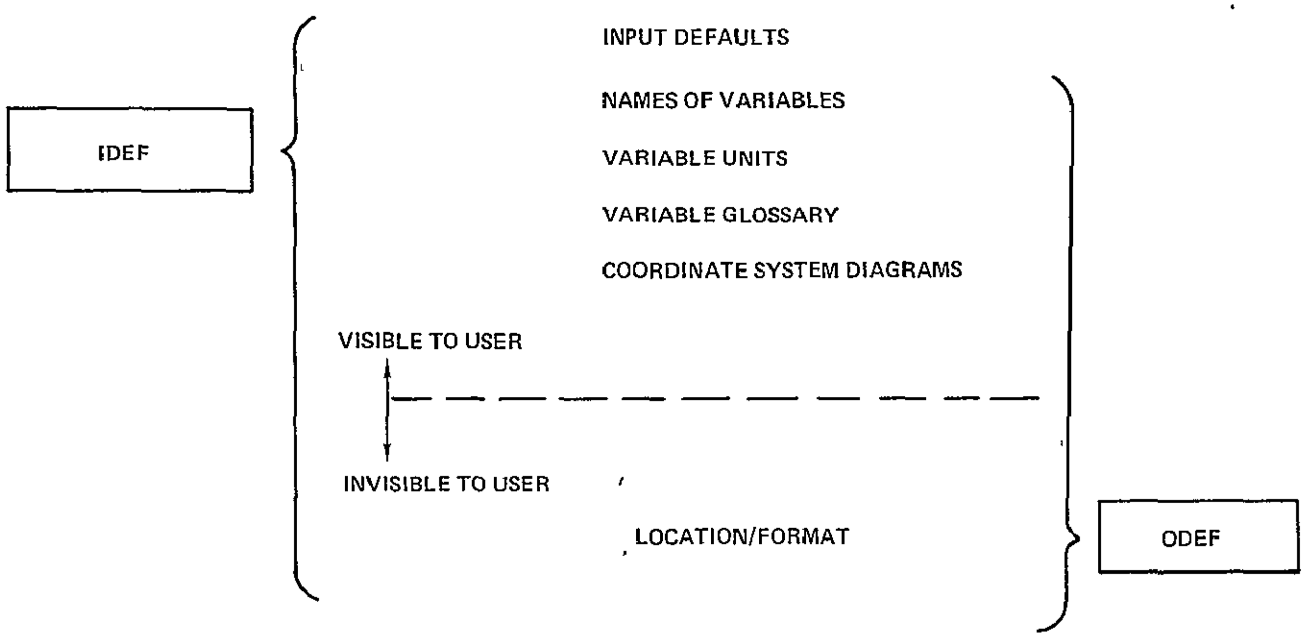


Figure 2-36. I/O Definition Substructure

It is envisioned that the inputs/outputs are controlled at the variable level, i.e., for each variable in the input (output) is provided the:

1. Name by which it is to be known.
2. Units (if applicable).
3. Glossary definition.
4. Coordinate system diagram (if applicable).
5. Default value (if an input and applicable).

2.3.4.1 The I/O Formatter (IOF) utility: The interactive utility that interfaces the user with the system for the purpose of formatting input or output is called the I/O Formatter or IOF (Figure 2-37).

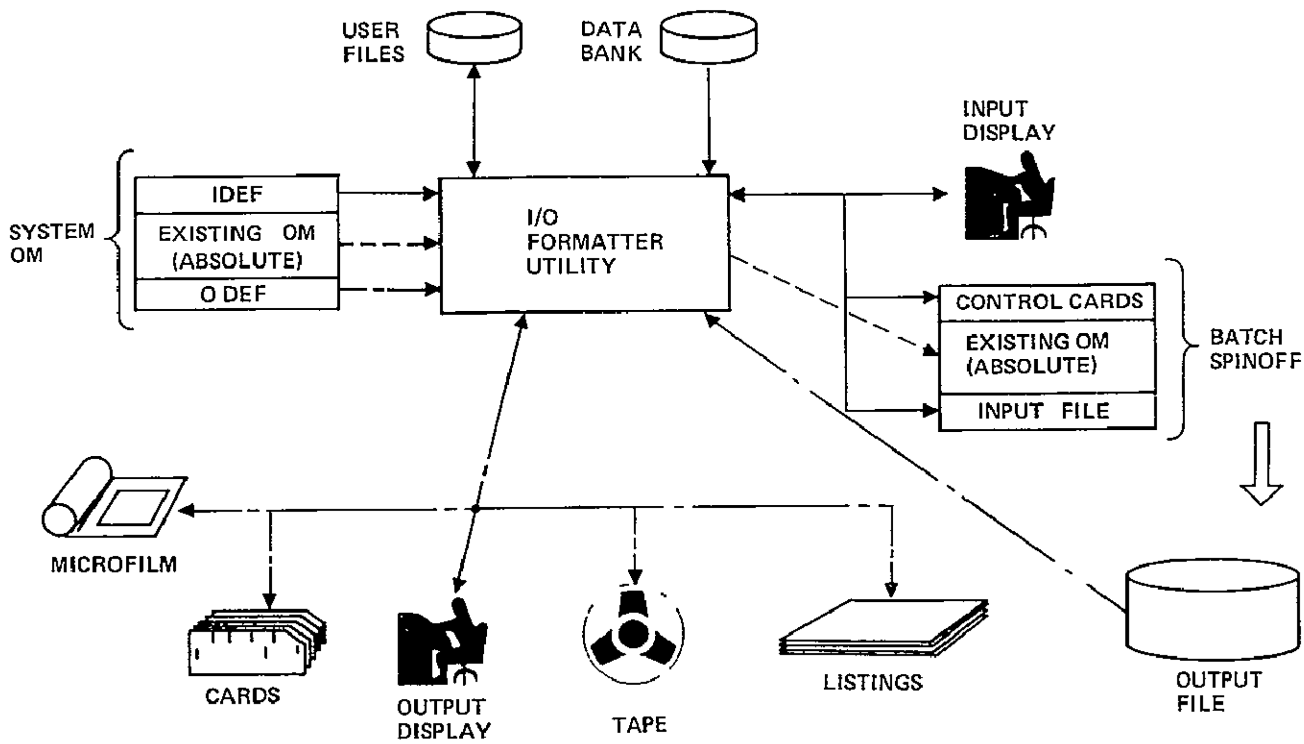


Figure 2-37. I/O Formatter Utility

Sitting at the interactive terminal the user interacts with the IOF (solid lines of figure):

1. Examining the input definition tutorial portion of the IDEF.
2. Acquiring data from the:
   a. Multidisciplinary Data Bank, which is the repository of design data.
   b. User files, which is provided to each user for storage of intermediate results, saved output from previous runs, etc.
3. Constructs the input file for the OM.

When complete, a user command causes the IOF to (dashed line of figure):

4. Finalize the input file.
5. Locate and supply the "system/OM" absolute (loaded object) code.
6. Supply the control card sequence (TCS string).
7. Cause the job to be executed.

The execution shown in the figure presumed a typical batch (i.e., non-interactive) job is being processed as a batch "spinoff"; while the job is being executed the user is busy at a different (but perhaps related) interactive task such as plotting previous results. Upon being informed by the system that the job has been completed, the user at the interactive terminal interacts with the IOF (broken dash, figure):

8. Examining the output definition tutorial portion of the ODEF.

9. Acquiring data from the output file and distributing it to:

   a. Listings.

   b. Magnetic tape.

   c. Punched card deck.

   d. Microfilm plots.

   e. User files (not shown).

If the job is terminated abnormally, the user examines the partial output file together with the input file and can, perhaps, rectify what went wrong.

The distinguishing feature of the conceptual design to this point are:

1. It presupposes an interactive environment, especially during initial OM linkage, to configure the required TCS.

2. Following initial linkage, specific tasks can be conducted in a batch environment via the TCS or interactively as appropriate.

3. OM incorporation as envisioned (and hence the IPAD design) is not explicitly tied to any discipline or process but only to the systematic way of conducting that process. As such, IPAD can accommodate many differing OMs.

2.3.5 IPAD system framework. - IPAD software can be thought of in two distinct contexts:

1. All IPAD related code in a fully complemented system, including all applicable Operational Modules (OMs).

2. The IPAD "system" framework consisting of IPAD without any OMs contained within the system.

This subsection is concerned with the IPAD system framework software.

The developmental objectives as related to the IPAD system software are:

1. Strive for the earliest release possible for the IPAD system consistent with satisfying the system objectives and immediate user needs.

2. Minimize the impact of future computer hardware/software development.

3. Avoid (where practicable) non-standard software development.

4. IPAD software shall be modular to the functional level.

5. In both design and implementation, all machine dependent code shall be clearly identified and isolated.

6. The IPAD system software shall be structured modularly to aid in reducing the time and effort required in transfering IPAD software to different hardware or software installations.

As envisioned, the IPAD system software consists of the:

1. EXECUTIVE which links the user with the host computer's operating subsystems, viz.:

   a. The interactive communications subsystem (e.g., INTERCOM 4.1 SCOPE 3.4).

   b. The batch (or remote batch) operating subsystem (e.g., CDC's SCOPE 3.4).

2. The I/O Formatter (IOF) utility which links the user with IPAD's data bases to provide input/output control over the user's OMs.

3. The Data Base Handler (software) which allows the Data Base Administrator (DBA, a person or group) to structure (restructure) the data base and update/delete data as required.

4. Special Purpose Utilities (SPUs) which are used principally by IPAD users in incorporating their OMs.

5. General Purpose Utilities (GPUs) which are used principally by IPAD users in conjunction with their routine tasks.

Figure 2-38 presents the typical contents of the envisioned GPU library.

2.3.6 The data bases. - As related to the data bases, IPAD must be able to:

1. Subordinate large self-contained projects consisting of many diverse disciplines under IPAD control.

2. Display current system design through on or off-line computer displays.

3. Allow user to incorporate proprietary features or classified data into the system.

- STATISTICAL PACKAGE
    - SAMPLE STATISTICS
    - TOLERANCE INTERVALS
    - CHI - SQUARED TEST
    - RANDOM # GENERATORS
- FILE MANAGEMENT
    - LIST FILES ATTACHED
    - SEEK FILE
    - ACCESS FILE
    - FILE DISPOSITION
        - CATALOG FILE
        - PURGE FILE
        - PLACE IN INPUT QUEUE
        - PLACE IN OUTPUT QUEUE
    - FILE EDITOR
    - COPY UTILITIES
    - LIST FILE CONTENTS
- DATA TRANSFORMATIONS
    - PRINT
    - PLOT
    - PUNCH
    - COORDINATE ROTATION
    - COORDINATE TRANSLATION
    - CURVE FIT
        - TRIGNOMETRIC
        - POLYNOMIAL
        - SPLINE
- OM MONITORING CONTROL
    - ROLLOUT TO FILE
    - RESTORE FROM FILE
    - RESTART
    - TERMINATE
- PROGRESS STATUS
    - CPU TIME
    - DISK SECTORS
    - LINE COUNT
    - CARD COUNT
    - COSTS
- TOPOLOGICAL I/O (CRT)
    - DISPLAY
    - EDIT
    - GROUP (LUMP) NODES
    - CREATE BRANCH
- TUTORIAL CLUES
    - MEANING ??
    - WHERE AM I ??
    - OPTIONS ??
    - WHAT'S TRANSPIRED ??
    - YOU THERE ??
    - PROVIDE HELP ??

Figure 2-38. General Purpose Utility Library Typical Contents

4. Retain user results for ease of modification during operation.

5. Allow data to be re-structured in a manner suitable to each OM.

6. Allow more than one OM to retrieve data concurrently.

7. Protect data against unauthorized modifications (or accidental destruction by undebugged programs).

8. Provide users with the capability of informing an authorized modifier of data needing updating.

9. Provide for mass storage device independence.

10. Make user OMs and IPAD software as independent of data format practicable.

11. Provide separate descriptions of the data in the data base and the data as known to a utility or an OM.

12. Provide and permit the use of a variety of search strategies for data retrieval.

13. Provide IPAD with a centralized capability to control the physical placement of data.

14. Be relatively unaffected by the size of a rapidly expanding data base which will inevitably grow quite large. (However its design should be such that redundant data copies and data file interaction are minimized without over-burdening the host computer operating system or inconveniencing the user.)

The conceptual design envisions that:

1. Every file is to contain its own (arbitrary) file structure.

2. Every file is to contain its own (arbitrary) file contents/directory.

3. Provision is to be made for every file variable to have a definition (in the file glossary) and its units/ coordinate systems as applicable.

This will provide maximum flexibility while still providing ease of use of the resulting proliferation of file types.

The following subsection discusses the file types identified to IPAD (as conceived).

2.3.6.1 Task action and status files: Figure 2-39 presents typical contents of the Engineering Review Board (ERB) Action File as it might appear on an interactive terminal display. The intent of the Action file is to inform the ERB of the status or results of requests the ERB has made and actions required of it in conjunction with its responsibilities.

The typical contents of the Task Directive/Status File as applied to the Performance group is shown in Figure 2-40. Note that the request denoted from RAT is from the Risk Assessment Team.

2.3.6.2 Multidisciplinary Data Bank (MDB): The Multidisciplinary Data Bank (MDB) is that portion of the IPAD data bases reserved for project approved ("blessed") data. It is the responsibility of the project's Data Base Administrator (DBA) to review and

- RESULTS OF WING SIZING STUDY WITH RECOMMENDATIONS
- LANDING GEAR STRUT LOADING PROBLEM: URGENT
- RESULTS OF CANARD LOCATION SUB-OPTIMIZATION
- PROGRAM COST PROJECTION: 5% OVERRUN
- CRITICAL PATH ANALYSIS 10 PROBLEMS, -4 WEEKS SLACK, URGENT
- MANPOWER LOADING ANALYSIS· 7 SKILL, 11 LEVEL PROBLEMS
- RISK ASSESSMENT: 4 CRITICAL AREAS DECISION URGENT
- NEW IDEA· USE OF ALLOY 718 TO RELIEVE NOSE CONE TEMPERATURE PROBLEM
- THREAT ASSESSMENT, (CLASSIFIED TITLE) INABILITY TO MEET
- MESSAGE: WILL BE OUT OF TOWN, CAN'T MAKE TUESDAY SESSION. R. SMITH
- REQUEST NEED DATE SLIPPAGE. CRUISE PERFORMANCE TO 4/18/77

Figure 2-39. ERB Action File
(Typical)

- PERFORMANCE
  - DASH. UPDATE COMPLETE AND SUBMITTED 03/29/77

    *** REJECTED ON CREDIBILITY *** ACTION BY 04/05/77
  - LOITER: UPDATE IN WORK, EXPECTED COMPLETION 04/07/77

    *** NEED DATE SLIP *** NEW NEED DATE 04/18/77 - 15 30
  - REQUEST (ERB). EFFECT OF WING SIZING STUDY ON DASH PERFORMANCE - USE UPDATE IDENT WINGSIZE WHEN AVAILABLE. NEED DATE 04/15/77 - 15:30
  - CRUISE IN HOLD AWAITING PROPULSION GROUP WEIGHTS UPDATE
  - REQUEST (RAT)· PROVIDE SENSITIVITY DATA ON EFFECT OF PROPULSION VARIABLE STRING ON DASH PERFORMANCE. NEED SUGGESTED COMPLETION DATE

Figure 2-40. Performance Group's Task Directive/Status File
(Typical)

approve all data submitted for inclusion in the MDB; the DBA (or his alternate) is the only one authorized to modify the MDB.

Figure 2-41 illustrates the data bank update "pages" and the archiving procedure. Past as well as present baselines must be available from the data base to support asynchronous design studies (some in progress, some completing and others just starting). It is envisioned that some fifty different baselines might be available over a three calendar week period in an IPAD environment; these fifty are the combined total of the various design versions as well as each version's past updates. If each of fifty versions/updates were stored as complete data sets, the mass storage requirements would be nearly fifty times greater than that actually required. The update (version) procedure illustrated (utilizing a Venn diagram concept for illustration purposes) employs only one complete data set (shaded at bottom of figure). Each

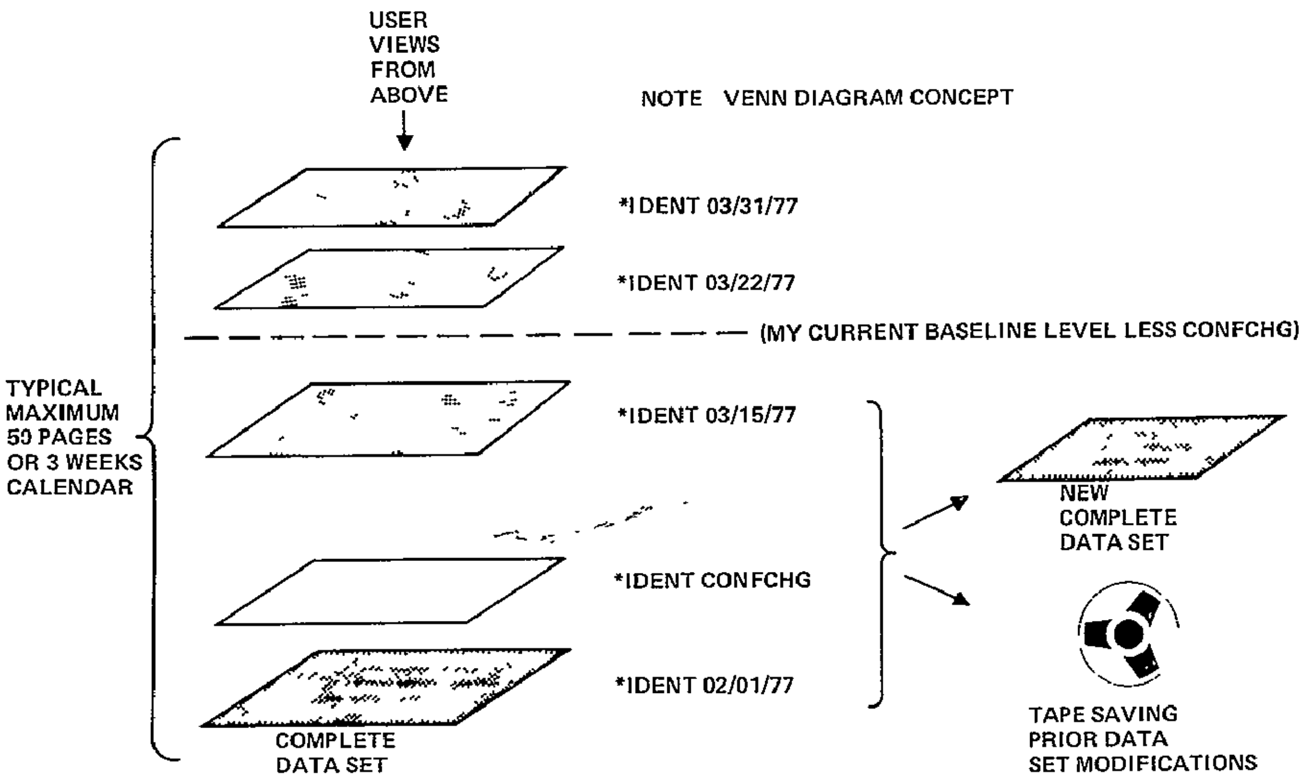


Figure 2-41. MDB Data Base File Management

version/update modifies only that portion of the data (shaded portions) requiring modification from the preceding baseline. The user views the current baseline from above, imaging all changes on the original complete data set. Periodically, the Data Base Administrator (DBA) establishes a new baseline level and merges the original complete data set and levels below the established baseline into a new complete data set; at the same time the original complete data set and the merged updates are copied onto magnetic tape to save the data for possible recovery. The new complete data set could similarly be archived. In this way nearly the total content of the merged updates can be eliminated from mass storage.

The update concept presented in Figure 2-41 is nothing more than a conceptual extension of CDC's UPDATE utility for coded card files (usually computer source code files). To obtain the data baseline denoted "my current baseline level less CONFCHG" requires ignoring correction sets 03/22/77 and 03/31/77 and also removing CONFCHG which might be accomplished via a TCS string as follows:

ACCESS DATA BANK

FETCH AERO

FETCH WEIGHTS

PULL ID 03/31/77

PULL ID CONFCHG

FETCH DATA SUBSET

2.3.6.3 Other data bases: It is envisioned that all changes made to the MDB pass through a special file for review and approval by the DBA prior to actual revision of the MDB. Figure 2-42 suggests the contents of the MDB Update File.

- WING WEIGHTS RESULTING FROM SIZING STUDY 03/11/77 - 181
- ERROR CORRECTION: HINGE LIMIT LOADS FOR CANARD
- WEIGHTS UPDATE: PROPULSION GROUP 04/03/77, URGENT
- AERODYNAMIC TRIM DATA FOR SUB-MODEL 3 - 311
- PERFORMANCE UPDATE: DASH 03/29/77 *** REJECTED ON CREDIBILITY *** ACTION REQUIRED

Figure 2-42. Data Update File Typical Contents

2.3.6.4 Miscellaneous file types: In addition to the above, IPAD data bases consists of the Utility Library, User's Library File and User's I/O Files as was previously discussed in Subsection 2.3.1, in Figure 2-19. The contents of each of these files is typified in Figures 2-38, 2-43, and 2-44 respectively. Note that to constrain the User's Library File size to a maximum allowable (Figure 2-43), these files or the least-used file parts (records) may actually be transient from tapes or private disk packs.

Note further that the maximum allowable size of User I/O File (Figure 2-44) remaining at signoff is an installation parameter, e.g., automatic procedures will copy the least used data from disk to tape in the event that a User's I/O File size exceeds the allowable.

- OM LIBRARY CONTENTS (MACRO & MICRO MENUS)
- OM FILE
- USER REFERENCE DATA (e.g. EXPERIMENTAL DATA FITS, MATERIALS PROPERTIES DATA, ETC.)
- SENSITIVITY DATA FILE
- OM RELATED TCS LIBRARY CONTENTS
- TCS FILE

Figure 2-43. User's Library File, Typical Contents

- RESULTS OF PREVIOUS ANALYSES (INCLUDING INPUT)
- SENSITIVITY INFORMATION IN PROCESS
- TRANSIENT DATA FROM TAPES (EG, WIND TUNNEL RESULTS)
- CURRENT CARD INPUT (READ IN BEFORE CURRENT RUN)
- PARTIALLY CONSTRUCTED OM INPUT FILES
- PARTIALLY DIGESTED OM OUTPUT FILES
- UNRELEASED SCRATCH AND OTHER LOCAL FILES
- UNDISPOSED CARD OUTPUT FILE
- UNDISPOSED MAGNETIC TAPE FILE (TEMPORARILY ON DISK)
- TCS MODIFICATION STRINGS

Figure 2-44. User's I/O File, Typical Contents

2.3.7 Summary of features of the Conceptual Design. - The conceptual design of IPAD, as envisioned, possesses the following features:

1. IPAD is envisioned as a structure (or framework) within which aerospace vehicle design (or any procedurally oriented task sequence for that matter) can be accomplished with speed, efficiency, and confidence. Note however that true operational efficiency is how quickly and easily the user can accomplish his task, not merely how fast or efficiently the computer accomplishes its task.

2. IPAD is envisioned to be user-organized (rather than self-organized). The user provides the overall system management during execution. In this regard, applications evolve within the IPAD structure provided.

3. It is easy to lose sight of the fact that the user is the key driver and decision-maker in the process. In this conceptual design, the user himself can be thought of as the principal IPAD executive routine.

4. IPAD in this conceptual design is relatively insensitive to the user's skill except during initial OM linkage. Each user can proceed through his task at his own pace skipping any monitoring steps he deems advisable under the circumstances.

5. IPAD as envisioned does not require a specific file structure. The structure, type and contents of the file are a part of that file; only that portion of information desired by the user is presented to him for consideration.

6. IPAD is equally applicable to any level of the (design) process. The major differences that do exist are the extent of hardware resource tieup at the succeedingly more complex levels.

7. Any objective (or merit) optimization function can be implemented through IPAD in either an interactive or non-interactive mode.

8. IPAD software is envisioned as a utility structure so only that portion of code servicing the task at hand (e.g. CRT display) resides in central memory. Substantial use is made of the computer system utilities (a part of the host computer system's provided software).

9. Interactive graphic operations, interactive non-graphic operations, and non-interactive (batch) operations all use the same IPAD system software.

10. As envisioned, IPAD is aware of the attached device's I/O limitations (relieving the user of that burden). It circumvents any TCS step that is not applicable to the device to which it is attached.

In the course of evolving the conceptual design, an operating philosophy was similarly evolved. The envisioned IPAD system in operation will feature:

1. User oriented design.
2. Man in the loop with appropriate man-machine interfaces.
3. Flexible system organization and command structure (TCSs).
4. Modular system and engineering software.
5. Ease of adaptation to change and growth.
6. A common multidisciplinary, safeguarded data bank (MDB).
7. Random data access.
8. Ease of implementation in various computer systems.
9. Time sharing of the CPU during interactive operation.
10. Program roll in/roll out (swap in/swap out) during interactive operation (i.e., memory sharing).
11. Language versatility
12. Acceptance of existing OM software (code).

2.4 Selection Studies

The following selection studies can be identified in support of this conceptual design:

1. IPAD utility library software required (Section 4.5).
2. Determine number and type of I/O terminals (Section 5.2).
3. Determine host computer hardware/software support (Section 5.4).
4. Existing OM's language/size limitations (deferred to Volume V, Part II, Section 8).
5. IPAD's operating philosophy and design (embedded throughout this volume, but best summarized in Volume V, Section 1 of Part III).

The report section where the answers may be found is noted.

3 IPAD USER SURVEY

Early in the IPAD feasibility study a survey was conducted among the current users of computers at several General Dynamics facilities to codify a decision rule for selecting among several IPAD system design approach options and suboptions developed throughout the study. A secondary purpose was to identify those attributes deemed most important by a large cross-section of potential users and designers of such a system. This section presents the results of this survey.

3.1 Survey Description

The IPAD survey contained a computer card deck of objectives to be ranked (ordered) by a cross section of engineering computer users as a guide to the IPAD development. The survey description is summarized in Figure 3-1. The specific objectives to be ranked are defined in Table 3-1.

- WHAT IS IT?
  - 29 QUANTIFIABLE OBJECTIVES (ON CARDS) TO BE RANKED BY THE SURVEY PARTICIPANTS IN ORDER FROM MOST TO LEAST IMPORTANT
  - 3 DIVISION CARDS TO DIVIDE RANK OBJECTIVES INTO

    THOSE APPEARING ABOVE THE 30% CARD CONTRIBUTE ≈30%
    THOSE APPEARING ABOVE THE 50% CARD CONTRIBUTE ≈50%
    THOSE APPEARING ABOVE THE 90% CARD CONTRIBUTE ≈90%
    THOSE APPEARING BELOW THE 90% CARD CONTRIBUTE ≤10%
  - AN ID CARD FOR NAME, DEPARTMENT, EXTENSION, ETC.
- WHAT IS ITS INTENDED USE?
  - TO GET INVOLVEMENT AND PARTICIPATION FROM A LARGE GROUP OF POTENTIAL IPAD USERS
  - TO GET FEEDBACK FROM POTENTIAL USER AS TO THEIR NEEDS AND DESIRES
  - TO DERIVE A LINEAR OBJECTIVE FUNCTION FOR SEVERAL CLASSES OF USERS, TO BE USED IN EVALUATING THE VARIOUS IPAD DESIGN APPROACHES
- THE OBJECTIVE FUNCTION
  - A LINEAR OBJECTIVE FUNCTION IS OF THE FORM $\Sigma W_i O_i$ WHERE THE W_i ARE THE WEIGHTS ASSIGNED TO THE 29 VARIOUS QUANTIFIABLE OBJECTIVES, O_i, WHICH EACH OF THE SURVEY PARTICIPANTS WERE ASKED TO RANK.
  - THE WEIGHTS ARE TO BE DETERMINED THROUGH AN ANALYSIS OF THE SURVEY RESULTS
  - VALUES ARE TO BE ASSIGNED TO EACH OF THE OBJECTIVES (WHICH ARE QUANTIFIABLE) IN A SUBJECTIVE WAY BY SEVERAL MEMBERS OF THE IPAD TEAM HAVING DETAILED KNOWLEDGE OF THE VARIOUS IPAD DESIGN APPROACHES UNDER EVALUATION. EACH OBJECTIVE IS TO BE EVALUATED IN TURN FOR EACH DESIGN APPROACH BEFORE CONSIDERING THE NEXT OBJECTIVE.
  - ONCE ALL THE OBJECTIVES (THE 29 O_i) ARE ASSIGNED VALUES, THE EVALUATION OF $\Sigma W_i O_i$ GIVES A FIGURE OF MERIT ASSIGNED TO EACH DESIGN APPROACH. THE APPROACH WITH THE HIGHEST FIGURE OF MERIT IS THE ONE DEEMED MOST DESIRABLE BY THE GROUP WHOSE RANKINGS GENERATED THE W_i
  - THE OBJECTIVE FUNCTION THUS HELPS FORMULATE AND SELECT THE DESIGN. IT IS A STANDARD TECHNIQUE UTILIZED IN THE RESOLUTION OF COMPLEX DECISION PROBLEMS.

Figure 3-1. IPAD Survey Description

TABLE 3-1
GLOSSARY OF QUANTIFIABLE OBJECTIVES

1.	ACCEPTABILITY	DEGREE TO WHICH IPAD SYSTEM WILL BE ACCEPTED (AND HENCE USED) BY POTENTIAL USERS.
2	ADAPTABILITY	THE ABILITY OF IPAD TO ADAPT TO NEW PROCESSES AND IMPROVEMENTS IN EXISTING PROCESSES/TECHNIQUES
3	AUTOMATION	DEGREE THAT THE SYSTEM CAN PERFORM COMPLICATED TASK STEPS AUTOMATICALLY
4	AUTO-TUTORIAL	DEGREE THAT USER CAN TEACH HIMSELF SYSTEM OPERATION DURING ACTUAL USE.
5	CONTROL	DEGREE TO WHICH THE USER CAN EXERCISE CONTROL OVER THE DESIGN PROCESS DURING USE.
6	CURRENCY	DEGREE TO WHICH CODING MODIFICATIONS ARE NOT REQUIRED TO ACHIEVE INCREASED CAPABILITY
7	DEPENDENCY	DEGREE OF DEPENDENCE ON THE HOST COMPUTER SYSTEM HARDWARE/SOFTWARE
8	DELAYS	THOSE ENCOUNTERED WHEN EXISTING OM'S ARE INCORPORATED INTO THE SYSTEM
9	DUPLICATION	OF FUNCTIONS OF OTHER COMPUTING SYSTEM SOFTWARE (OR WITHIN IPAD PROPER) RESULTING IN DUPLICATION OF CODE.
10	EFFICIENCY, COMPUTATIONAL	OVERALL IPAD SYSTEM PERFORMANCE MEASURED BOTH IN COST AND THROUGHPUT
11.	EFFICIENCY, USER	OVERALL USER PERFORMANCE IN HIS ASSIGNED TASK AS MEASURED BY USEFUL OUTPUT AND LACK OF FRUSTRATING/IMPEDING INCIDENCES
12.	EXPEDIENCY	DEGREE TO WHICH IPAD SYSTEM DESIGN LENDS ITSELF TO AN EARLY RELEASE IN A USEFUL INITIAL OPERATIONAL CONFIGURATION.
13.	INVESTMENT	INITIAL SYSTEM COST, MEASURED TO FIRST OPERATIONAL USE.
14	LIFE	ECONOMIC LIFE OF IPAD SYSTEM, THAT IS ESTIMATED DURATION DURING WHICH SYSTEM REMAINS A COST EFFECTIVE TOOL WHICH CANNOT ECONOMICALLY BE REPLACED WITH AN ALTERNATIVE SYSTEM.
15	MAINTAINABILITY	EASE OF IPAD SYSTEM MAINTAINENCE (SHOULD IT BE REQUIRED) DUE PRINCIPALLY TO ITS DESIGN APPROACH
16	MODULARITY	DEGREE TO WHICH SYSTEM DESIGN IS REPRESENTED AT THE LOWEST FUNCTIONAL LEVEL.
17.	OPEN-ENDEDNESS	DEGREE TO WHICH IPAD SYSTEM DESIGN PROVIDES FOR ADDITIONAL ADD-ON CAPABILITY WITHOUT RESTRICTIONS BEYOND THOSE NECESSITATED BY HARDWARE (EG CPU SPEED) LIMITATIONS
18.	RATING	ESTIMATED RATING A USER WOULD GIVE TO SYSTEM DESIGN AS HE VIEWS IT.
19	RELEVANCY	DEGREE TO WHICH SYSTEM MEETS THE SPECIFIC NEEDS OF THE USER
20.	RESPONSE	RESPONSE TIME EXPERIENCED BY A USER DURING OPERATION WHICH IS ATTRIBUTABLE TO THE IPAD SYSTEM DESIGN
21	RISK, DEVELOPMENT	THE DEGREE TO WHICH THE SYSTEM CAN BE DEVELOPED AS CONCEIVED WITHIN THE TIME AND BUDGET ESTIMATE. (COMPLEXITY IS AN OBVIOUS RELATED PARAMETER).
22	RISK, OPERATIONAL	THE DEGREE TO WHICH THE SYSTEM WILL FUNCTION AS CONCEIVED WITHOUT EVIDENCE OF SHORTSIGHTEDNESS, INAPPLIC-ABILITY, OR BEING FAILURE PRONE.
23.	SKILL	TIME-AVERAGE SKILL LEVEL REQUIRED BY USER OF SYSTEM (ACCOUNTING FOR BOTH LEVEL AND DURATION OF USE BY THAT LEVEL); THE OBJECTIVE IS THE LOWEST SKILL LEVEL COMMENSURATE WITH THE TASK
24	STANDARDIZATION	DEGREE TO WHICH IPAD OPERATION IS STANDARD BETWEEN INPUT/OUTPUT DEVICES AND BETWEEN SUBSCRIBER INSTALLATIONS
25.	TOLERANCE	DEGREE TO WHICH SYSTEM IS TOLERANT OF USER ERRORS AND PROVIDES TUTORIALS TO CIRCUMVENT ERRORS WHICH HAVE BEEN MADE.
26	TRANSFERABILITY	DEGREE TO WHICH IPAD SOFTWARE CAN BE TRANSFERRED FROM ONE HARDWARE AND/OR SOFTWARE INSTALLATION TO ANOTHER WITHOUT REQUIRING MODIFICATION.
27.	UPKEEP	THE DEGREE TO WHICH SYSTEM MAINTAINENCE IS NOT REQUIRED
28.	UTILITY	A MEASURE OF MAXIMUM CAPABILITY FOR MINIMUM COST OVER THE ECONOMIC LIFE OF THE SYSTEM
29	VERSATILITY	THE APPLICABILITY OF THE IPAD SYSTEM TO DIFFERING PROCESSES.

3.2 The Quantifiable Objectives

Table 3-1 contains the definitions provided with the Survey card deck. It was recognized at the onset that the selected 29 quantifiable objectives were not independent nor did they cover all the significant attributes of the intended design. However, to even approach the problem of insuring independence of verbal definitions was so formidable a task (with questionable results considering each participant's individualistic interpretation) that an attempt was made only to eliminate obvious dependences by selection of the objective name and providing a careful definition.

More emphasis was placed upon completeness than on independence — e.g., although MODULARITY directly effects MAINTAINABILITY both objectives are important in their own right and were both included. Some subtleties were contained in the definitions to terms . . . . MAINTAINABILITY was distinguished from UPKEEP in that the former represented the ease of performing maintenance whereas the latter represented the requirement to do so. Time also entered to complicate the picture — although RATING is meant to measure the user's initial acceptance of the system (and hence effect the rate of spread of acceptance), LIFE is an obvious long term measure of user acceptance.

After considerable reassignment of objective names to concepts, combinations, deletions and rework of definitions, the 29 quantifiable objectives presented with their definitions were distributed to approximately one hundred current computer users at three General Dynamics facilities.

3.3 The Survey Analysis

To insure a high return of the survey packet, the objective names were put on computer cards for ease in ranking. Included were the definitions of these objectives (preceding page) which could be frequently referred to throughout the ranking process. Also included was the Introduction to the Statement of Work of the IPAD Contract to insure an adequate but unbiased statement regarding the objectives to be achieved by IPAD to familiarize the participant with the concept.

The participant was told the major objective of the survey ("to compile the information required to derive a linear objective function which can be used to evaluate the various IPAD design approach options"). He was instructed to rank the objectives ("which IPAD must meet to some degree") from most to least importance. It was suggested that he divide the objectives into three stacks ("HIGHLY IMPORTANT, NOT CERTAIN, LEAST IMPORTANT") and then rank and rerank each stack separately before recombining them into one stack. He was instructed to review the combined single stack and rework as applicable to finally achieve ranked order. When ranked he was instructed to insert the three "division" cards (supplied) "to indicate overall weighting of the ranking". An identification card was to be filled out identifying the respondent and his department (i.e., engineering discipline).

Of the one hundred (approximate) packets distributed, 86 were returned representing a spectrum of job classifications among engineering computer users. These packets were then input into a simple computer program to classify the respondents into 3 to 5 groups, as indicated on Figure 3-2. Although the technique resulted in considerable juggling of respondents among classes, the resulting classes - once formed - were quite stable and distinct. As indicated on the figure, a revision to the candidate insertion criterion was necessary to class almost all participants adequately.

- HOW MANY PARTICIPANTS WERE THERE?
  - 104 PARTICIPANTS FROM THREE GENERAL DYNAMICS FACILITIES
    86 FROM CONVAIR AEROSPACE, SAN DIEGO, CALIFORNIA
    16 FROM CONVAIR AEROSPACE, FORT WORTH, TEXAS
    2 FROM ELECTRO DYNAMICS, POMONA, CALIFORNIA
  - THE PARTICIPANTS REPRESENT A SPECTRUM OF JOB CLASSIFICATIONS
- HOW WERE THESE REDUCED?
  - THE DIVISION CARDS WERE USED TO OBTAIN A BEST LINEAR FIT OF A WEIGHT FUNCTION FOR THE RANKED OBJECTIVES FOR A GIVEN USER (OTHER FITS WERE ATTEMPTED BUT LINEAR GAVE THE MORE SATISFACTORY RESULTS).
  - SMALL GROUPS OF RESPONDENTS WERE FORMED WHO CLEARLY AGREED IN BOTH RANKING AND WEIGHTING.
  - THE PARTICIPANTS IN THE "POOL" WERE THEN TESTED FOR INSERTION INTO THE CLASS. IF THE PARTICIPANT'S RMS DEVIATION FROM THE CLASS MEAN (AFTER INSERTION) WAS LESS THAN 1.5% THEY WERE CONSIDERED FOR INSERTION. INSERTION WAS MADE INTO THE CLASS FOR WHICH THE RMS DEVIATION WAS MINIMUM IF LESS THAN 1.5%.
  - THIS TECHNIQUE RESULTED IN 3 STABLE CLASSES BUT HAD 47 REJECTIONS (IE PARTICIPANTS WHICH DID NOT FIT INTO ANY OF THE CLASSES).
  - IT WAS DISCOVERED THROUGH CLOSER INSPECTION THAT MANY PARTICIPANTS WERE BEING REJECTED NOT THROUGH THEIR ASSIGNMENT OF RANK BUT RATHER VARIATIONS IN WEIGHT. IT WAS DECIDED TO EVALUATE THE PARTICIPANTS RMS VALUE BY GIVING TO HIS RANKING THE GROUP'S WEIGHTING - - REASONING THAT RANK IS MUCH MORE IMPORTANT THAN WEIGHT. THIS RESULTED IN 3 STABLE CLASSES AND ONLY 5 REJECTIONS.
  - SUBSEQUENT DETERMINATION THAT BOARD DESIGNERS (WITH LITTLE OR NO COMPUTER EXPERIENCE) MIGHT NOT BE REPRESENTED RESULTED IN EXPANSION OF THE SURVEY. WITH 104 FINAL RESPONDENTS, ALL BUT TWO WERE SUCCESSFULLY CLASSIFIED.

Figure 3-2. IPAD Survey Analysis

It was noted that five respondents were rejected and could not be placed into any of the three classes. Two of the five were nearly accepted into the USER and MANAGERIAL groups. The remaining three were chronic problems throughout the grouping process. Closer inspection revealed that two of these three were responding alike and would constitute another PROGRAMMER group judging by their emphasis on computer programming standard practices. The third was the only respondent representing the missing DESIGNER group which — owing to their general lack of computer experience — were not adequately represented in the survey.

It was subsequently decided to seek out approximately 20 board designers with little or no computer experience for inclusion in the survey under the assumption that these potential users were not adequately represented. Of the 18 sampled, 16 fell into the USER class and two into the MANAGERIAL class so that all were classified. In addition, the two which had been nearly accepted (above) were also classed due to a slight alteration of the USER and MANAGERIAL groups. Further, the single, original "DESIGNER" respondent was also barely classed with the USER group. Thus, with 104 final respondents, all but 2 were classified.

3.4 The Analysis Results

The results when viewed only as a ranking do not appear substantially different, as noted in Table 3-2. All groups ranked DUPLICATION last. The "USER" group and "SYSTEM ANALYST" group both ranked ACCEPTABILITY first; the "MANAGERIAL" group ranked it seventh. Indeed, in examining the first five responses from each group it was noted that all three groups share RELEVANCY, CONTROL and EFFICIENCY, USER and two of three share ACCEPTABILITY, ADAPTABILITY, and TOLERANCE — a remarkable singlemindedness among a sample this large. Yet there are significant differences, e.g., VERSATILITY (6th, 23rd, 9th), TRANSFERABILITY (26th, 27th, 13th), TOLERANCE (13th, 5th, 4th), etc. Why not form a single class? Because it was recognized that such a single class would obscure differences that may be important during actual selection when in conflict. Further, no attempt was made to obtain a random sampling of respondents.

The names assigned to the groups were derived during analysis through knowledge of the participants initially forming the groups and from those participants most representing the group mean (lowest RMS error). By no means were all the respondents who were classed with a group necessarily affiliated with that group in real life. Sometimes the affiliation was easy to rationalize (e.g., John Doe's background working for many groups on many problems gives him a SYSTEM ANALYST background) whereas others were obscure (e.g., Jim Doe — a programmer recently out of school with a Bachelors degree in Mathematics — in the MANAGERIAL Class?).

Figure 3-3 presents the weight function plotted against the objectives in ranked order.

TABLE 3-2

SURVEY RESULTS, RANKED OBJECTIVES BY GROUP

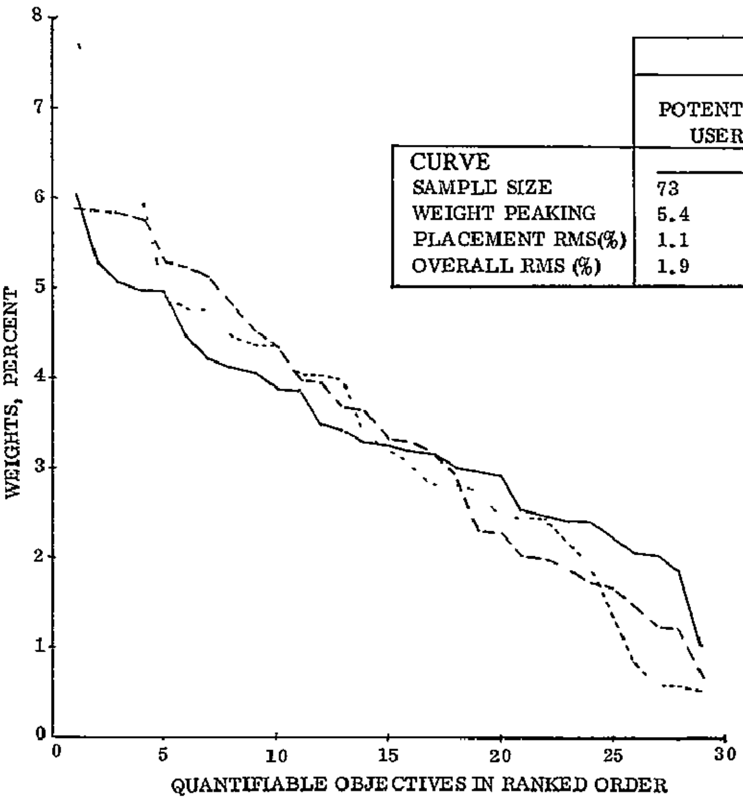
USER GROUP	SYSTEM ANALYST GROUP	MANAGERIAL GROUP
1. ACCEPTABILITY	1. ACCEPTABILITY	1. EFFICIENCY, USER
2. RELEVANCY	2. EFFICIENCY, USER	2. CONTROL
3. CONTROL	3. CONTROL	3. RELEVANCY
4. ADAPTABILITY	4. RELEVANCY	4. TOLERANCE
5. EFFICIENCY, USER	5. TOLERANCE	5. ADAPTABILITY
6. VERSATILITY	6. AUTOMATION	6. RESPONSE
7. RISK, OPERATIONAL	7. RISK, OPERATIONAL	7. ACCEPTABILITY
8. OPEN-ENDEDNESS	8. ADAPTABILITY	8. AUTO-TUTORIAL
9. EFFICIENCY, COMPUTER	9. OPEN-ENDEDNESS	9. VERSATILITY
10. EXPEDIENCY	10. RESPONSE	10. AUTOMATION
11. RESPONSE	11. AUTO-TUTORIAL	11. OPEN-ENDEDNESS
12. AUTOMATION	12. RATING	12. MODULARITY
13. TOLERANCE	13. EFFICIENCY, COMPUTER	13. TRANSFERABILITY
14. AUTO-TUTORIAL	14. EXPEDIENCY	14. SKILL
15. MAINTAINABILITY	15. LIFE	15. STANDARDIZATION
16. RISK, DEVELOPMENTAL	16. UTILITY	16. MAINTAINABILITY
17. UTILITY	17. SKILL	17. EFFICIENCY, COMPUTER
18. MODULARITY	18. RISK, DEVELOPMENTAL	18. RISK, OPERATIONAL
19. SKILL	19. MODULARITY	19. UPKEEP
20. INVESTMENT	20. INVESTMENT	20. CURRENCY
21. CURRENCY	21. MAINTAINABILITY	21. DEPENDENCY
22. RATING	22. DELAYS	22. RATING
23. LIFE	23. VERSATILITY	23. UTILITY
24. STANDARDIZATION	24. CURRENCY	24. LIFE
25. UPKEEP	25. UPKEEP	25. EXPEDIENCY
26. TRANSFERABILITY	26. DEPENDENCY	26. RISK, DEVELOPMENTAL
27. DELAYS	27. TRANSFERABILITY	27. INVESTMENT
28. DEPENDENCY	28. STANDARDIZATION	28. DELAYS
29. DUPLICATION	29. DUPLICATION	29. DUPLICATION

The inserted table on the figure gives the legend and statistics for each identified group. Weight Peaking is simply the maximum weight divided by the minimum weight and is a measure of the "intensity" of the ranking, or, in other words, the overall slope of the weight function. It is noted that the SYSTEM ANALYST had the largest peaking of 14.0 and the POTENTIAL USER the lowest at 5.4. To a certain extent this also reflects the size and mix of the group.

The Placement RMS was obtained from the revised criterion used to determine if a candidate should be placed in a class. It principally reflects rank variation in that it is the RMS error that the respondents had as a group when their individual weight function was replaced by the weight function of the class (the one plotted in the figure).

The Overall RMS was obtained directly from each user's weighted objectives as they were compared to the class weighted objectives (the mean of the class members). It reflects both rank and weight variations.

Sensitivity was measured by selecting two respondents to undergo the survey a second time approximately one week after their first attempt. Each when compared



	GROUP		
	POTENTIAL USER	SYSTEM ANALYST	MANAGERIAL
CURVE			
SAMPLE SIZE	73	15	14
WEIGHT PEAKING	5.4	14 0	8 2
PLACEMENT RMS(%)	1.1	1.2	1.2
OVERALL RMS (%)	1.9	1.6	1.6

Figure 3-3. IPAD Survey Weight Function

against himself formed a class with two entries with an Overall RMS of 0.54 and 0.56 or about 30% of the Overall RMS of the classes shown. This demonstrates that the classes formed were suitably compact.

3.5 Evaluating the Objective Function, an Example of Its Use

An example of the use of the Survey in evaluating various design approaches is shown in Figure 3-4. Each evaluator indicates his judgement as to the score the design approach would get for each of the twenty-nine quantifiable objectives. A score of 10 is maximum and 0 is minimum; that is, a score of 10 indicates that the objective is met fully and a score of 0 implies it was not even partially met. A score of 0 is also assigned across all design approaches whenever — in the evaluator's judgement — the objective does not pertain. In this way each evaluator quantifies the objectives for each design approach. He must be certain that the quantification is accurate in a relative sense (that is that each design approach option has been accurately evaluated relative to the other design approach options) and is also accurate in an absolute sense (that is an assignment of a value of 8.5 to one objective has the same meaning as an assignment of 8.5 to a different objective). For this reason the assignments are always made by row (by objective for each candidate design approach option) and,

EVALUATION OF SOFTWARE DESIGN OPTIONS
OBJECTIVE FUNCTION VALUES FOR EVALUATOR...HURLEY

	OPTION 1 VERSION -0	OPTION 2 VERSION -0	OPTION 3 VERSION 1	OPTION 3 VERSION 2
1. ACCEPTABILITY	4.00	5.00	6.00	8.00
2. ADAPTABILITY	5.00	6.00	7.00	7.50
3. AUTOMATION	0.00	0.00	0.00	0.00
4. AUTO-TUTORIAL	0.00	0.00	0.00	0.00
5. CONTROL	0.00	0.00	0.00	0.00
6. CURRENCY	1.00	5.00	10.00	9.00
7. DEPENDENCY	10.00	5.00	2.00	1.00
8. DELAYS	0.00	0.00	0.00	0.00
9. DUPLICATION	1.00	5.00	10.00	7.50
10. EFFICIENCY, COMPUTER	5.00	6.00	7.00	10.00
11. EFFICIENCY, USER	3.00	2.00	1.00	5.00
12. EXPEDIENCY	1.00	5.00	8.00	10.00
13. INVESTMENT	.50	7.00	9.00	10.00
14. LIFE	1.00	3.50	6.00	9.00
15. MAINTAINABILITY	3.00	7.50	10.00	9.00
16. MODULARITY	0.00	0.00	0.00	0.00
17. OPEN-ENDEDNESS	5.00	7.00	8.00	10.00
18. RATING	9.00	8.00	8.00	10.00
19. RELEVANCY	0.00	0.00	0.00	0.00
20. RESPONSE	9.00	0.00	8.00	10.00
21. RISK, DEVELOPMENTAL	2.00	7.00	9.00	10.00
22. RISK, OPERATIONAL	2.00	5.00	10.00	10.00
23. SKILL	0.00	0.00	0.00	0.00
24. STANDARDIZATION	10.00	3.00	8.00	5.00
25. TOLERANCE	0.00	0.00	0.00	0.00
26. TRANSFERABILITY	10.00	4.00	.50	2.00
27. UPKEEP	10.00	9.00	8.00	7.50
28. UTILITY	5.00	7.00	9.00	10.00
29. VERSATILITY	0.00	0.00	0.00	0.00
USER GROUP EVALUATION	3028.70	3947.19	4765.30	5532.92
SYSTEM GRP EVALUATION	2750.96	3681.72	4527.21	5519.81
MANAGEMENT EVALUATION	3162.83	3534.40	3974.43	4644.16

Figure 3-4. Example of Evaluation of the Objective Function

when complete, checked by column (by candidate design approach option to be assured that equal scores reflect equal emphasis).

The figure of merit tabulated at the bottom of Figure 3-4 reflects the evaluation each group would have given the design approach options were they to have achieved the same intimate understandings of the design approach options and their subleties as the evaluator. In this example all three groups are unamimous in their selection of Option 3, Version 2, which corresponds to the sub-subsystem level design shown in Figure 2-12. Although different evaluators can be expected to differ in assignment of scores while still being in basic agreement, it is unlikely that their selection would differ.

4 OM QUESTIONNAIRE

Early in the IPAD study it became apparent that a set of design requirements representing the engineering/design process was needed for the design of IPAD. To supply the needed design requirements, a Questionnaire was developed to provide statistical data for a typical cross section of candidate OMs for an IPAD system. This section describes the Questionnaire and computer analysis, and presents some typical results.

4.1 Objectives

The specific objectives of the Questionnaire were to determine - as a minimum - the following for IPAD:

1 Minimum host computer hardware configuration required to support IPAD, e.g..

   a CPU size and speed.

   b. Disk size and speed

   c. Peripherals.

   d. Terminals, e.g., CDC 274, Tektronix 4010, etc.

   e. Mini-computers and their use.

2. Minimum host computer operating system required to support IPAD, e.g.:

   a. CDC's KRONOS time sharing system or its equivalent.

   b. CDC's SCOPE 3.2, 3.3 or 3.4 or its equivalent.

   c. CDC's INTERCOM 2.0, 3.0, or 4.0 or its equivalent.

3. IPAD specific requirements, e.g.:

   a. Utilities to be included in IPAD system.

   b. System structure.

   c. Input/output requirements.

   d. Data base structure.

In developing the Questionnaire it was felt that the responses to the questions asked would be sufficient to develop the host computer (hardware and software) configuration and IPAD software design requirements.

4.2 The Questionnaire

Figure 4-1 completely characterizes the OM Questionnaire and the respondents. The actual Questionnaire form is presented in Figure 4-2.

WHAT IS IT?
- 11 BY 17-INCH FORM TO BE COMPLETED FOR "REPRESENTATIVE" CANDIDATE PROGRAMS THAT WOULD (MIGHT) BECOME IPAD OMs
- EACH RESPONSE WAS QUANTIFIABLE AND/OR
  - OBJECTIVE (e g, CENTRAL PROCESSOR TIME & MEMORY REQUIREMENTS)
  - SUBJECTIVE (e g, TYPICAL NUMBER VARIABLES REVIEWED BEFORE EACH RUN)
  - OPINION (e g, WOULD YOU HAVE USE OF AN OPTIMIZER UTILITY?)
- TO ENSURE UNIFORM RESPONSE, EACH FORM WAS ACCOMPANIED BY RATHER DETAILED INSTRUCTION SHEET

WHAT IS ITS INTENDED USE?
- TO STATISTICALLY TYPIFY A REPRESENTATIVE OM COLLECTION
- TO OBTAIN SPECIFIC DATA RELATING TO COMPUTER REQUIREMENTS (e g, CENTRAL MEMORY SIZE & MASS STORE REQUIREMENTS)
- TO INFER (COMPUTE FROM DATA AVAILABLE) ADDITIONAL INFORMATION NOT CURRENTLY AVAILABLE (e g, IF JOB BECAME INTERACTIVE, WHAT TYPE OF INTERACTIVE TERMINAL WOULD IT REQUIRE?)

WHO WERE THE RESPONDENTS?
- 93 CANDIDATE PROGRAMS SELECTED FROM FOLLOWING AREAS
  - ABSTRACT ANALYSIS (60) – STRUCTURAL DYNAMICS, STABILITY & FLIGHT CONTROLS, THERMODYNAMICS, AERODYNAMICS, FLIGHT MECHANICS
  - DESIGN ANALYSIS (13) – PROPULSION, MASS PROPERTIES, STRUCTURAL ANALYSIS
  - DESIGN (6) – MECHANISMS DESIGN, ELECTRONIC (CIRCUIT) DESIGN, DRAFTING
  - MANUFACTURING (2) – CIRCUIT BOARD PACKAGING, AUTOMATIC PROGRAMMED TOOLS (APT)
  - MISCELLANEOUS (11) – EMPIRICAL DATA REDUCTION, MANAGEMENT, MARKETING
- THESE WERE EXISTING PROGRAMS THAT PRODUCED BIAS TOWARD MORE COMPUTERIZED AREAS (i e, ABSTRACT ANALYSIS)
- BALANCE IN COMPLEXITY WAS ATTEMPTED TO ENSURE REPRESENTATIVE RESULTS
  - MOST COMPLEX – NASTRAN, TRAJEX
  - LEAST COMPLEX – PROGRAM THAT REDUCED THESE QUESTIONNAIRES

Figure 4-1. IPAD OM Questionnaire

The 13-page set of instructions accompanying the form (Reference 10) contained:

1. General instructions for filling out the form.
2. A collection of topic titles to categorize the program usage by code (question line 2.01 of Questionnaire - see Figure 4-2).
3. Code number of computers/operating systems on which the program (in its present form) has been known to run (question line 2.06).
4. Code number of interactive devices if program is interactive (question line 2.08). If the program had been used in an interactive mode, the respondent was requested to fill out the Questionnaire reflecting (principally) the interactive mode of operation.
5. Identification of general purpose programs for which the Programming Data portion of the Questionnaire (Part 3) need not be filled out (since an individual had already been selected to complete that part):

   a. NASTRAN -- NAsa STRuctural ANalysis, a general program for computing the static and dynamic loads on complex structures.

0. 01 DATE ______________ YOUR EXTENSION ________
02. NAME ______________________________
03. EMPLOYEE NO ____________ DEPT. NO. ________

IPAD OPERATING MODULE (OM) QUESTIONAIRE
(CONVAIR AEROSPACE/SAN DIEGO OPERATION FORM)

DO NOT FILL IN
LOG NO. ______

1. PROGRAM IDENTIFICATION DATA
01 IS THIS PROGRAM EXISTING? ___ IN DEVELOPMENT? ___ OR PROPOSED? ___ (CHECK ONE) PROGRAM IDENTIFIER (BOTH NAME AND NUMBER PREFERRED) ____________
02. PROGRAM TITLE ________________________________
03. FUNCTIONAL DESCRIPTION (WHAT IT DOES OR IS USED FOR) ________________________________

2. PROGRAM USAGE DATA
01. TYPE OF USE (SEE SHEET FOR CODE) ____________ DO YOU REQUIRE SPECIAL OUTPUT FORMS WHEN LISTING? YES ___ NO___ (IF YES, FORM NO.________ ____________)
02. EXTENT OF PROGRAM USE (NUMBER OF CASES) DURING CONCEPTUAL _____, PREDESIGN _____, AND DETAILED DESIGN __________ PHASES OF AEROSPACE DESIGN PROCESS.
03. DAY FILE SUMMARY (TYPICAL RUN OF ___ CASES) CM REQUIRED FOR LOADING ________ CM REQUIRED FOR EXECUTION ________ CARDS READ ____ TAPES MOUNTED ___ CP TIME (SEC) ___ MONITOR REQUESTS ____ MEMORY UNITS ___ PP TIME (SEC) ___ REAL TIME (SEC) ___ IGS CONSOLE TIME (MIN) ___ LINES PRINTED ____ CARDS PUNCHED ____
04 CASES PER COMPUTER RUN (TYPICAL) ________ TURNAROUND PER COMPUTER RUN (TYPICAL) OVERNIGHT? YES ___ NO ___ AND/OR ________ HOURS WHEN (DATE) LAST USED ________
05. SKILL LEVEL OF USERS OF PROGRAM OUTPUT (IN % OF USE) AIDE ____ ASSISTANT ____ ENGINEER ____ SR. ENG. ____ DESIGN SPEC ____ STAFF SCIENTIST ____ SUPERVISION ____ MANAGEMENT ____
06. MODE OF USE (IN % OF TOTAL USE) BATCH ____ INTERACTIVE ____ COMPUTERS/OPERATING SYSTEMS THIS (OR A VERY SIMILAR) PROGRAM HAS SUCCESSFULLY RUN ON (SEE SHEET FOR CODE) (__________)
07. AMOUNT OF INFORMATION USER REGULARLY REVIEWS AS A PART OF EACH RUN (FILL IN AS MANY BLANKS AS MIGHT APPLY) (__________)
ON INPUT ____ WORDS ____ "CARDS" ____ LINES (TEXT) ____ GRAPHS ____ VARIABLES ____ PICTURES (CHECK: 2D? ____ AND/OR 3D? ____)
ON OUTPUT ____ WORDS ____ "CARDS" ____ LINES (TEXT) ____ GRAPHS ____ VARIABLES ____ PICTURES (CHECK: 2D? ____ AND/OR 3D? ____)
DURING EXECUTION (IF INTERACTIVE) ____ WORDS ____ "CARDS" ____ LINES (TEXT) ____ GRAPHS ____ VARIABLES ____ PICTURES (CHECK: 2D? ____ AND/OR 3D? ____)
08. INTERACTIVE DEVICE(S) USED IF APPLICABLE (SEE SHEET FOR CODE) ____________. IF INTERACTIVE, DO YOU REQUIRE HARD COPY? YES ____ NO ____ (IMMEDIATELY? YES ____ NO___).
09. AMOUNT (IN %) OF INPUT WHICH COMES DIRECTLY OR INDIRECTLY FROM OTHER COMPUTER RUNS? ________. AMOUNT (IN %) OUTPUT FEEDING OTHER RUNS? ____________.
10 WOULD THIS PROGRAM BE MORE USEFUL TO YOU IF IT WERE INTERACTIVE? YES ___ NO___, HAD INTERACTIVE MONITORING DURING EXECUTION? YES ____ NO ____.
11. WOULD GRAPHICAL? (YES _____ NO _____) OR PICTORIAL? (YES____ NO ____) INPUT OR OUTPUT VIEWING BE HELPFUL IN USING THIS PROGRAM?

3. PROGRAMMING DATA (NOTE IF THIS IS ONE OF THE GENERAL PURPOSE PROGRAMS LISTED ON YOUR INSTRUCTION SHEET, THIS PART HAS ALREADY BEEN COMPLETED BY OTHERS).
01. WHEN (DATE) WAS THIS PROGRAM FIRST DEVELOPED? ________ WHEN (DATE) WAS THIS PROGRAM LAST MODIFIED? ________ DEVELOPING AGENCY(S) (ABBREVIATE) ____________
02. COGNIZANT PROGRAMMER'S NAME ____________ AND EXTENSION ________. COGNIZANT ENGINEER'S NAME ____________ EXT. ________.
03. PROGRAM DOCUMENTATION (IF ANY GIVE REPORT/MEMO TITLE, NUMBER AND DATE) ____________
04. WOULD YOU DESCRIBE THE PROGRAM AS MULTIDISCIPLINARY? YES ___ NO ___ GENERAL PURPOSE? YES ___ NO ___ SELF CONTAINED? YES ___ NO ___
A PROGRAMMING "SYSTEM"? YES ___ NO ___ (WHAT PROGRAMS, PREFERRABLY BY NUMBER, BELONG TO THIS SYSTEM? ____________)
A MEMBER OF A PROGRAMMING SYSTEM? YES ___ NO ___ (WHAT SYSTEM BY NAME AND/OR NUMBER? ____________)
05. PROGRAMMING LANGUAGES IN USE (IN % OF USE) FORTRAN IV ____ COBOL ____ BASIC ____ APT ____ COMPASS ____ OTHER? (PLEASE SPECIFY) ____________.
06. WAS THIS PROGRAM GENERATED BY MIDAS? YES ___ NO ___ CSMP? YES ___ NO ___ OTHER PRECOMPILER/TRANSLATOR? (PLEASE SPECIFY) ____________
07. AMOUNT OF FORTRAN CODE (IN % OF TOTAL FORTRAN CODE) DEVOTED TO CALLS TO IGS ROUTINES ________ SC4020 ROUTINES ________ CALCOMP ROUTINES ________
08. AMOUNT OF REMAINING FORTRAN CODE (IN % OF TOTAL FORTRAN CODE IGNORING THAT REPORTED DIRECTLY ABOVE) THAT IS SPECIFICALLY MACHINE DEPENDENT ____. (EXPLAIN ____________).
09. AMOUNT OF FORTRAN CODE (IN % OF TOTAL FORTRAN CODE) THAT IS IN DOUBLE PRECISION ________ ESTIMATE OF TRANSFERRABILITY TO OTHER COMPUTERS HIGH _____ MED ____ LOW _____.
10 IS THIS PROGRAM OVERLAYED OR SEGMENTED? YES ___ NO ___. IF YES, PLEASE ATTACH SHEET SHOWING STRUCTURE/DESCRIPTION (SEE ATTACHED EXAMPLES)

4. FILE DATA FILL OUT DATA ON ALL FILES IN USE BY THE PROGRAM. BE SURE TO INDICATE ADDITIONAL (EG PUNCH OR SCRATCH) FILES INDICATED ON THE PROGRAM CARD, DATA MGR. FILES, AND PROG. FILES/TAPES

	FILE FUNCTION	FILE NAME	FILE MEDIUM	FILE TYPE	FILE STRUCTURE	TYPICAL NUMBER* OF COMPUTER WORDS IN THIS FILE	TO BE SAVED FOR OTHERS	TO BE SAVED FOR USER	TYPICAL NUMBER* OF VARIABLES	GIVE BRIEF FUNCTIONAL DESCRIPTION OF FILE CONTENTS
01.	INPUT	TAPE5	C	C	S					
02.	OUTPUT	TAPE6	L	C	S					
03.										
04										
05.										
06										
07										
08.										

FILE FUNCTION: INPUT, OUTPUT, SCRATCH, PUNCH, PLOT, RESTART, RECOVERY, PROGRAM, ETC.
FILE MEDIUM: D ~ DISK, T ~ MAGNETIC TAPE, P ~ PAPER TAPE, C ~ CARDS, L ~ LISTING
FILE TYPE: C ~ CODED, B ~ BINARY, M ~ MIXED
FILE STRUCTURE: S ~ SEQUENTIAL, R ~ RANDOM, D ~ DATA MANAGER, I ~ INDEX SEQUENTIAL, M ~ MIXTURE
TO BE SAVED FOR OTHERS: eg, DESIRED IN COMMON DATA BANK (FOR USE BY MORE THAN ONE GROUP)
TO BE SAVED FOR USER: eg, TO BE RETAINED IN A USER FILE FOR SUBSEQUENT REFERRAL

CHECK BOX IF COMMENTS ON BACK ☐

* APPEND Ø FOR OCTAL AND D FOR DECIMAL AS APPLICABLE - IF NOT NOTED, DECIMAL (D) IS ASSUMED.

Figure 4-2. IPAD OM Questionnaire Form

b. APT - Automatically Programmed Tools, a translator for describing the tool path to follow when machining parts.

c. APT/IGS - an Interactive Graphics version of APT.

d. MIDAS - a general purpose simulation precompiler which produces FORTRAN source programs in response to an input model network.

e. NAMESIM- a general purpose interactive graphics programming system for the CDC 274 graphics terminal; originally designed for continuous-variable simulation programs employing FORTRAN NAMELIST input. (NAMESIM is described in Subsection 2.2.1.)

f. CLAAS - Convair's Linear Analysis of Aircraft Structures, a finite element program for stress analysis (static) of three-dimensional structures made up of panels and bars.

g. IS&R - Information Storage and Retrieval, a general purpose program for building, searching and displaying of information in large data files.

h. TRAJEX - TRAJectory EXecutive, a programming system which produces flight simulations of multistage aerospace vehicles in the gravitational field of a rotating central body.

i. VOODOO - Volumes Of Original Data Organized and Output, a general purpose plotting program for the SC-4020 microfilm recorder

j. DYNES - DYNamic Engineering System, a programming system for controls and dynamics analyses of launch vehicle related control systems. (DYNES is described in Subsection 2.3.3.1.)

k. CSMP - Continuous System Modeling Program, a general purpose simulation precompiler similar to MIDAS.

l. DYNAMO - DYNamics Assembler and MOdeling system, a programming system for structural dynamics analyses of structures represented by one-dimensional lumped-parameter models.

6. Conversion formulae (for the convenience of respondent) to provide ready conversion to:

   a. Cards, e.g., from boxes or inches (stack).

   b. Words, e.g., from cards (and density), lines (and density), or disk sectors.

   c. Lines, e.g., from sheets or inches of listing (and density).

   d. Graphs, e.g., from microfilm length or microfilm roll ID and OD, or from Copyflo stack (inches).

7. Examples of completed Questionnaires for:

   a. NAMESIM (see Item 5e above).

   b. A satellite (OV-1) ejection analysis simulation (generated by MIDAS - see Item 5d above).

The general instructions for filling out the form detailed that Part 2 (Figure 4-2) should be obtained from the "day file" summary at end of any typical computer run. Also the terminology "cases" should be distinguished from "runs" in that several cases typically make up a computer run.

It was further instructed that Part 3 should be completed by the programmer or from programmer documentation. A programming "system" (question line 3.04 of Figure 4-2) was characterized as a collection of distinct, self-contained programs that had been interconnected to run as a unit. The respondent was to distinguish between such a system and a member (i.e., individual program) of such a system.

Part 4 was to be completed jointly by the programmer and user since it contained file details as well as application details. The respondent was instructed that data to be "saved" (columns 7 and 8 of Part 4) need only consist of that data necessary for subsequent viewing or processing by the user ("saved for user") and then discarded, as opposed to data for general consumption ("saved for others") as that put in a general data bank.

4.3 Questionnaire Analysis

Figure 4-3 is a compact summary of the analysis accomplished on the completed Questionnaires. Figure 4-4 lists the 150 individual data items available for subsequent analysis.

Not all of the first 147 data items (Figure 4-4) were obtained directly off the completed Questionnaire forms as implied by Figure 4-3. Specifically, the use of the proposed general purpose utilities to be provided by the IPAD system (data items 75 through 89 of Figure 4-4) were inferred from discussions with the individual respondents and from answers to related questions on the Questionnaire. Also data items 7 through 14 (Figure 4-4) were normalized on a per-case basis by dividing the Questionnaire's results, question line 2.03 of Figure 4-2, by the number of cases specified for that run (first item of question line 2.03); the number of cases utilized for normalization was not saved as a data item. Finally, various data items were changed in units, e.g., Central Memory requirement (data item 6) was changed from octal to decimal.

C-2

Each completed Questionnaire was subjected to extensive pre-reduction checks to catch any obvious misinterpretations or extreme estimates. Over 80 percent of the respondents were re-contacted and questioned regarding the more subjective estimates, e.g., "Extent of Program Use" (question line 2.02 of Figure 4-2). These checks provided an independent analysis of each response (correlated with related responses) before these were lost in the maze of statistics to follow. Approximately 40 percent of the completed Questionnaires required some corrections.

As noted in Figure 4-3, three data items (148 through 150 of Figure 4-4) were estimated from Questionnaire data available. The following subsections detail these estimates.

HOW WERE THE QUESTIONNAIRES REDUCED?

- 145 DISTINCT DATA ITEMS (NUMBERS)-WERE OBTAINED DIRECTLY FROM QUESTIONNAIRE
- 2 NON-NUMERIC ITEMS WERE RETAINED FOR IDENTITY & SORTING
- 3 REQUIRED DATA ITEMS WERE INFERED FROM DATA AVAILABLE
  - TYPE OF INTERACTIVE TERMINAL (RANGED FROM 0 THROUGH 5)
  - TERMINAL TIME PER (TYPICAL) SITTING (MINUTES)
  - NUMBER OF CASES ACCOMPLISHED PER TERMINAL SITTING

HOW WAS DATA USED?

- SOME RESULTS WERE USED STATISTICALLY (e g, CENTRAL MEMORY REQUIREMENTS HISTOGRAM TO HELP SIZE COMPUTER)
- SOME RESULTS WERE SUMMED (e g, NUMBER OF USERS OF OPTIMIZER UTILITY-ANSWER IS COMPOSED OF RESPONSE FROM "USE OPTIMIZER?", "USE SENSITIVITY EXTRACTOR?" & "USE PARAMETERIZER?")
- SOME RESULTS WERE OBTAINED BY COMBINING DATA ITEMS (e g, ESTIMATED TERMINAL HOURS BY ENGINEERING SKILL LEVEL (FOR AIRCRAFT DESIGN) WAS OBTAINED FROM EQUATION

HOURS = (TERMINAL TIME/60) * (NO OF CASES PER PHASE)
/ (NO OF CASES PER TERMINAL SITTING)
* (% USED BY SKILL LEVEL)/100

WHERE

- PHASE REFERS TO CONCEPTUAL, PREDESIGN & DETAILED DESIGN
- SKILL LEVEL REFERS TO SUPERVISION, . , SENIOR ENGINEER, . , AIDE

HOW DO WE KNOW THESE ARE TRULY REPRESENTATIVE?

- SIZE OF RESPONSE (93 INDIVIDUAL QUESTIONNAIRES)
- EXTENSIVE CHECKS (DATA COMPARISON WITH OUR RECENT PHASE B SPACE SHUTTLE BOOSTER PREDESIGN, CURRENCY, CHECKS, ETC)
- INTERACTIVE DIGITAL REDUCTION (ELIMINATES ERRORS & PROVIDES MORE COMPLETE RESULTS)

Figure 4-3. IPAD OM Questionnaire Analysis

4.3.1 <u>Type of interactive terminal (data item 148)</u>. - It was necessary to project the interactive terminal usage based on the OM Questionnaire responses even though only 20 of the 93 programs included were in fact interactive.

A wide range of interactive computer capability exists in today's market. The display requirements for engineering users range from simple alphanumeric displays

1. (Log Number)
2. (Program Number)
3. No. Cases, Conceptual
4. No. Cases, Pre-Design
5. No. Cases, Design
6. CM_{10} Requirements
7. Cards Read
8. CP Time, Sec.
9. Monitor Requests
10. PP Time, Sec.
11. Real Time, Sec.
12. Console Time, Min.
13. Lines Printed
14. Cards Punched

PER CASE (items 7–14)

15. Tapes Mounted
16. Typical No. Cases
17. Turnaround, Min.
18. Days Since Last Used
19. % Use by Aide
20. % Use by Assistant
21. % Use by Engineer
22. % Use by Sr. Engr.
23. % Use by Design Spec.
24. % Use by Staff Scien
25. % Use by Supervision
26. % Use by Management
27. % Batch Use
28. % Interactive Use
29. % FORTRAN IV
30. % COBOL
31. % BASIC
32. % ALGOL
33. % COMPASS
34. % Double Precision
35. * Transferability
36. Overlayed ?
37. Days Since Origin
38. Input Words Scanned
39. Output Words Scanned
40. Words Interactive
41. Input "Cards" Scanned
42. Output "Cards" Scanned
43. "Cards" Interactive
44. Input Lines Scanned
45. Output Lines Scanned
46. Lines Interactive
47. Input Graphs Scanned
48. Output Graphs Scanned
49. Graphs Interactive
50. Input Variables Scanned
51. Output Variables Scanned
52. Variables Interactive
53. Input Pictures Scanned
54. Output Pictures Scanned
55. Pictures Interactive
56. Input Picture Dimension
57. Output Picture Dimension
58. Picture Dimension Interactive
59. % Input From Programs
60. % Output to Programs.
61. Desire Interactive?
62. Interactive During?
63. Need Graph. Plotter?
64. Need Pictorial Display?
65. Prog. Multidisciplinary?
66. Prog. General Purpose?
67. Prog. Self-Contained?
68. Prog. a System?
69. System Member?
70. Programmed by MIDAS?
71. Programmed by CSMP?
72. Programmed by Other?
73. (Discipline Code)
74. Days Since Modified
75. Use Statistical Pkg ?
76. Use File Manager?
77. Use Graphic Plotter?
78. Use Pictorial Display?
79. Use Generalized Fitter?
80. Use Topological Input?
81. Use Text Editor ?
82. Use Optimizer?
83. Use Sensitivity Extractor
84. Use Data Checker?
85. Use Tutorial Aids?
86. Use Drafting Module?
87. Use Parameterizer?
88. Use Compilers?
89. Use Movie Sequences?
90. Interactive Device
91. Need Hard Copy?
92. Hard Copy Immediately?
93. No. Files in Job
94. No Files Input
95. No. Files Output
96. No. Files Scratch
97. No. Files Punch
98. No Files Plot
99. No Files Restart
100. No Files Recovery
101. No Files Program
102. No. Files Disk
103. No Files Tape
104. No Files Paper Tape
105. No Files Cards
106. No Files Listings
107. No Files Microfilm
108. No Files Coded
109. No Files Binary
110. No. Files Mixture
111. No. Files Sequential
112. No. Files Random
113. No Files Data Mtgr
114. No. Files Index Seq.
115. No Files Mixture
116. No. Words Input
117. No. Words Output
118. No Words Scratch
119. No. Words Punch
120. No. Words Plot
121. No. Words Restart
122. No. Words Recovery
123. No Words Program
124. Saved for Others, Input
125. Saved for Others, Output
126. Saved for Others, Scratch
127. Saved for Others, Punch
128. Saved for Others, Plot
129. Saved for Others, Restart
130. Saved for Others, Recovery
131. Saved for Others, Program
132. Saved for User, Input
133. Saved for User, Output
134. Saved for User Scratch
135. Saved for User, Punch
136. Saved for User, Plot
137. Saved for User, Restart
138. Saved for User, Recovery
139. Saved for User, Program
140. No. Variables, Input
141. No. Variables, Output
142. No. Variables, Scratch
143. No Variables, Punch
144. No. Variables, Plot
145. No. Variables, Restart
146. No Variables, Recovery
147. No Variables, Program
148. **Type Interactive Terminal
149. **Console Time, Min.
150. **Number Interactive Cases

** - Estimated

* 1 = High, 2 = Med., 3 = Low.

Figure 4-4. Data Items Available From OM Questionnaires for text editing, program debug, etc., to integrated displays of bar charts, histogram and complex multidimensional shapes that must allow for topological manipulation. Economic necessity calls for a corresponding range of terminal hardware. Practical considerations of maintenance, computer interfacing and programming, dictate as small a variety of terminals as possible.

To facilitate matching the engineering and design needs with an economically appropriate capability, five functional terminal types were identified, and are shown in Figure 4-5. These are arranged according to capability (and price). Note that each device type can do all of the functions of any device of a lower type number. Some terminals are designed in modular units (e.g., IMLAC) so as to fit into several types according to which hardware options are selected. Note that the typical lease costs vary by two orders of magnitude.

TYPE	PHYSICAL DEVICE	A/N KEYBOARD	TOPOLOGICAL INPUT	VECTOR OUTPUT	USE OF MINICOMPUTER	TYPICAL COST ($/MO)	EXAMPLES
1	ALPHANUMERIC ELECTROMECHANICAL TYPEWRITER	YES	NO	NO	NO	50 to 100	TTY NCR 260 IBM 2740 TI 725
2	ALPHANUMERIC CRT- REFRESHED OR TV RASTER	YES	NO	NO	NO	50 to 120	CDC 211 COMPUTEK 200 IBM 3270
3	VECTOR DRAWING CRT - REFRESHED, DVST OR TV RASTER	YES	MINIMAL	YES	NO	150 to 450	TEKTRONIX 4010 COMPUTEK 300 IMLAC PDS-1
4	VECTOR DRAWING CRT - REFRESHED, or DVST	YES	GOOD	YES	SOME	500 to 2,000	TEKTRONIX 4002A COMPUTEK 400 IMLAC PDS-1 CDC 240 VECTOR GENERAL DD 2
5	VECTOR DRAWING CRT - REFRESHED	YES	EXCELLENT	YES	ALMOST ALWAYS	2,000 to 6,500	CDC 274 CDC GPGT IBM 2250 VECTOR GENERAL 2DR3/3D3 LUNDY SYS 32

Figure 4-5. Interactive Terminal Classifications

For IPAD purposes, the interactive terminal systems to be supported may be characterized by display capability (amount of data and resolution), type of user interaction, data communication rates, the communication media, required signal converters, and the degree of portability. The discussion which follows classifies interacti terminals into distinct types as envisioned for IPAD support. These terminal types and Figure 4-5 are described more completely in the subsections which follow (Representative terminal hardware is enumerated and is discussed in Section 5.3.)

The reader is referred to Chapter 2 of Reference 11 for an excellent discussion of the general characteristics of interactive graphics systems and illustrations of the various devices being discussed.

4.3.1.1 Display capability: There are essentially two distinct display capabilities available, the typewriter terminal displays (Type 1 in Figure 4-5) are essentially all alike in that they typically produce an 80 character line of type at six lines per inch on an 8.5 inch wide continuous sheet of paper (from a roll). Most use an electromechanical typewriter, however, the NCR 260 and Texas Instruments portable TI 725 employ a solid-state silicon thermal print head that develops characters on special thermal sensitive paper thus eliminating impact printing and associated noise.

The CRT display screen sizes range from 7.5 x 5.6 inches to large screen projected displays, about 7 x 8 feet. There are three basic commercially available types of display CRTs which encompass Types 2 through 5 of the figure. (Plasma displays and other discrete element displays were not considered in this study.)

The refreshed CRT redraws the image periodically (usually 30-40 times per sec.) and is the most common type of display in use. The display commands may be stored in a special display memory, computer central memory or on a fast drum. These CRTs may display only alphanumeric data (using extrusion forms, dot matrix or a stroke method) - as with Type 2 terminals - or may be able to also do vector (line) drawing - as with Types 3 through 5. The amount of data displayed is a function of the screen size, the refresh speed and display buffer size, whether hardware is used for character operation and whether absolute and incremental vector drawing is used. Resolution is invariably good, especially the Vector drawing CRTs which typically employ a 1024 × 1024 addressable point raster.

The direct view storage tube (DVST) CRT makes use of a special long lasting (retentive) phosphor on the CRT and special circuitry. It doesn't need to be refreshed to prevent flicker. Technical problems have so far limited this screen size to about 8 x 6 inches. To change any item displayed, the whole image must be erased and redisplayed. Resolution is good - usually a 1024 x 1024 raster count; about 3300 characters can be displayed.

Television raster CRTs make use of a constant raster scan and modulate the intensity or brightness to present data. With a normal 512 line TV raster, about 1000 characters can be displayed and vectors can be drawn on a 256 x 256 line grid. Resolution can be improved by going to a 940 line TV raster but this is more expensive.

Tonal displays (movies, etc.) can be easily shown on this type of display. Screen size ranges from 12-inch diagonal up to 27 inches diagonal.

4.3.1.2 User interaction: - All the terminals allow for alphanumeric keyboard input by the user. Type 4 and 5 systems have available some mechanisms for distinguishing and selecting elements that are being displayed (termed "Topological I/O"). Most of these selection mechanisms will return only the screen coordinates to the calling program and therefore require system software to determine the closest item to the spot that was selected. Devices that return the screen coordinates include the cross-hair cursor driven by a joy stick, track ball, mouse, analog tablet, or thumbwheel.

Using the light pen as the user interaction device allows easy hardware implementation of display item identification in addition to returning screen coordinates. Note that the light pen is a feasible device only on refreshed CRT displays.

4.3.1.3 Data rates: Terminal data communication rates vary from a low of 10 characters per second for the TTY Type 1 device to 5000 characters per second for typical Type 5 devices on a direct channel (computer) connection. Most Type 1 devices operate up to 30 characters per second maximum.

4.3.1.4 Communications media: Most Type 1 through 4 devices are setup to operate through standard voice grade dial-up telephone lines. This is a great enhancement for portable or semi-portable devices since they can be operated from any place within the plant - or indeed from home or across country - by merely acoustically coupling into the inplant and, as required, public telephone (switched network) systems. However practical consideration often limit systems using standard voice grade telephone lines to 30 characters per second (300 baud).

Private line service, or dedicated lines, are tariffed in accordance with line conditioning employed. (The reader is referred to Reference 12 for a concise introduction to the problems involved.) For example, a C2 conditioned line has been used up to 7600 baud "with no real evidence of data loss" (Reference 13). Although theoretically only good to 1200 baud, the direct dialing network has been used up to 2000 baud and is "generally good; if you have a bad circuit, redialing generally solves the problem" (op. cit.).

The real problems lie with inplant lines where calls originate and terminate (unless the computer at the terminal end is an outside commercial computer network). "Company lines are generally poorer than public lines" (op. cit.), presumably due to use of less expensive, older equipment and less preventative maintenance. This is especially true for companies employing operator switchboards.

The technical problems involved, which vary with transmission frequency, are deviations in delay distortion and db loss. Conditioned lines are actually private dedicated lines with point-to-point service and conditioning (e.g., delay distortion and gain matching) at every node. Even these private circuits can become troublesome, but the problems generally disappear after rework.

At 300 baud (30 characters a second), standard voice grade lines are highly reliable. At 3000 baud and above leased lines usually are required. Between these rates, voice grade dial-up lines can produce problems; however, when difficulties are isolated they often originate within the company complex.

4.3.1.5 Modems: Modems (short for modulator-demodulators) are equipment on the terminal ends of communication lines that convert the binary dc signal (the digital bit stream) into a form suitable for transmission over the communication lines. Modems are also called data sets or, when leased from Bell System, by the trade name Dataphones. Reference 12 also presents a concise overview of modem equipment.

Modern modems are more than just signal converters, often employing signal conditioning, error detection/correction and even the ability to adapt to changing line conditions. The cost for modems for the higher transmission rate (e.g. 9600 baud) can be a factor of ten higher than typical modems in the voice-grade range (op. cit.). These costs can approach the costs of the terminals themselves and are a factor to consider when determining realistic terminal use.

4.3.1.6 Portability: Type 1 devices are usually small and lightweight, contain a built-in acoustic coupler/modem, and can be hand-carried about the company as required. The TI 725 (Figure 4-5), in particular, weighs 35 pounds in its luggage type carrying case.

Type 3 through 4 CRT terminals could only be considered semi-portable and must be wheeled or transported from place to place. Type 5 terminals are typically quite large, heavy, and packaged in several racks, and are intended as semi-permanent installations. All but Type 5 devices typically operate on standard 120 VAC service in an ambient work-area environment. Type 5 devices typically require 220 VAC 3 phase service and air conditioning.

4.3.1.7 Estimating the device type: The type of interactive terminal was deduced from the OM Questionnaire responses as follows:

1. The device type was at least Type 1 (alphanumeric electromechanical typewriter) if a yes response was accorded any of these questions:
   a. Would this program be more useful to you if it were interactive (data item 61 of Figure 4-4)?

b. Would you use a general interactive statistical utility package (data item 75)?

c. Would you use a general interactive file manager (data item 76)?

d. Would you use a general text editor (data item 81)?

e. Would this program benefit significantly if it were accompanied by a set of tutorial aids (data item 85)?

2. The device type was upgraded to Type 2 (same as Type 1 except with CRT alphanumeric output and higher output rate) if any one of the following conditions were met concerning the amount of information the user regularly reviewed as a part of each case:

   a. Input words regularly scanned (data item 38) are greater than 8,000.

   b. Output words regularly scanned (data item 39) are greater than 15,000.

   c. Input card images regularly scanned (data item 41) are greater than 200.

   d. Output card images regularly scanned (data item 42) are greater than 200.

   e. Input line images regularly scanned (data item 44) are greater than 200.

   f. Output line images regularly scanned (data item 45) are greater than 200.

   g. Input variables regularly scanned (data item 50) are greater than 30.

   h. Output variables regularly scanned (data item 51) are greater than 60.

   i. Number of variables input (data item 140) is regularly greater than 400.

   j. Number of variables output (date item 141) is regularly greater than 200.

   The critical values listed above were empirically determined considering the amount of information to be processed at an interactive terminal. The results were then compared against known Type 1 applications to insure that the selection process was correct.

3. The device type was upgraded to Type 3 (alphanumeric, vector drawing CRT with alpha-numeric keyboard input) if any of the following questions were answered yes:

a. Is graphical information regularly scanned during input preparation (data item 47)?

b. Is graphical information regularly scanned during output review (data item 48)?

c. Is graphical information regularly scanned during execution of the program, if interactive (data item 49)?

d. Would you use a general interactive plotter utility (data item 77)?

e. Would you use a utility to generate and sequence displays interactively, e.g., movies (data item 89)?

4. The device type was upgraded to Type 4 (same as Type 3 except with limited topological input capability, e.g., thumb-wheel driven cross-hair cursor) if an affirmative response was accorded:

   a. Would this program benefit significantly if it could present information as a pictorial display (data item 78)?

   b. Would you use a general interactive curve or surface fitter (data item 79)?

   c. Would this program benefit significantly if information could be selected or altered interactively, viz., using a topological input device (data item 80)?

5. The device type was upgraded to Type 5 (similar to Type 4 but with a large screen CRT and an enhanced topological input device, e.g., light pen) if the user could utilize a general interactive drafting module (data item 86) in conjunction with this program.

If none of the above conditions were met (device type "0"), it was assumed that the program usage would not significantly benefit from interaction.

Figure 4-6 presents the computer printout histogram of device type. It is noted that all 93 programs could benefit significantly from interaction (no type "0"). Further over half required some (perhaps limited) topological input device (Types 4 and 5). Over 80 percent required vector drawing capability (Types 3, 4 and 5). It should be emphasized that the OM Questionnaire sampled current computer usage; the limited requirement for Type 5 interactive terminals reflects the general absence of design (as opposed to analysis) programs in current computer usage. On the contrary, the limited requirements for Type 1 terminals is indicative of a general lack of capability of electromechanical typewriter devices.

A check of the interactive programs against the assigned device type verified that the assignment was reasonably accurate; the exceptions were programs that were clearly on the wrong device (e.g., the computer program that reduced this questionnaire should have utilized a Type 3 device rather than the Type 2 device actually used - if one were available).

```
                                                48
      50.0 +---------------+---------------+---------------+
           I               I               IXXXXXXXX       I
N          I               I               IXXXXXXXX       I
U          I               I               IXXXXXXXX       I
M          I               I               IXXXXXXXX       I
B     40.0 +---------------+---------------+XXXXXXXX-------+
E          I               I               IXXXXXXXX       I
R          I               I               IXXXXXXXX       I
           I               I               IXXXXXXXX       I
O          I               I               IXXXXXXXX       I
F     30.0 +---------------+---------------+XXXXXXXX-------+
           I               I               IXXXXXXXX       I
O          I               I          23   IXXXXXXXX       I
C          I               I        XXXXXXXXXXXXXXXX       I
C          I               I        XXXXXXXXXXXXXXXX       I
U     20.0 +---------------+--------XXXXXXXXXXXXXXXX-------+
R          I               I        XXXXXXXXXXXXXXXX       I
E          I               I        XXXXXXXXXXXXXXXX       I
N          I               I        XXXXXXXXXXXXXXXX       I
C          I               I   10   XXXXXXXXXXXXXXXX       I
E     10.0 +---------------+XXXXXXXXXXXXXXXXXXXXXXXX-------+
S          I           6   IXXXXXXXXXXXXXXXXXXXXXXXX   6   I
           I        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
           I        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
           I        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       0.0 +--------XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                       1       2        3       4       5
             SAMPLE SIZE  93  TYPE OF INTERACTIVE TERMINALS
```

Figure 4-6. Type of Interactive Terminals

4.3.2 Console time (data item 149) and number of interactive cases (data item 150). - Any estimate of console time must account for the human delays in the interactive process as well as the computer delays. The human delays are invariably the longest and are composed separately of setup delays and a per-unit delay for each item being processed (e.g., each graph being prepared). The console time for interactive jobs is the console time per case (data item 12) times the typical number of cases (data item 16) and served as a control on the estimate.

The console time for non-interactive programs was deduced from the OM Questionnaire responses as follows:

1. The principal computer delay was obtained from the summation of the central processor (or "arithmetic unit") time (data item 8) and peripheral processor time (data item 10) and converted to minutes. If the computer delay was in excess of six minutes, it was assumed that the user would not wish to simply wait until the case was run. In these instances it was assumed that the case was run as a "batch spinoff" (spunoff as a non-interactive job) while the user busied himself with other interactive tasks (e.g., plotting previous results or file management) and the computer delay was reset to zero. The terminal type (data item 148) was negated to indicate the "batch spinoff" mode of operation.

2. To the computer delay was added a setup delay (if any units were being processed) and a process delay (based on a per-unit delay times the number of processed units) for each of the following items:

    a. Input words regularly scanned (data item 38).
    b. Output words regularly scanned (data item 39).
    c. I/O words regularly interactive (data item 40).
    d. Input card images regularly scanned (data item 41).
    e. Output card images regularly scanned (data item 42).
    f. I/O card images regularly interactive (data item 43).
    g. Input line images regularly scanned (data item 44).
    h. Output line images regularly scanned (data item 45).
    i. I/O line images regularly interactive (data item 46).
    j. Graphs regularly scanned during input preparation (data item 47).
    k. Graphs regularly scanned during output evaluation (data item 48).
    l. Graphs regularly scanned during interaction (data item 49).
    m. Input variables regularly scanned (data item 50).
    n. Output variables regularly scanned (data item 51).
    o. I/O variables regularly interactive (data item 52).
    p. Pictures regularly scanned during input preparation (data item 53).
    q. Pictures regularly scanned during output evaluation (data item 54).
    r. Pictures regularly scanned during interaction (data item 55)

    The setup and per-unit delays were initially estimated judgmentally from existing interactive applications, subsequent variation was then made in an attempt to derive an estimated console time per case for interactive jobs close to the true console time per case (data item 12). This could not be accomplished accurately due to the subjective nature of both the user estimates (items a through r above) and the analyst's estimates (setup delay and per-unit process delay). Further no attempt was made to distinguish the degree of complexity in the various applications (e.g., of the output "picture") even though a comparison of the estimate to the actual console times clearly pointed out these differences. The estimated console time per case was accepted when a "reasonable," mean-zero variation between estimated and actual console time was obtained by juggling the setup and per-unit process delays.

3. The number of interactive cases (data item 150) was then estimated as the largest integral number of cases that could be processed in one hour (a "reasonable" console sitting) subject to the restrictions:

   a. If the computed number of cases was less than the typical number of cases (data item 16), the typical number of cases was accepted.

   b. At least one case must be processed.

   c. The console time per sitting was then recomputed as the number of cases times the console time per case. If this time was greater than two hours, the number of cases was reduced by one until either the console time dropped below two hours or the number of cases dropped to one.

4. The console (sitting) time (data item 149) was finally recomputed by multiplying the resulting number of interactive cases (item 3 above) and the console time per case (item 2 above).

Figure 4-7 presents the resulting console time for the representative OMs. It is noted that the results are bi-modal with a mode at approximately 80 minutes and another at around 120 minutes. Few OMs exceeded 150 minutes (2.5 hours) continuous console sitting, the mean console time was 76 minutes and the minimum was 45 minutes.

The final results indicated that 15 of the 93 candidate OMs were being run as "batch spinoff" jobs (i.e., indicating that their computer delay, item 1 above, was in excess of 6 minutes).

4.4 Validity of Results

Validity of the results obtained from the OM Questionnaire depends directly on the size of the sample, whether the sample is a representative cross-section of candidate OMs for the design process, and the accuracy of data reduction. The size of the

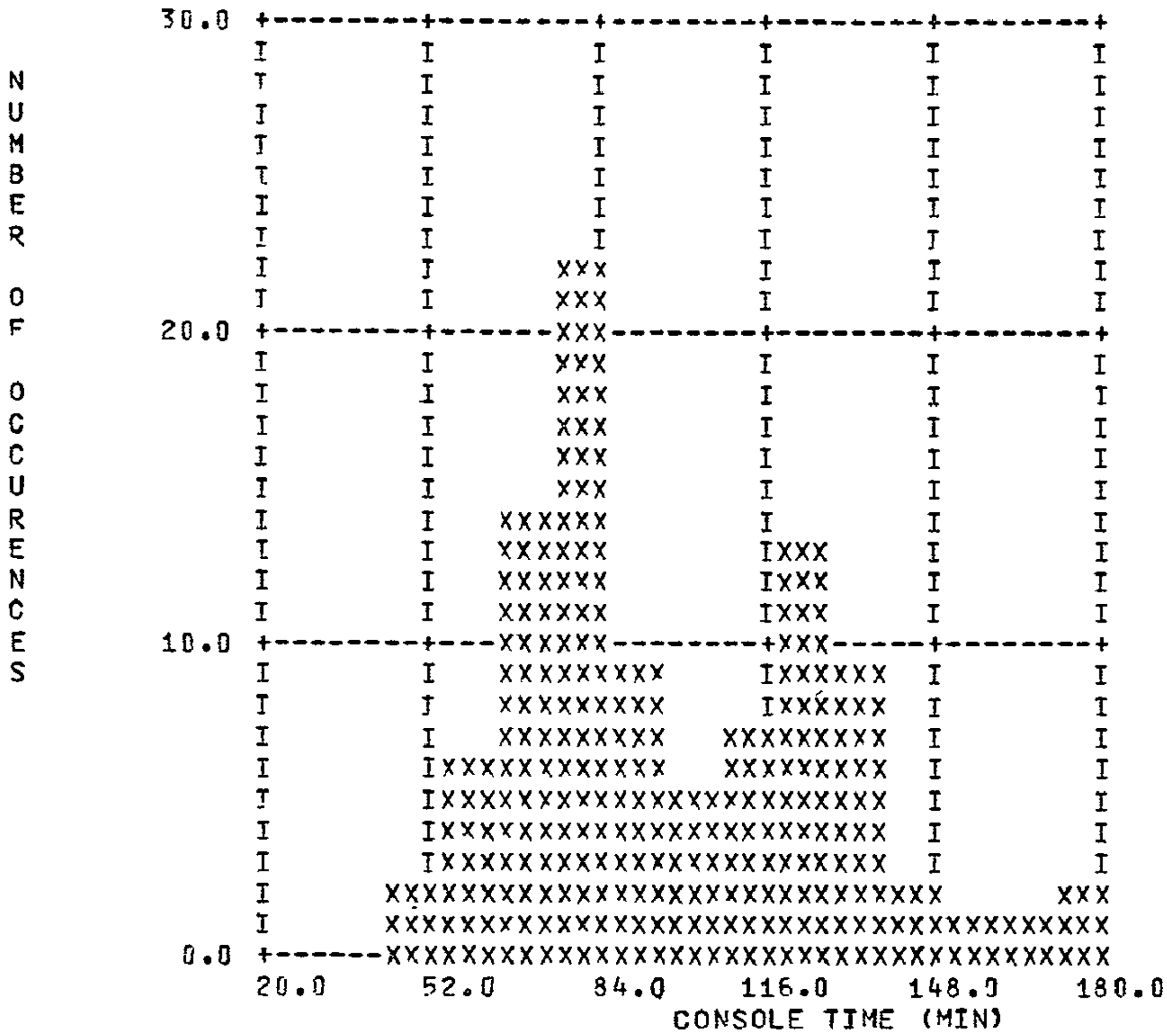


Figure 4-7. Console Time for Representative Sample of OMs

sample (93 reduced Questionnaires) is quite adequate for statistical inferences of this sort. Further, data reduction was accomplished by interactive digital reduction yielding both adequate precision and the visibility to detect errors in OM Questionnaire reduction (which there were). This left only representation to be evaluated.

4.4.1 Currency. - Figure 4-8 presents the calendar time elapsed since the candidate OMs were last used. With few exceptions the programs were in use within the six months immediately preceding receipt of the Questionnaire. The mean usage lag was just under two months whereas the extreme was just under three years. A check into the few programs falling beyond the six month interval uncovered that they all were model development programs (e.g., to derive influence coefficients) which had not been recently required since the mathematical models in use (Atlas Space Launch Vehicles) are of demonstrated precision.

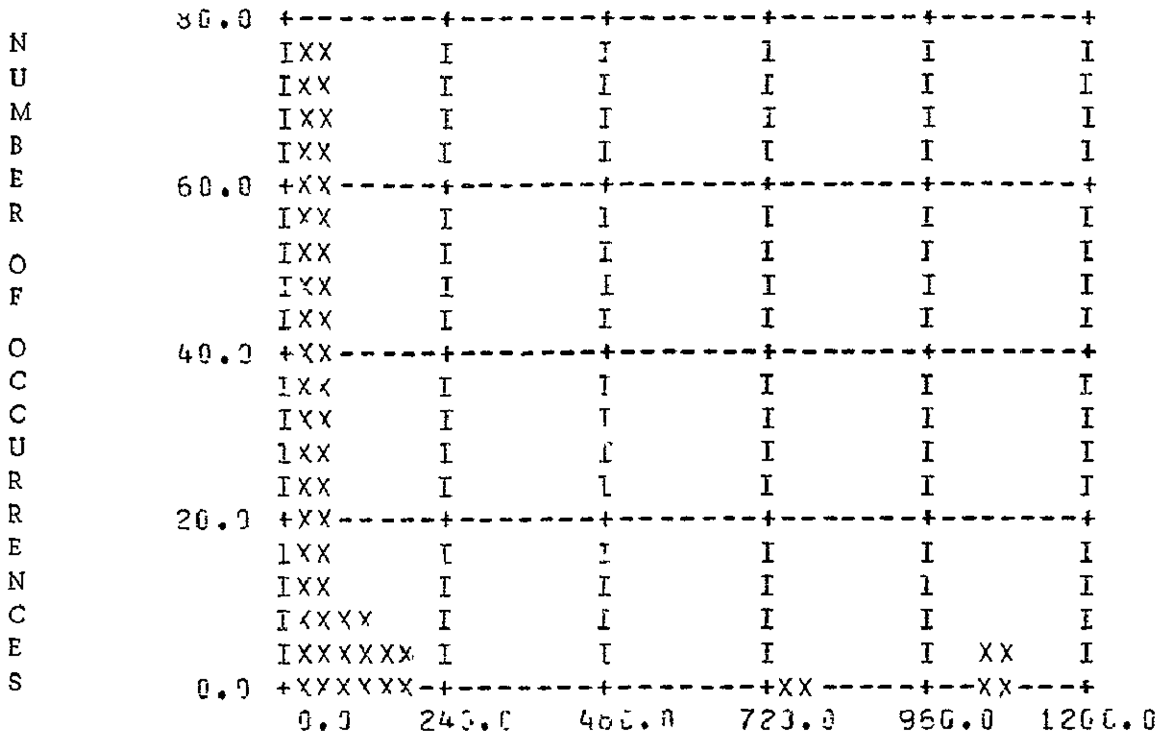


Figure 4-8. Calendar Days Since Last Program Use

Figure 4-9 presents the calendar time elapsed since the candidate OMs were last modified indicating (to a certain extent) programming currency. Again with few exceptions the programs had all been modified within the 3 years preceding the Questionnaire As would be expected most of the exceptions were the model development programs in infrequent use. The mean modification lag was 0.8 year and the extreme just under 7 years (last modified in 1965).

Figure 4-10 presents the calendar time elapsed since the candidate OMs were first conceptualized and created, indicating program stability. The figure has the appearance of a classical Poisson distribution with a mean of 4.5 years (1968) and an extreme of 17 years (1955). In interpreting this figure it must be bore in mind that the programs had undergone (usually extensive) modification since origination (Figure 4-9) but that their original intended use had been envisioned quite some time in the past, e.g., as far back as 1955. This figure emphasizes the tremendous investment aerospace has in existing (principally FORTRAN) digital computer program - i.e., in candidate IPAD OMs.

NUMBER OF OCCURRENCES

```
60.0 +--------+--------+--------+--------+--------+
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
40.0 +XX------+--------+--------+--------+--------+
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
20.0 +XX------+--------+--------+--------+--------+
     IXX      I        I        I        I        I
     IXX      I        I        I        I        I
     IXXXX    I        I        I        I        I
     IXXXX    I        I        I        I        I
     IXXXX    I        I        I        I        I
     IXXXXXXX I        I        I        I        I
     IXXXXXXX I        I        I        I        I
     IXXXXXXX IXX      I        I        I        I
     IXXXXXXXXXXXXXXXXX       XX      XX IXX      I
 0.0 +XXXXXXXXXXXXXXXXX-------XX------XX-+XX------+
      0.0     600.0    1200.0   1800.0   2400.0   3000.0
```

Figure 4-9. Calendar Days Since Last Program Modification

It was concluded that the collection of 93 candidate OMs represented a reasonably stable, current collection of aerospace capability in reasonably current usage.

4.4.2 Cross-sectional applicability. - To ascertain whether the Questionnaire sample contained a representative cross-section for aircraft design, a comparison with a recently completed aircraft project phase was undertaken. The best available data for this comparison was the Space Shuttle Phase B booster detailed predesign (Reference 14) which ran from 1 July 1970 to 30 June 1971.

Computer charges for the Space Shuttle booster detailed predesign were collected for comparison with the Questionnaires. Figure 4-11 presents the project's computer changes as a function of calendar time and illustrates a typical workforce buildup with

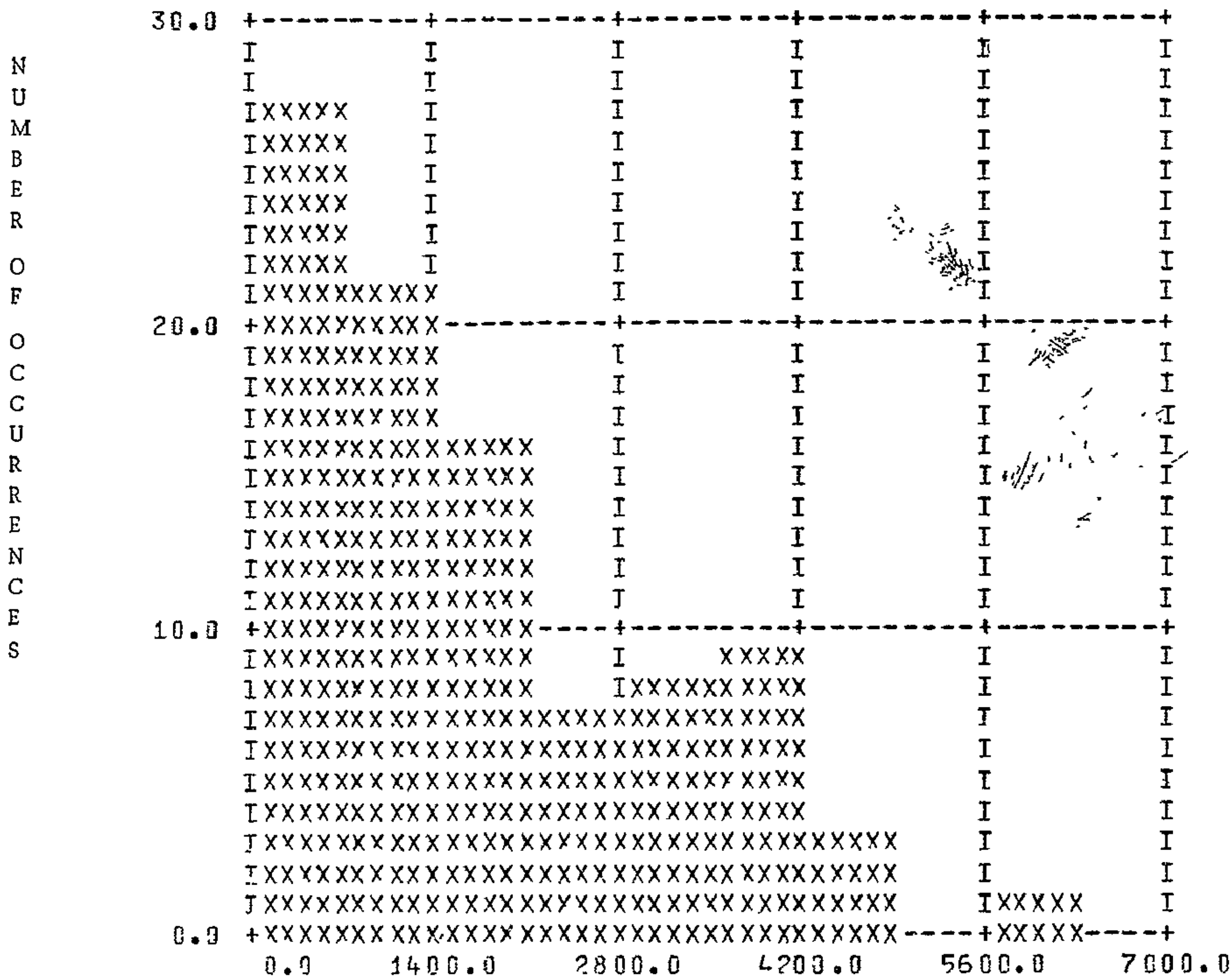


Figure 4-10. Calendar Days Since Program Initial Creation

a peak of activity in the closing phase when final study results were being completed. An analysis of the computer time-charging algorithms (there was a switch in the algorithm toward the end of the study) yielded an estimated 271 hours expended on the CDC 6400 central processor (CP).

For comparison, the Questionnaire estimates were obtained by multiplying the Central Processor (CP) time per case in seconds (data item 8 of Figure 52) with the estimated number of cases for a predesign phase (data item 4) and dividing by 3600 to convert to hours. These hours were then summed - excluding those programs used only for (radio-guided) launch analysis - and 208 CDC 6400 CP hours resulted. These results are presented in Table 4-1 together with results from the Conceptual (data item 3) and Detailed Design (data item 5) phases of the complete design process.

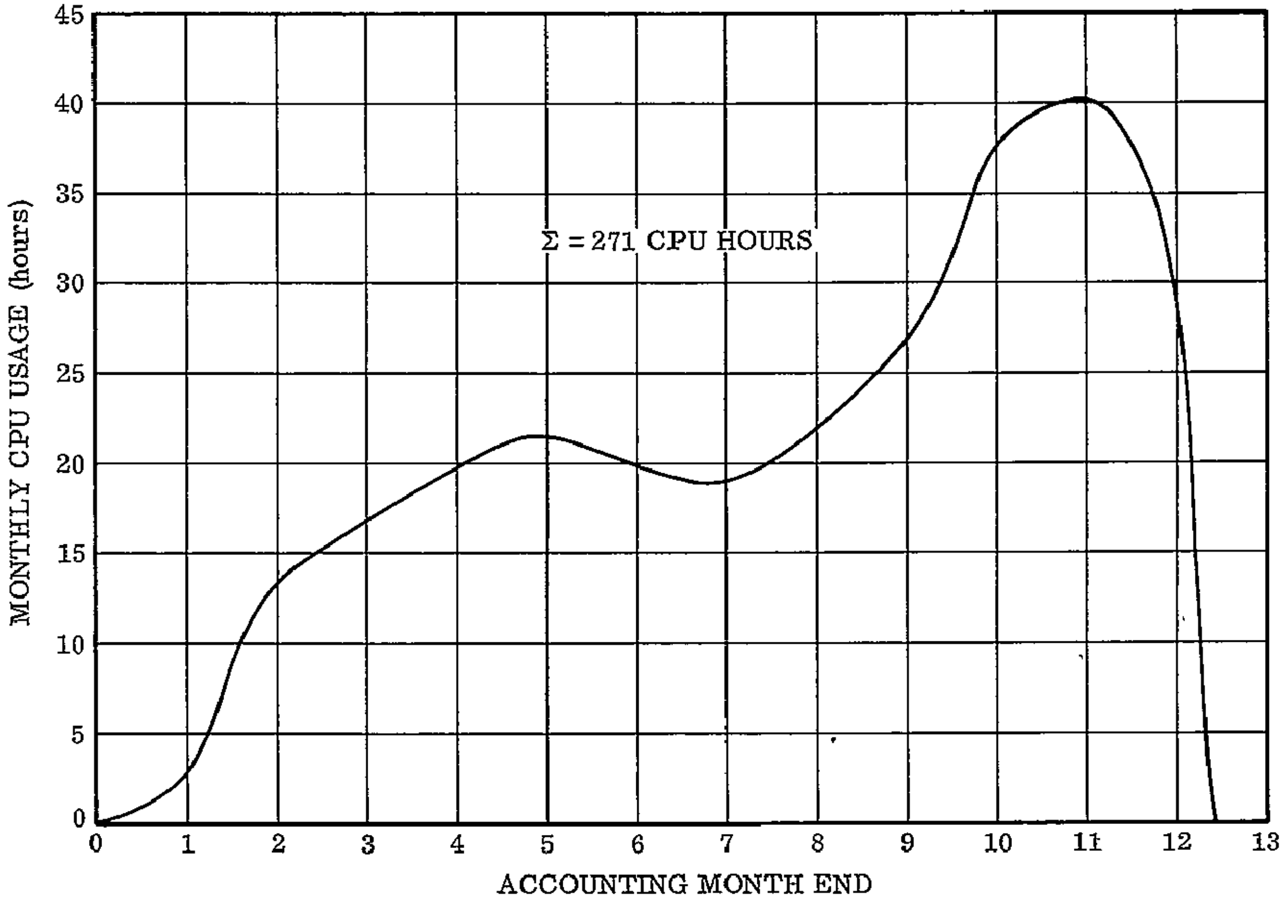


Figure 4-11. CDC 6400 Computer Usage by Month, Space Shuttle Project, Phase B

Table 4-1 shows estimated CP hours for a typical aircraft project. It is noted that of the 93 programs for which Questionnaires were reduced, only 59 were directly applicable to aircraft predesign. However 59 is a sufficient sample for comparative purposes. If this reduced sample were representative, it would appear that the Space Shuttle booster detailed predesign was approximately 30 percent "over budget" (271/208 = 1.30). This is not at all unrealistic considering that the Space Shuttle booster

TABLE 4-1
ESTIMATED CDC 6400 CP HOURS FOR
A TYPICAL AIRCRAFT PROJECT

Project Phase	Sample Size	CP Hours
Conceptual	40	89.5
Predesign	59	208.4
Detailed Design	60	1807.1

under study was a winged, recoverable booster that launched like a missile, staged like an external store, reentered like a high L/D manned reentry vehicle, and landed like a conventional jet transport. Analysis of its total flight profile span that of a conventional high-performance jet aircraft in addition to the missile-like launch and complex staging phases. A typical interactive OM that was used in these projections is documented in Reference 15.

It was concluded that the subset of 59 candidate OMs directly applicable to an aircraft predesign phase presented a realistic picture when compared to a recent detailed predesign conducted for the Space Shuttle booster over a year's period.

4.4.3 General applicability. - At the onset it must be stressed that the Questionnaire surveyed computer programs in existence and current usage, as such it could not adequately represent the total design process of IPAD, particularly in the detailed design phase (e.g., the table above). Most notably missing is a adequate representation of board designers who have yet to become fully computerized. Indeed, candidate Design OMs accounted for only 6 of the 93 reduced Questionnaires whereas Abstract Analysis, the largest single category and the one which has pioneered computerization, accounted for 60 (Figure 4-1). At best the sample represents projected usage of OMs immediately following a first-release IPAD system. Although such a limitation is not desired, there is no practical way to estimate and include disciplines which lack full computerization.

Further, although not directly involved in the design process, it would have been beneficial to get more management applications in the sample. All but two programs were FORTRAN, the two were in BASIC - there were no COBOL programs in the sample although 11 of the 93 programs were management related (Figure 4-1). Five of the FORTRAN programs were actually generated by translators (or precompilers) rather than computer programmers.

Every effort was made to secure as many operational interactive programs for the sample as were available since it was presumed that IPAD would be principally interactive. Twenty of the 93 programs were interactive (several could operate either interactively or batch as suited the user's purpose). However 74 of the 93 respondents saw immediate benefits from the ability to interact with the input and/or output of their programs; 35 of these 74 expressed the desire to interact with their programs during program execution (e.g., redirecting a simulation which is obviously going awry). As noted in Subsection 4.3.1 (on interactive terminal types) when consideration was given to the use of envisioned IPAD general purpose utilities, all of the 93 candidate OMs would be run, at least in part, interactively. Indeed 69 of the 93 respondents expressed a need for a graphical plotter and 45 expressed the need for a pictorial plotter.

A general effort was also made to include programming systems in the sample, i.e., collections of individual self-contained programs which have been modified to operate optionally as a single run-unit. There were four such programming systems included (NAMESIM, NASTRAN, DYNES and DYNAMO, see item 5 of Section 4.2 for details) together with 19 member programs.

Of the 93 candidate OMs, 66 were termed "general purpose" (meaning that they addressed a general rather than specific application, e.g., General Heat Transfer) and 23 were termed "multidisciplinary" (meaning that they were used by more than one discipline, e.g., NASTRAN). An unusually large number (47) were overlayed (a "layered" program structure so that only those portions of the program which must communicate need be in the computer's central memory), this was perhaps due to unusually small central memory (65,000 storage locations) on the host CDC 6400 computing system. Only 16 of the 93 used double precision arithmetic (probably due to CDC's unusually large 60-bit standard computer "word"). Over one-third (34) regularly used microfilm as a means of recording output (principally graphical).

4.5 Representative Results

Questionnaire results were utilized in three distinct forms throughout the study:

1. Direct histogram output of a given data item, e.g., Figures 4-6 through 4-10.

2. Summation of various data items, e.g., Figure 4-1 which presents the projected usage of candidate IPAD interactive general purpose utilities by Questionnaire respondents. The Optimizer/Sensitivity Extractor/ Parameterizer utility usage estimate was obtained from data items 82, 83 and 87 (Figure 4-4).

3. Output synthesized from several data items, e.g., Figure 4-13 which combines data items 149, (3, 4 or 5), 150, and (19+20, 21, 22, 23+24, or 25+26) of Figure 4-4 in accordance with the equation presented in Figure 4-3. As noted on Figure 4-13 (the abscissa of which is in a logarithmic scale), the requirement for engineering aides (or assistants) is only significant in the Detailed Design phase where - it is suspected - that the Optimizer/Sensitivity Extractor/Parameterizer and the Graphic Plotter interactive utilities (Figure 4-12) may serve to reduce this requirement two orders of magnitude (commensurate with the needs in the Predesign phase); this follows from the observation that (according to Quesionnaire respondents) engineering aides were being used to prepare runs for parameter studies and to cross-plot results.

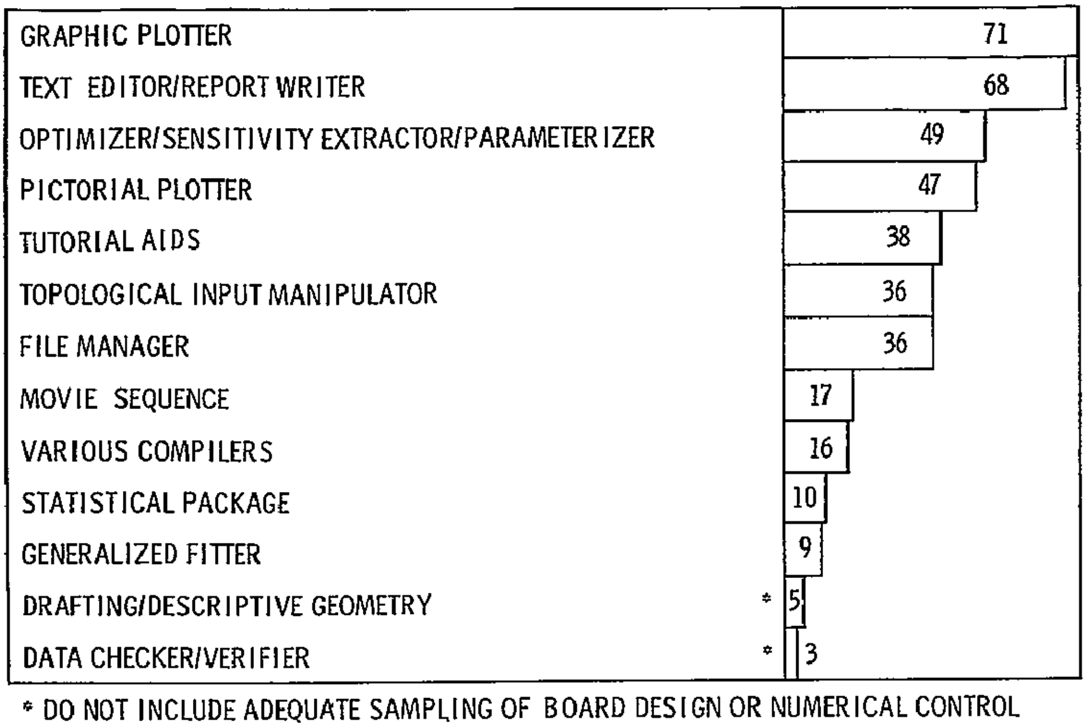


Figure 4-12. Projected Usage of IPAD Interactive Utilities by Questionnaire Respondents

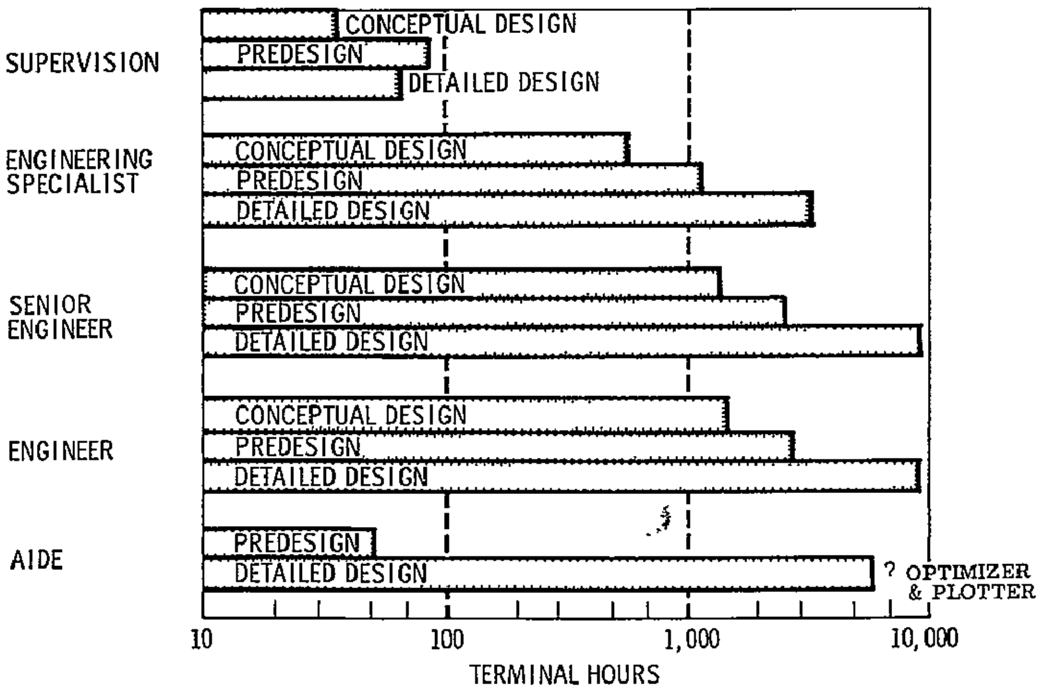


Figure 4-13. Estimated Terminal Hours by Design Phase and Engineering Skill Level

5 HARDWARE/SOFTWARE SYSTEM SCOPING

This section defines the minimum hardware and software required in support of IPAD.

The study began by developing a project basis which is presumed to draw upon IPAD support (Section 5. 1). The interactive terminal requirements were then estimated from the OM Questionnaire results of Section 4; and finally typical terminal requirements were determined (Section 5. 2).

The role of the minicomputer is briefly examined (Section 5. 3) as a means of reducing response time, and the recommended terminal configurations are defined. The role of the host (maxi) computer is examined to ascertain those features (hardware and software) deemed important to IPAD; typical configurations are examined and candidate large scale scientific computing system are established as a minimum requirement for IPAD (Section 5. 4). Three "target" computing systems are established to test the efficacy of the IPAD design (Section 5. 5). The role of computing networks is briefly treated (Section 5. 6). The technical discussion is concluded with a brief treatment of advanced computing devices/systems (Section 5. 7).

5. 1 Reduction in Design Time

In order to form a basis for the purpose of system scoping, it was necessary to identify the duration of the various aerospace system design phases and make a projection of reasonable calendar time reductions possible with the envisioned IPAD system. Typical projections are shown in Table 5-1. That a reduction will take place is indisputable although the amount of reduction may be open to question and would depend on the project.

TABLE 5-1
REQUIRED DESIGN TIME (MONTHS)

	CONCEPTUAL	PREDESIGN	DETAILED DESIGN
NOW	3	9	18
WITH IPAD?	2	4	12

5. 1. 1 <u>Conceptual design phase.</u> - The bulk of the conceptual design phase is spent in defining and evaluating the system requirements, performing vehicle trade studies, and evaluating multiple configuration options. Emphasis is placed on the physics of

the process; conceptual "solutions" are conceived and evaluated relative to the functional requirements. Every effort is made to be complete so that no good solution is overlooked.

During this phase concepts are ranked in terms of uncertainty and risk; computer solutions are made to quantify uncertainty (sensitivity studies); several possible designs are evolved; and gross sizing (computer studies) are accomplished to bracket the possible designs. The conceptual design phase typically terminates with a definition of the better functional approaches and at least one possible design (hardware) approach.

The envisioned reduction in time span includes increased efficiency of the analysts due to improved computer solution feedback.

5.1.2 Predesign phase. - The predesign phase begins after the conceptual design ends and is rigorously practical (e.g., even tooling considerations are taken into account). Computers are the main tool of the predesigner and a considerable amount of computer time is expended in design trade studies. Several parallel design alternatives are usually under evaluation, usually with differing emphasis. An optimum design approach is selected and expanded in detail, final projections of performance (computer solutions) are worked out, and sensitivity (and risk) studies are concluded. The predesign phase is typified by very high computer usage.

It is envisioned that IPAD will significantly reduce the predesign phase time span. The envisioned reduction is limited only by man's ability to digest the results and foresee the correct path to follow in achieving the optimum design. The reduction includes increased efficiency resulting from improved communications between predesigners through IPAD's communications files (see Subsection 2.3.6).

5.1.3 Detailed design phase. - The detailed design phase follows the predesign phase and can slightly overlap. Detailed analysis, drawings, specifications, and tests are made for the final design which is to be manufactured; make or buy decisions are important here. Again the computer becomes a necessary tool in this design and manufacturing process. The time for this process must be reduced and it must first evolve from its present restricted use of the computer (the limited number of available OMs) before it can benefit more than indirectly through IPAD. IPAD will however assist greatly in this evolution.

The envisioned design phase time reduction reflects only the short term benefits derived from IPAD, as would be in evidence, perhaps, within two years from IPAD's first release. It would be difficult to estimate the eventual total impact of computer automation, nor is it required in this scoping of hardware.

5.1.4 Conclusion. - It is reasonable to assume that the envisioned reduction in design time (nearly 40 percent overall) could come about through IPAD within the first few years following its first release. This reduction is used as the basis of the analyses which follows.

5.2 Interactive Terminal Selection

The estimated terminal hours by device type and design phase - Figure 5-1 - was derived from the OM Questionnaire in much the same manner as Figure 4-13 of Section 4.5. It is used to select an appropriate number of interactive terminal types as envisioned for a single project within IPAD. The terminal configuration is sized for one project. A company is usually engaged in a number of projects and technical activities in more or less parallel fashion in time. Since activity in any analytical process varies with time in a project (i.e., builds up to a peak activity and tails off to a lower level), only the incremental increase in the hardware configuration is actually needed to accommodate other projects rather than a full replication of this configuration for each project. The representation of multiple projects — although not difficult to accomplish — would necessitate statistical inference and additional computer usage (e.g., Monte Carlo) and was considered to be outside the scope of this trade study.

In the following discussion it is important to keep in mind that for terminal type $m<n$, a type n terminal can do all the functions of a type m terminal (see Subsection 4.3.1 for definition of terminal types).

5.2.1 Consolidation of device types. - Due to the relative insignificance of Type 1 usage in comparison with other device types - Figure 5-1 (but note the logarithmic scale) - Type 1 usage was lumped with that of Type 2. This followed since the Type 2 CRT devices could accomplish all the Type 1 functions with nearily identical cost, while offering substantially better features (see Figure 4-5 and related discussion in Subsection 4.3.1).

Further, Type 3 devices - with only minimal topological input capabilities - were lumped with Type 4 devices with good topological input features. Here the costs are essentially non-overlapping (Figure 4-5), however it was reasoned that for comparable features (i.e., excluding topological input) the device costs are nearly identical. There is essentially little hardware difference between Type 3 and 4 devices. Type 4 devices additionally employ a hand manipulator (e.g., joy-stick) and circuitry to locate displayed items (e.g., positioning a cross-hair cursor on the item). The location of these items in the display is then returned to the requesting program (software), typically when a button has been pressed. This provides for exceptionally better functional capabilities to Type 4 devices for essentially insignificant costs.

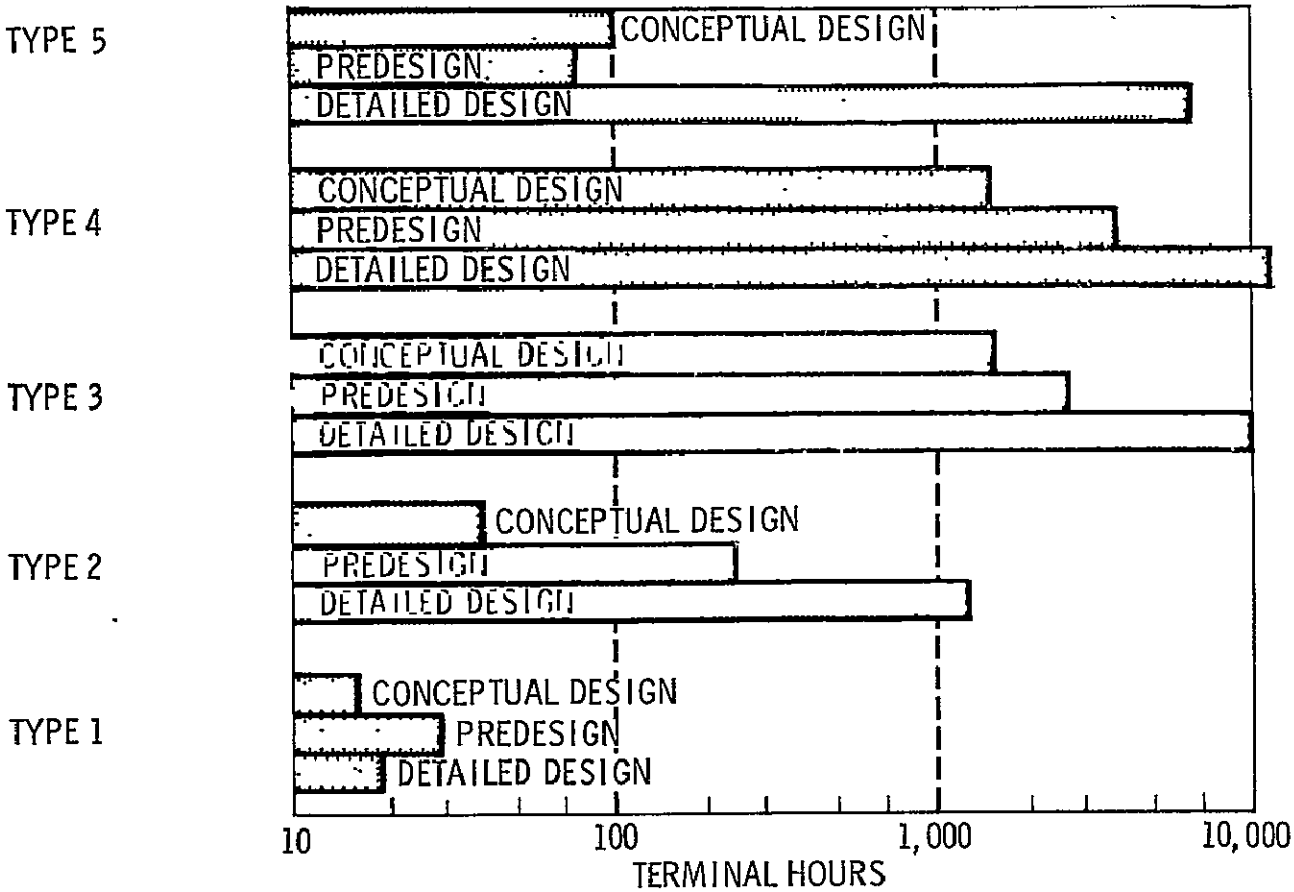


Figure 5-1. Estimated Terminal Hours by Device Type

Unlike the more limited Type 3 devices, Type 4 devices range in capability (and cost) to approach the least expensive Type 5 devices which they emulate.

5.2.2 Estimated terminal loading. - The estimated terminal loading was computed from the estimated terminal hours for each device type (Figure 5-1), the time span of each design phase (Table 5-1), and the consolidation and projected utilization of device types discussed in the preceding subsection. At 168 manhours per month (one shift), the following number of terminals result.

NUMBER OF TERMINALS
(CURRENT SCHEDULES/IPAD SCHEDULES)

GROUP (TYPE)	CONCEPTUAL	PREDESIGN	DESIGN
2 (WITH 1)	0.1/0 2	0.2/0.6	0.4/0.6
4 (WITH 3)	6.0/9.0	3.1/9.3	6 9/10.3
5	0.2/0.3	0.04/0.1	2.1/3.2

The maximum loading occurs in the design phase principally due to the large numbers of personnel involved in that phase (as opposed to increased computer utilization). Note however that the utilization of each device type (except Type 5) is reasonably level throughout the three design phases (producing level loading). The minimal use of Type 2 terminals (even with Type 1 devices lumped in) is not considered economical. If these are lumped together with Type 4 devices, the following typical loading - Type 4 being essentially independent of design phase - results (rounded off):

- **TYPE 4 TERMINAL LOADING ESTIMATE – 11**
- **TYPE 5 TERMINAL LOADING ESTIMATE – 3**

IPAD would provide a complete interactive drafting descriptive geometry utility with a numerical control interface to assist the board designer who has a minimal number of computerized tools. Some of the board designer's tasks could be accomplished on Type 4 devices; the more difficult tasks require Type 5 devices. The availability of this design tool will (subjectively) increase the loading to:

- **TYPE 4 TERMINAL LOADING ESTIMATE – 15**
- **TYPE 5 TERMINAL LOADING ESTIMATE – 6**

However the peaking phenomenon also affects these estimates. Provisions for more than the average number of terminals required must be made to account for fluctuations in the demand (a subjective 33 percent peaking is assumed):

- **TYPE 4 TERMINAL LOADING ESTIMATE – 20**
- **TYPE 5 TERMINAL LOADING ESTIMATE – 8**

This is the final loading and reflects the peak demand on the host computing systems.

5.2.3 Numbers of interactive terminals by type. - Actual terminal requirements will always exceed loading estimates due to:

1. Availability constraints:

   a. Device is in use (unavailable).

   b. Device location (an available device is located in another building too far removed).

   c. Hardware/software difficulties (down time).

2. Nonconstant usage constraints:

   a. Momentary interruptions (slack time).

   b. Unforeseen problems requiring release of scheduled time with nobody immediately available to use device.

   c. Scheduling slack (e.g., between scheduled users).

It is difficult to quantify a utilization parameter knowing only those factors which influence utilization. However actual experience from scheduling similar devices (e.g., analog and hybrid computers) suggests that 30 percent is reasonable. This utilization parameter was used in the subsequent calculation for Type 4 devices.

The Type 5 terminals will experience somewhat less demands due principally to the expected uniform loading. (This is essentially due to a single class of users – the designers – each spending more hours per sitting and more sittings per week.) Further, as noted above, the conceptual and predesign phases make little use of Type 5 terminals. A 25 percent availability factor was assumed for Type 5 devices.

The estimated final terminal requirements for a single, comprehensive aerospace project is contained in Table 5-2.

TABLE 5-2
NUMBER AND TYPE OF INTERACTIVE TERMINALS

- TYPE 4 – 26
- TYPE 5 – 10

Table 5-3 presents an excellent summary of the major features of Type 4 and 5 Graphic CRT Terminals. This table was pieced together from the original in Reference 16 and is presented as a convenience to the reader with the kind permission of MODERN DATA. Note that the table has been modified to show the IPAD type and to update several entries, principally the addition of CDC's GPGT. Several Type 3 terminals have been included for comparative purposes. Note particularly the pricing information (although somewhat out of date).

5.3 Typical Terminal Configurations and the Role of the Minicomputer

Usually it is not appropriate from a central memory management standpoint that the large host computer be burdened with refreshing the image on the display terminal. Special hardware memories were added to the display controllers to accomplish this function. As minicomputer prices plummetted (Figure 5-2) the minicomputer replaced these special memories and also served as the refresh device, the communications handler, and the user-interaction handler.

This approach was partially used by CDC in the implementation of the CDC 274 system with their 1700 minicomputer. While the CDC minicomputer system is

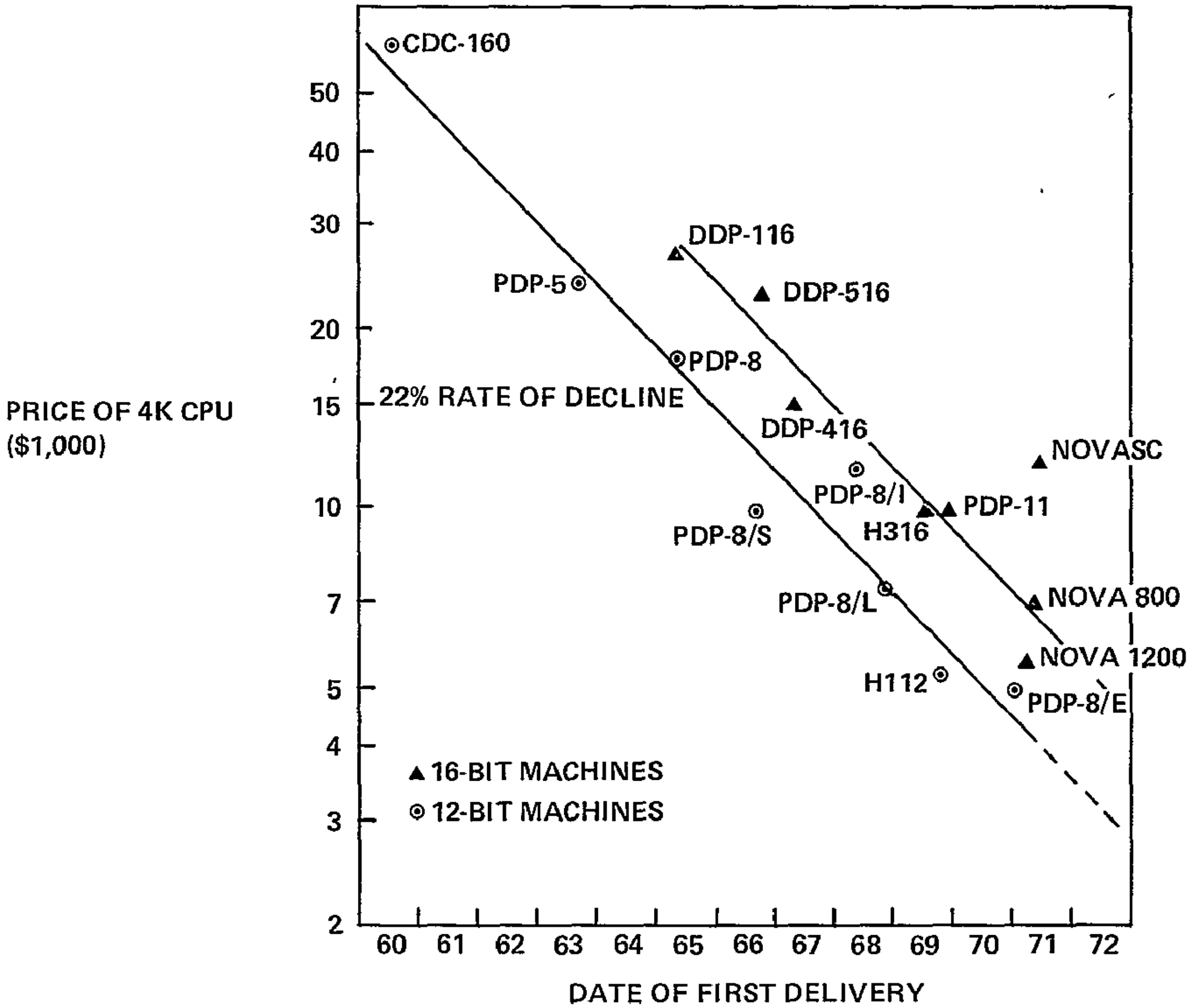


Figure 5-2. Minicomputer Costs

versatile, does remote job entry (RJE) and can handle up to six CDC 274 CRTs concurrent with the RJE, it is basically a communication processor, and much of the interactive processing is still carried out on the host computer. The next generation CDC terminal (due out in late Fall 1973), the General Purpose Graphic Terminal (GPGT), incorporates a full capability minicomputer (SC-1700) as its controller. It uses part of the controller's central memory for refresh (instead of separate refresh store) and takes over some interactive chores such as item erasing, item moving (translation), display area scissoring, and zooming (scaling) via hardware and table driver schemes in the local mini.

IBM has used similar techniques in the past with the IBM 1130 computer (no longer supported for graphics), and DEC with their PDP-8 and PDP-11 computers.

TABLE 5-3.*
GRAPHIC CRT TERMINALS

COMPANY	ADAGE				COMPUTEK
MODEL	ARDS 100B	AGT 110	AGT 130	AGT 150	400 Display System
DISPLAY Size (inches)	6 5 X 8 5	13 X 14	13 X 14	13 X 14	8 25 X 6 4
Char Capacity	4160	3840	3840	3840	3400
Char/Line	80 or 40	96	96	96	85
Total Lines	52 or 26	40	40	40	40
No Char & Code	96, 192 (opt)	64/96 ASCII	64/96 ASCII	64/96 ASCII	96 ASCII
Char Generation	7 X 9 Dot	Stroke	Stroke	Stroke	Stroke
Refresh Rate (Hz)	ST - See 1	60 (variable)	60 (variable)	60 (variable)	ST
DATA ENTRY Insert/Delete	Line Delete (opt)	Char & Line (std)	Char & Line (std)	Char & Line (std)	————
Tabulation	Horz (opt)	Horz & Vert (std)	Horz & Vert (std)	Horz & Vert (std)	————
Formatting	————	std	std	std	————
Page Roll	————	std	std	std	————
Split Screen	————	std	std	std	————
Other	See 1, 2, 3	————	————	————	See 1
GRAPHICS Visible Raster	1169 X 1501	13 X 14 inch	13 X 14 inch	13 X 14 inch	1,024 X 800
Max Component	1023 raster units	10 inches	10 inches	10 inches	————
Positioning Modes	Absolute	Absolute	Absolute	Absolute	————
Vector Modes	————	Absolute	Absolute	Absolute	Absolute & Relative
View Manipulation	See 1	Shift, 2D Zoom (std)	Shift, Rotate, 2D & 3D Zoom (std)	Shift, Rotate, 2D & 3D Zoom (std)	————
Pointers	All Types	All Types	All Types	All Types	Light Pen Tablet Joystick
Other	See 4	————	See 1	See 1	————
INTERFACING Interface	————	RS 232 B Parallel Computer	RS 232 B Parallel Computer	RS 232 B Parallel Computer	RS 232
Transmission Rate	50 Kbps	to 1 Mbps	to 1 Mbps	to 1 Mbps	to 9600 bps
Mode	Half/Full Duplex Echoplex	Half/Full Duplex	Half/Full Duplex	Half/Full-Duplex	Half/Full Duplex
Other	See 5	301B, 203	301B, 203	301B, 203	————
INTERNAL PROCESSOR	————	Adage DPR-4 8K X 30 32K Expandable	Adage DPR-4 8K X 30 32K Expandable	Adage DPR 4 8K X 30 32K Expandable	————
OPTIONS	PC	PC, LP, PT, MT, DD	PC, LP, PT, MT, DD	PC, LP, PT, MT, DD	————
PRICING	$8,700	$98,000	$147,000	$167,000	$6,700
REMARKS & OTHER FEATURES	[1]Refresh Mode Editing & Dynamics [2]Keyboard Cursor Control (opt) [3]Repeat Key (opt) [4]HW Dashed Lines [5]TTY-, IBM 360-Compatible	————	[1]HW Array Transformation for Object Scale, Picture Scale, 3D Translation & Rotation, Depth Cueing· Intensity Scale & Displacement		[1]Keyboard Cursor Control
IPAD TYPE	3	4 or 5	5	5	4

ABBREVIATIONS OPTIONS PC - Printer/Copier (Thermal, Optical, Electrostat) • LP-Line Printer • PLT - Plotter • PT - Paper Tape • MT - Magnetic Tape Transport • DD - Disk Drive
OTHER ST - Storage Tube Display • HW — Hardware Feature

* [illegible]

ORIGINAL PAGE IS OF POOR QUALITY

TABLE 5-3.*
GRAPHIC CRT TERMINALS, Contd

COMPANY	CONOGRAPHIC	CONTROL DATA			DIGITAL EQUIPMENT
MODEL	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
DISPLAY [illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	———	[illegible]	[illegible]	[illegible]	[illegible]
Char/Line	to 341	64/86	Varies - 136 to 46 4 Sizes	146	72
Total Lines	to 158	43/64	Varies - 136 to 46 4 Sizes	102	55
No Char & Code	96 ASCII	64 BCD	128 ASCII	Software	ASCII
Char Generation	Stroke	7 X 9 Stroke	5 X 7 Stroke	Stroke	Stroke
Refresh Rate	ST - See 1	50 (programmable)	40	40	30
DATA ENTRY Insert/Delete	———	Char & Line (std)	Char & Line (std)	Char & Line (std)	———
Tabulation	Horz & Vert (opt)	Vert (std)	Horz & Vert (std)	Horz & Vert (std)	———
Formatting	———	std	std	std	———
Page Roll	———	opt	opt	opt	———
Split Screen	———	opt	std	opt	———
Other	———	———	———	———	———
GRAPHICS Visible Raster	2048 X 1558	12 X 12 inch	20 inch round	20 inch round	1024 X 1024
Max Component	2048 raster units	1024 raster units	4096 Raster Units	4096 raster units	1024 raster units
Positioning Modes	Absolute & Relative	Absolute & Relative	Absolute & Relative	Absolute & Relative	Absolute
Vector Modes	Absolute & Relative	Absolute & Relative	Absolute & Relative	Absolute & Relative	Relative
View Manipulation	Shift, Rotation (std) See 2	Shift (std) Rotate (opt)	Virtual Windows Logical Scissoring Hardware Zoom	Shift, Rotate, 2D Zoom (opt)	Shift, 2D Zoom (std)
Pointers	Tablet, Mouse Joystick	Light Pen	Light Pen	Light Pen, Tablet	Light Pen, Tablet
Other	See 3	———	———	———	See 1 2
INTERFACING Interface	RS 232 C Parallel Computer	RS 232 B Parallel Computer	Parallel Computer	Parallel Computer	RS 232 B Current Loop
Transmission Rate	———	———	40.8K	40.8 K	———
Mode	Half/Full-Duplex	Full-Duplex	———	———	———
Other	TTY Compatible	201 A/B, 301 B	———	301 B	———
INTERNAL PROCESSOR	Conographic 16 4K X 16	CDC 241/242/243 4K X 12 12K Expandable	Integral CDC SC-1700 4K X 16 32K Expandable	CDC 3344/1744 4K X 16 16K Expandable	DEC PDP-15 4K X 18 128K Expandable
OPTIONS	PC	PC, LP, PT, MT, DD		———	PC, LP, PT, MT, DD
PRICING	$8,950	$68,900	$90,000	$91,160	$36 000
REMARKS & OTHER FEATURES	[1]40 Hz Refresh Mode [2]Scaling 1 to 15X [3]HW Point, Vector Curve, Figure, Char & Symbol Generators	———	1 2^{16} X 2^{16} Addressable Virtual Picture 2^{12}X2^{12} Display Raster (4096)	———	[1]Scissoring, 16 Scales [2]Software Rotate & 3D
[illegible]	[illegible]	[illegible]			

ABBREVIATIONS [illegible]
MT - Magnetic Tape Transport • DD - Disk Drive
OTHER ST - Storage Tube Display • HW — Hardware Feature

* [illegible]

TABLE 5-3.*
GRAPHIC CRT TERMINALS, Contd

COMPANY	EVANS & SUTHERLAND	HAZELTINE	HAZELTINE (Contd)		HONEYWELL
MODEL	LDS-1	DDG-1	DDG-3	DDG-5	316/516 – 7420
DISPLAY Size (inches)	—	TV Monitor	TV Monitor	TV Monitor	12 X 12
Char Capacity	2000	7738	7738	4176	—
Char/Line	50	146	146	87	—
Total Lines	40	53	53	48	—
No Char & Code	256	128	128	128	64 ASCII
Char Generation	Stroke	to 32 X 32 Dot	to 16 X 16 Dot	to 14 X 20 Dot	Stroke
Refresh Rate (Hz)	30	30	30	30	40
DATA ENTRY Insert/Delete	—	Char (std)	Char & Line (std)	—	—
Tabulation	—	—	Horz & Vert (std)	—	—
Formatting	—	—	std	—	—
Page Roll	—	—	std	—	—
Split Screen	—	—	std	—	—
Other	—	See 1	—	—	—
GRAPHICS Visible Raster	4096 X 4096	1024 X 480	1024 X 480	612 X 439	1024 X 1024
Max Component	262 raster units	1024 raster units	1024 raster units	945 raster units	—
Positioning Modes	Absolute & Relative	Absolute	Absolute	Absolute	Absolute & Relative
Vector Modes	Absolute & Relative	Absolute	Absolute & Relative	Absolute	Absolute & Relative
View Manipulation	Shift, Rotate, 2D & 3D Zoom (opt)	Shift (std)	—	—	—
Pointers	Tablet	—	Light Pen	—	Light Pen
Other	See 1	See 2, 3	See 1	See 1, 2	See 1
INTERFACING Interface	Parallel Computer	Parallel Computer	Parallel Computer	Parallel Computer	Parallel Computer
Transmission Rate	—	9 Mbps	9 Mbps	3 2 Mbps	—
Mode	—	—	—	—	—
Other	—	—	IBM 2701	IBM 2701	Honeywell Series 16
INTERNAL PROCESSOR	See 2	Hazeltine 20K X 32	Hazeltine 32K X 16	Hazeltine 16K X 48	—
OPTIONS	—	PC	PC	DD	LP, PLT, PT, MT, DD
PRICING	$60,000	—	—	—	—
REMARKS & OTHER FEATURES	[1]Clipping Divider, Color & Stereo Display, Char Generators (opt) [2]Controller Uses Host Computer Memory	[1]Random Selective Update/Erase of Char & Lines [2]Storage of 8 Programmable Graphs [3]10 Displays/System	[1]20 Displays/System 2 Interactive	[1]8 Displays/System [2]1975 Stored Backgrounds for Call Up	[1]HW Char, Plotting, Circle, Line Generators (opt)
IPAD TYPE	5	2 or 3	3	3	3

ABBREVIATIONS OPTIONS PC - Printer/Copier (Thermal, Optical, Electrostat) • LP-Line Printer • PLT - Plotter • PT - Paper Tape • MT - Magnetic Tape Transport • DD - Disk Drive
OTHER ST - Storage Tube Display • HW – Hardware Feature

*Table obtained from Reference 16; unmodified except as noted by shaded areas

TABLE 5-3.*
GRAPHIC CRT TERMINALS, Contd

ORIGINAL PAGE IS
OF POOR QUALITY

COMPANY	IBM	IMLAC	INFORMATION DISPLAYS		LUNDY
MODEL	2250	PDS-1	IDIgraf	IDIIOM	System 32
DISPLAY Size (inches)	12 X 12	8 X 10	10 X 10	13 X 13	20 (round)
Char Capacity	3848	3200	2048	2048	6000
Char/Line	74	80, 128 (opt)	73	128	160
Total Lines	52	40	51	85	80
No Char & Code	63	96 ASCII EBCID	128 ASCII	128 ASCII - See 1	96/192 ASCII, EBCDIC
Char Generation	Stroke	7 X 9 Stroke	16 X 16 Stroke	7 X 9 Stroke	Stroke
Refresh Rate (Hz)	30	40	30	30/20	10 to 100
DATA ENTRY Insert/Delete	———	Char & Line (std)	Char & Line (opt)	Char & Line (std)	———
Tabulation	———	Horz & Vert (std)	Horz & Vert (opt)	Horz & Vert (std)	———
Formatting	———	std	opt	std	———
Page Roll	———	std	opt	std	———
Split Screen	———	std	opt	std	———
Other	———	———	———	———	———
GRAPHICS Visible Raster	1024 X 1024	1024 X 1026	1024 X 1024	1024 X 1024	2048 X 2048 4096 X 4096 (opt)
Max Component	———	1024 raster units	1024 raster units	1024 raster units	2048 raster units
Positioning Modes	Absolute & Relative	Absolute & Relative	Absolute & Relative	Absolute & Relative	Absolute & Relative
Vector Modes	Absolute & Relative	Relative	Absolute & Relative	Absolute & Relative	Absolute & Relative
View Manipulation	———	Shift (std)	Shift, Rotate (opt)	Shift, Rotate, 2D & 3D Zoom (std)	Shift & Rotate (std) 3D Zoom (opt)
Pointers	Light Pen	Light Pen Tablet Mouse	Light Pen, Tablet	Light Pen, Track-ball, Joystick	Light Pen, Tablet, Trackball, Joystick Mouse
Other	———	See 1	———	See 2	See 1, 2
INTERFACING Interface	Parallel Computer	RS 232 B Parallel Computer	RS 232 B Current Loop	Parallel Computer	RS 232 B Parallel Computer
Transmission Rate	———	1 6 Mbps	5 Kbps	50 Kbps	4800 bps
Mode	———	Half/Full-Duplex Echoplex	Full-Duplex	Full-Duplex	———
Other	IBM 1130	TTY & IBM Compatible	———	———	IBM 2250 Compatible
INTERNAL PROCESSOR	IBM 2K X 16 4K Expandable	IMLAC 4K X 16 32K Expandable	ID 1K X 10 8K Expandable	Varian 620/f 4K X 16 32K Expandable	Lundy 4K X 16 32K Expandable
OPTIONS	PC, LP, MT, DD	PC, LP, PT, MT, DD	———	PC, LP, PT, MT, DD	LP, PT, MT
PRICING	$120,000	$9,620	$8,000	$65,000	$48,000
REMARKS & OTHER FEATURES	———	[1]Software Controlled Graphics (std)	Up to 4 Displays per Controller	[1]Plus 64 Programmable [2]HW Char, Circle Generators	[1]HW Char, Circle, Ellipse, Rectangle, Dot/Dash Generators [2]Stand Alone or Multi-Display Systems
IPAD TYPE	4 or 5	4 or 5	4	4 or 5	5

ABBREVIATIONS OPTIONS PC - Printer/Copier (Thermal, Optical, Electrostat) • LP-Line Printer • PLT - Plotter • PT - Paper Tape • MT - Magnetic Tape Transport • DD - Disk Drive
OTHER ST - Storage Tube Display • HW — Hardware Feature

*Table obtained from Reference 16; unmodified except as noted by shaded areas

TABLE 5-3.*
GRAPHIC CRT TERMINALS, Contd

COMPANY	SANDERS	SYSTEMS ENGR. LABS		TEKTRONIK	UNIVAC
MODEL	ADDS/900-960	80 816	80-821	T-4002A	1557/1558
DISPLAY Size (inches)	14 X 14	10 X 10	12 3 X 12.3	8 3 X 6 5	12 X 12
Char Capacity	—	2380	2380	3315	—
Char/Line	112	85	85	85	—
Total Lines	74	64	64	40	—
No Char & Code	128 ASCII	128 ASCII	128 ASCII	ASCII	64 ASCII
Char Generation	Stroke	5 X 7 Stroke	5 X 7 Stroke	7 X 9 Dot	Stroke
Refresh Rate (Hz)	60	30/60	—	ST - See 1	—
DATA ENTRY Insert/Delete	—	—	—	Char & Line (std)	—
Tabulation	—	—	—	Horz (std)	—
Formatting	—	—	—	std	—
Page Roll	—	—	—	—	—
Split Screen	—	—	—	std	—
Other	—	—	—	See 1	—
GRAPHICS Visible Raster	1024 X 1024	1024 X 1024	1024 X 1024	1024 X 1024	1024 X 1024
Max Component	—	—	—	1024 raster units	1024 raster units
Positioning Modes	Absolute & Relative	Absolute	Absolute	Absolute & Relative	Absolute & Relative
Vector Modes	Absolute & Relative	Relative	Absolute	Absolute & Relative	Absolute & Relative
Manipulation	Shift, Rotate (opt)	—	—	—	—
Pointers	Light Pen Track ball, Joystick	Light Pen Trackball	Light Pen Trackball	Joystick. Tablet Mouse	Light Pen
Other	See 1	—	—	—	See 1
INTERFACING Interface	RS 232 B	Parallel Computer	Parallel Computer	RS-232 B	RS 232 B
Transmission Rate	—	5 Mbps	5 Mbps	—	—
Mode	—	—	—	—	—
Other	IBM 360	SEL 800	SEL 800	See 2	UNIVAC 1100
INTERNAL PROCESSOR	Varian 620/i 4K X 16 32K Expandable	—	—	—	UNIVAC 1557 8K X 18 16K Expandable
OPTIONS	PT	—	—	PC	—
PRICING	—	$26 000	$30,000	$8,800	—
REMARKS & OTHER FEATURES	[1]HW Char Vector Conic Generators	—	—	[1]Refresh Scratch Pad Areas @ 40 Hz for 85 Char [2]TTY-Port Interfaces for Minicomputers, IBM 360	[1]Controller Drives 1 to 3 Displays
IPAD TYPE	5	5	3 or 4	4	4 or 5

ABBREVIATIONS OPTIONS PC - Printer/Copier (Thermal, Optical, Electrostat) • LP Line Printer • PLT Plotter • PT Paper Tape • MT Magnetic Tape Transport • DD Disk Drive
OTHER ST - Storage Tube Display • HW — Hardware Feature

*Table obtained from Reference 16, unmodified except as noted by shaded areas

TABLE. 5-3
GRAPHIC CRT TERMINALS, Concluded

COMPANY	VECTOR GENERAL	XEROX
MODEL	Graphics Display	7580
DISPLAY Size (inches)	13 X 14	10 X 10
Char Capacity	7200	———
Char/Line	120/80/60/32	———
Total Lines	60/40/30/16	———
No Char & Code	192 ASCII	64 ASCII
Char Generation	3 X 2 Stroke	Stroke
Refresh Rate	30	———
DATA ENTRY Insert/Delete	Char & Line (std)	———
Tabulation	Horz & Vert (std)	———
Formatting	std	———
Page Roll	opt	———
Split Screen	opt	———
Other	———	———
GRAPHICS Visible Raster	4096 X 4096	1024 X 1024
Max Component	4096 raster units	1024 raster units
Positioning Modes	Absolute & Relative	———
Vector Modes	Absolute & Relative	———
View Manipulation	Shift Rotate 2D & 3D Zoom (opt)	———
Pointers	Light Pen, Tablet Mouse	Light Pen
Other	See 1	———
INTERFACING Interface	Parallel Computer	———
Transmission Rate	———	———
Mode	———	———
Other	———	Sigma 5 & 7
INTERNAL PROCESSOR	4K X 16 128K Expandable	———
OPTIONS	PC LP PT, MT, DD	———
PRICING	$19,800	———
REMARKS & OTHER FEATURES	[1] 16 Intensity Levels	———
IPAD TYPE	4 or 5	4

ABBREVIATIONS
OPTIONS PC - Printer/Copier (Thermal Optical Electrostat) • LP Line Printer • PLT - Plotter • PT - Paper Tape • MT - Magnetic Tape Transport • DD - Disk Drive
OTHER ST - Storage Tube Display • HW — Hardware Feature

* [illegible]

Vector General's, IDI's, Lundy's, etc. display hardware is meant to be used with a variety of current, inexpensive mini-computers. The nature of human inter-action with an expensive maxicomputer — with its precious core resources — dictates that the program in the host computer be rolled out to a secondary storage device whenever it is waiting for a user interaction. Each time the program must be rolled back in to accom-plish a user request requires system overhead. The highest overhead is re-quired when the interactive program has to be rolled in for every user interaction. Clearly there will be occasions when the user can take a number of actions (e.g., point to a number of items to be deleted from the display) before he needs to have response from the host computer. The advent of the inexpensive minicomputer has provided this stringing of requests and cut down main computer operation overhead.

An extension of this same philosophy would assign the minicomputer at the ter-minal the role of more generalized file ma-nipulation and only use the host computer for major computations or occasional ac-cess to a major data base. For example, the total results of a computation may be sent to the minicomputer, stored on its local mass storage memory, and then view-ed at the user's leisure without disturbing the host computer. Similarly, a structural model may be viewed from different orien-tations by doing the appropriate coordinate transformational computations on the mini-computer rather than on the host maxicom-puter. Local text editing of a text file is another example of this advantageous div-ision of tasks.

Table 5-4 presents an excellent summary of minicomputer characteristic (including price) obtained from Reference 17. This table was also reproduced with the kind permission of MODERN DATA. The three basic types of computer and terminal system configurations are shown in Figure 5-3. As shown at the top of Figure 5-3, the host computer handles all the interactive and display processing. In mid-figure, the mini (or midi) computer is basically a communications processor and handles some of the display processing. At the bottom, all the display processing and all (or most all) the interrupt processing is handled by the minicomputer.

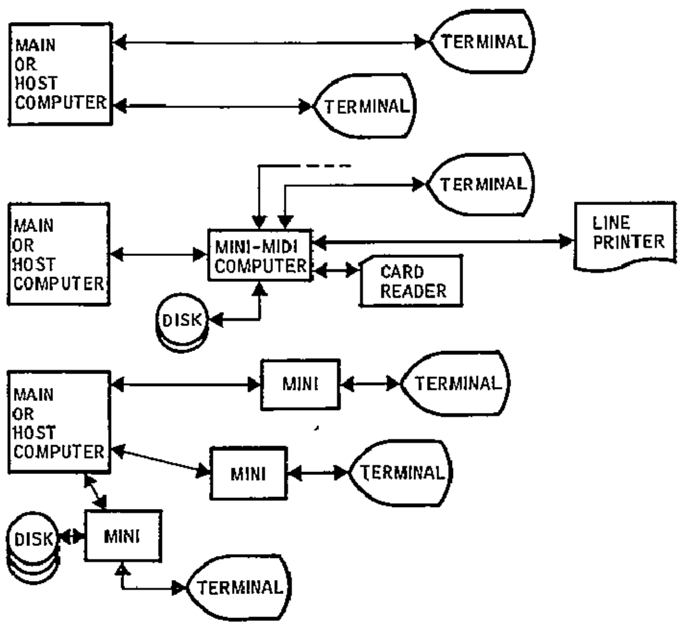


Figure 5-3. Terminal System Configurations

In the following subsections, system configurations are illustrated for Type 4 and 5 devices. The section is concluded with an investigation into target prices necessary to encourage widespread use of such terminals.

5.3.1 Typical Type 4 terminal configurations. - This type of terminal could be configured with the host computer in two basic ways, as shown in Figure 5-4. The actual connection to the host computer may be directly into the multiplexer (MUX) or the data channel where a remote location for the terminals is not needed. A typical configuration, e.g., that given in Figure 5-4a, might be with the terminal hardware given in Table 5-5.

A typical configuration with a minicomputer - as shown in Figure 5-4b - is given in Table 5-6.* This configuration typically has many options such as:

1. Additional 4K memory @ $3800.
2. Additional interactive display monitor and keyboard @ $3550.
3. Programmer maintenance control panel @ $2350.
4. Additional (14") slave display monitor @ $2150.
5. Storage tube interface and display (Tektronix 611 or equivalent) @ $3800.
6. Plotter interface (Calcomp 565 or equivalent) @ $450.

Useful options can add upwards to $5000 to the approximate purchase price shown in Table 5-6.

* Data for these and subsequent tables were principally obtained from Tables 5-3 and 5-4.

TABLE 5-4.*
MINICOMPUTER CHARACTERISTICS

COMPANY	ATRON	CINCINNATI MILACRON	CLARY DATACOMP	COMPUTER AUTOMATION		
MODEL	501 Datamanager	CIP/2100	CDS 404	ALPHA-8/NAKED MINI-8	108	208
MEMORY Word Size (bits)	8	8 or 9	16	8	8	8
Memory Size (words)	4K to 32K	4K to 32K	1K to 65K	4K to 32K	4K to 16K	4K to 16K
Cycle Time (μsec)	2 00	1 10	2 00	1 60	1 60	2 60
Parity Check	Option	Option	———	Option	———	———
Memory Protect	Option	Option	Option	Option	Option	Option
Direct Addressing (words)	32K	32K	1K	512	512	512
Indirect Addressing	Single Level	Multi Level	Multi Level	Multi Level	Multi Level	Multi Level
CPU Registers	Variable	15 Gen Purpose 1 Index	4 Gen Purpose 2 Index	1 Gen Purpose	1 Gen Purpose	1 Gen Purpose
Hardware Multiply-Divide	———	Standard	Standard	———	———	———
Immediate Instructions	Standard	Standard	Standard	———	———	———
Double Word Instructions	———	Standard	Standard	Standard	Standard	Standard
Byte Processing	Standard	Standard	Option	Standard	Standard	Standard
INPUT/OUTPUT I/O Word Size (bits)	8 + Parity	16 in 2 bytes	16/32/48/64	8	8	8
Priority Interrupt Levels	Variable	8 to 64	16	3	3	3
Direct Memory Access Channel	Option	Option	Option	Standard	Standard	Standard
I/O Maximum Word Rate (word/sec)	500 kHz	900 kHz (bytes)	———	120 kHz	120 kHz	68 kHz
OTHER FEATURES Real Time Clock	Option	Option	Option	Option	Option	Option
Power Fail/Restart	Standard	Option	Option	Option	Option	Option
Peripheral Device Options	DD MT, PT PC, LP, TP	DD, MT, CT, PT, LP, TP, CRT	MT, CT, PT, LP, TP	MT, CT, PT, PC, LP, TP, CRT, PLT	MT, CT, PT, PC, LP TP, CRT	MT, CT, PT, PC LP TP, CRT
Software	Assembler	———	Basic	———	———	———
PRICE Computer with Basic Memory	$7,475	$4,565	$8,000	$2,800 (Alpha 8) $1,975 (Naked Mini 8)	$5,490	$5,190
Add On Memory Increment	$1,250/4K	$1,700/4K X 8 $1,800/4K X 9	$3,500/4K	$1,700/4K	$2,600/4K	$2,600/4K
Abbreviations	• DD — Disk Drives • DRD — Drum Drives • MT — Mag Tape Transports • CT — Cassette/Cartridge Transports • PT — Paper Tape Equip • PC — Punch Card Equip • LP — Line/Page Printers • TP — Teleprinters • CRT — CRT Displays • PLT — Plotters					

* Table obtained from Reference 17.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 5-4.*
MINICOMPUTER CHARACTERISTICS, Contd

COMPANY	COMPUTER AUTOMATION (Cont'd)				COMPUTER LOGIC SYSTEMS	CONTROL DATA
MODEL	808	ALPHA-16/NAKED MINI-16	116	216	CLS-18	SC-1700
MEMORY Word Size (bits)	8	16	16	16	18	18
Memory Size (words)	4K to 16K	2K to 32K	4K to 32K	4K to 32K	4K to 265K	4K to 32K
Cycle Time (μsec)	8 00	1 60	1 60	2 60	0 96	1 50
Parity Check	———	Option	———	———	———	Standard
Memory Protect	Option	Option	Option	Option	Standard	Standard
Direct Addressing (words)	512	1K	1K	1K	512	32K
Indirect Addressing	Multi Level	Multi Level	Multi Level	Multi Level	Multi Level	Multi Level
CPU Registers	1 Gen Purpose ———	2 Gen Purpose 1 Index	2 Gen Purpose 1 Index	2 Gen Purpose 1 Index	8 to 32 Gen Purpose 4 to 16 Index	9 Gen Purpose 2 Index
Hardware Multiply-Divide	———	Standard	Standard	Standard	Option	Standard
Immediate Instructions	———	Standard	Standard	Standard	Standard	Standard
Double Word Instructions	Standard	Standard	Standard	Standard	Standard	———
Byte Processing	Standard	———	———	———	Standard	Option
INPUT/OUTPUT I/O Word Size (bits)	8	16	16	16	18	16 + Parity & Protect
Priority Interrupt Levels	3	3	3	3	8	16
Direct Memory Access Channel	Standard	Standard	Standard	Standard	Standard	Option
I/O Maximum Word Rate (word/sec)	25 kHz	714 kHz/625 kHz	714 kHz	714 kHz	1 0 MHz	300 kHz
OTHER FEATURES Real Time Clock	Option	Option	Option	Option	Option	Option
Power Fail/Restart	Option	Option	Option	Option	Option	Standard
Peripheral Device Options	MT CT, PT, PC, LP, TP CRT	All Types	All Types	All Types	———	DD, DRD, MT, PT, PC, LP, TP, CRT
Software	———	Basic Fortran	Basic, Fortran	Basic, Fortran	Assembler	Fortran, Autran
PRICE Computer with Basic Memory	$4 990	$3,550/4K (Alpha-16) $2,500/4K (Naked Mini)	$8 490	$7 990	$9 870	$15,900
Add On Memory Increment	$2 700/4K	$2,200/4K	$3 800/4K	$3,800/4K	$3,200/4K	$4,500/4K
Abbreviations	• DD — Disk Drives • DRD — Drum Drives • MT — Mag Tape Transports • CT — Cassette/Cartridge Transports • PT — Paper Tape Equip • PC — Punch Card Equip • LP — Line/Page Printers • TP — Teleprinters • CRT — CRT Displays • PLT — Plotters					

*Table obtained from Reference 17.

TABLE 5-4.★
MINICOMPUTER CHARACTERISTICS, Contd

COMPANY	DATA GENERAL			DATACRAFT	DATAMATE COMPUTER SYSTEMS	
MODEL	NOVA 1200	NOVA 800	SUPERNOVA SC	DC 6024/5	DM-16	DM-70
MEMORY Word Size (bits)	16	16	16	24	16	16
Memory Size (words)	2/4K to 32K	2/4K to 32K	4K to 16K	4K to 32K	8K to 32K	4K to 32K
Cycle Time (μsec)	1 20	0 80	0 30	1 20	1 00	1 00
Parity Check	———	———	———	Standard	———	———
Memory Protect	———	———	———	Option	———	———
Direct Addressing (words)	1K	1K	1K	32K	512	1K
Indirect Addressing	Multi Level	Multi Level	Multi Level	Multi Level	Multi Level	Multi Level
CPU Registers	4 Gen Purpose 2 Index	4 Gen Purpose 2 Index	4 Gen Purpose 2 Index	5 Gen Purpose 3 Index	2 Gen Purpose 1 Index	4 Gen Purpose 2 Index
Hardware Multiply Divide	Option	Option	Option	Standard	Standard	Option
Immediate Instructions	———	———	———	Standard	Standard	Standard
Double Word Instructions	———	———	———	Standard	Standard	———
Byte Processing	Standard	Standard	Standard	Standard	Standard	Standard
INPUT/OUTPUT I/O Word Size (bits)	16	16	16	8/24	16	16
Priority Interrupt Levels	16	16	16	16	8 to 64	64
Direct Memory Access Channel	Standard	Standard	Standard	Option	Standard	Standard
I/O Maximum Word Rate (word/sec)	833 kHz	1 25 MHz	1 25 MHz	833 kHz	1 0 MHz	1 0 MHz
OTHER FEATURES Real Time Clock	Option	Option	Option	Option	Option	Option
Power Fail/Restart	Option	Option	Option	Option	Standard	Standard
Peripheral Device Options	DD MT PT PC, LP TP PLT	DD MT PT PC LP TP, PLT	DD, MT, PT, PC LP TP PLT	DD MT PT, LP TP PLT	DD MT CT PT PC, LP TP CRT PLT	DD, MT CT PT PC LP TP CRT PLT
Software	Algol Basic Fortran	Algol Basic Fortran	Algol Basic Fortran	Fortran	Fortran	———
PRICE COMPUTER with Basic Memory	$5 100	$6 600	$11 500	$15 500	$14 900	$8 500
Add On Memory Increment	$2 200/2K 2,700/4K	$2,500/2K $3 000/4K	$2 800/1K (SC) $3 650/2K (SC) $5 950/4K (SC) $3 650/4K (core)	$4 800/4K	$4 000/4K	$2 700/4K
Abbreviations	• DD – Disk Drives • DRD – Drum Drives • MT – Mag Tape Transports • CT – Cassette/Cartridge Transports • PT – Paper Tape Equip • PC – Punch Card Equip • LP – Line/Page Printers • TP – Teleprinters • CRT – CRT Displays • PLT – Plotters					

Table obtained from Reference 17.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 5-4.*
MINICOMPUTER CHARACTERISTICS, Contd

COMPANY	DIGITAL COMPUTER CONTROLS				DIGITAL EQUIPMENT	
MODEL	D-112	D-112H	D-116	D 216	PDP 8/I	PDP-8/L
MEMORY Word Size (bits)	12	12	16	16	12	12
Memory Size (words)	4K to 32K	4K to 32K	4K to 32K	4K to 32K	4K to 32K	4K to 8K
Cycle Time (μsec)	1 20	0 30	1 20	1 20	1 50	1 60
Parity Check	Option	Option	———	———	Option	Option
Memory Protect	Standard	Standard	Option	Option	Option	Option
Direct Addressing (words)	256	256	32K	32K	256	256
Indirect Addressing	Single Level	Single Level	Single Level	Single Level	Single Level	Single Level
CPU Registers	2 Gen Purpose 8 Auto Index	2 Gen Purpose 24 Auto Index	4 Gen Purpose 4 Index	8 Gen Purpose 8 Index	4 Gen Purpose 8 Auto Index	4 Gen Purpose 8 Auto Index
Hardware Multiply Divide	Option	Option	Option	Option	Option	———
Immediate Instructions	———	———	Standard	Standard	———	———
Double Word Instructions	Option	Standard	———	Standard	———	———
Byte Processing	———	Standard	———	Standard	———	———
INPUT/OUTPUT I/O Word Size (bits)	12	12	16	16	6	6
Priority Interrupt Levels	1	1	16	4	4	4
Direct Memory Access Channel	Option	Option	Standard	Standard	Standard	Standard
I/O Maximum Word Rate (word/sec)	833 kHz	3 3 MHz	833 kHz	833 kHz	666 kHz	625 kHz
OTHER FEATURES Real Time Clock	Option	Option	Option	Option	Option	Option
Power Fail/Restart	Option	Option	Option	Option	Option	Option
Peripheral Device Options	DD, DRD, MT, CT PT LP TP, PLT	DD, DRD, MT, CT, PT, LP, TP PLT	DD, DRD, MT, CT, PT, LP, TP, PLT	DD DRD MT CT PT, LP, TP, PLT	All Types	All Types
Software	Algol, Basic, Cobol Fortran	Algol Basic, Cobol, Fortran	Algol Basic Cobol Fortran	Algol Basic Cobol Fortran	Algol Basic Cobol Fortran	Algol, Basic Cobol Fortran
PRICE Computer with Basic Memory	$3 990	$6,800	$4,000	$4 300	$12 800*	$8,500*
Add On Memory Increment	$2 700/4K	$700/256	$2,700/4K	$2 700/4K	$4,000/4K	$4 000/4K
OTHER REMARKS					*with ASR 33	*with ASR 33
Abbreviations	• DD – Disk Drives • DRD – Drum Drives • MT – Mag Tape Transports • CT – Cassette/Cartridge Transports • PT – Paper Tape Equip • PC – Punch Card Equip • LP – Line/Page Printers • TP – Teleprinters • CRT – CRT Displays • PLT – Plotters					

*Table obtained from Reference 17.

TABLE 5-4.* MINICOMPUTER CHARACTERISTICS, Contd

COMPANY	DIGITAL EQUIPMENT (Cont'd)				DIGITAL SCIENTIFIC	ELECTRONIC ASSOCIATES
MODEL	PDP-8/E	PDP-11/20	PDP-11/15	PDP-15	META 4-16/4001	EAI 640
MEMORY Word Size (bits)	12	16	16	18	18	16
Memory Size (words)	4K to 32K	1K to 124K	1K to 32K	4K to 32K	4K to 65K	8K to 32K
Cycle Time (μsec)	1 20/1 40	0 95	0 95	0 80	0 90	1 65
Parity Check	Option	Option	Option	Option	Standard	————
Memory Protect	Option	————	————	Option	Standard	Standard
Direct Addressing (words)	256	32K	32K	————	65K	512
Indirect Addressing	Single Level	Multi Level	Multi Level	————	Single Level	Multi Level
CPU Registers	5 Gen Purpose 8 Auto Index	8 Gen Purpose 8 Index	8 Gen Purpose 8 Index	20 Gen Purpose 1 Index	32 Gen Purpose ————	8 Gen Purpose 1 Index
Hardware Multiply-Divide	————	Option	Option	Option	Standard	Standard
Immediate Instructions	————	Standard	————	————	Standard	Standard
Double Word Instructions	————	————	————	————	Standard	Standard
Byte Processing	————	Standard	Standard	————	————	————
INPUT/OUTPUT I/O Word Size (bits)	6	16	16	18	16	16
Priority Interrupt Levels	12	4	1	4	Variable	64
Direct Memory Access Channel	Standard	Standard	Standard	Standard	Standard	Standard
I/O Maximum Word Rate (word/sec)	833 kHz	2 5 MHz	2 5 MHz	1 0 MHz	1 0 MHz	600 kHz
OTHER FEATURES Real Time Clock	Option	Option	Option	Option	Option	Option
Power Fail/Restart	Option	Standard	Option	Option	————	Standard
Peripheral Device Options	All Types	All Types	All Types	All Types	DD DRD MT PT LP TP PLT	DD, MT CT, PT, LP, TP, CRT, PLT
Software	Algol Basic Cobol Fortran	Basic Fortran	Basic, Fortran	Algol Fortran	Fortran	Basic, Fortran
PRICE Computer with Basic Memory	$6 500*	$10 800*	$6 200	$16 500	$25,000* & 2K ROM	$24,500*
Add On Memory Increment	$3 000/4K	$3 500/4K	$3 500/4K	$8 000/4K	$5 550/4K	$15 000/8K
OTHER REMARKS	*with ASR 33	*with ASR-33			*with I/O Typer	*with ASR 33
Abbreviations	• DD – Disk Drives • DRD – Drum Drives • MT – Mag Tape Transports • CT – Cassette/Cartridge Transports • PT – Paper Tape Equip • PC – Punch Card Equip • LP - Line/Page Printers • TP – Teleprinters • CRT – CRT Displays • PLT – Plotters					

*Table obtained from Reference 17.

TABLE 5-4.★
MINICOMPUTER CHARACTERISTICS, Contd

COMPANY	ELECTRONIC PROCESSORS	GENERAL AUTOMATION			GRI COMPUTER	
MODEL	EPI-118	SPC-12	SPC-16	18/30 System	GRI-909/10 & 20	GRI 909/30 & 40
MEMORY Word Size (bits)	18	8	16	18	16	16
Memory Size (words)	4K to 32K	4K to 32K	4K to 32K	4K to 32K	1K to 4K	4K to 32K
Cycle Time (μsec)	0 90	0 60	0 96	0 96	1 30	1 30
Parity Check	—	—	—	Standard	—	—
Memory Protect	Standard	—	—	Standard	Option	Option
Direct Addressing (words)	32K	16K	32K	32K	4K	32K
Indirect Addressing	Single Level	Single Level	Multi Level	Multi Level	Single Level	Single Level
CPU Registers	2 Gen Purpose —	6 Gen Purpose 3 Index	16 Gen Purpose 3 Index	20 Gen Purpose 3 Index	2/8 Gen Purpose —	2/8* Gen Purpose —
Hardware Multiply-Divide	Option	—	Option	Standard	Option	Option*
Immediate Instructions	—	Standard	—	Standard	Standard	Standard
Double Word Instructions	—	Standard	—	Standard	Standard	Standard
Byte Processing	Standard	—	Standard	—	Option	Option
INPUT/OUTPUT I/O Word Size (bits)	21	12	16	16	16	16
Priority Interrupt Levels	18	1	64	61	—	—
Direct Memory Access Channel	Standard	Option	Standard	Standard	Standard	Standard
I/O Maximum Word Rate (word/sec)	900 kHz	460 kHz	520 kHz	833 kHz	568 kHz	568 kHz
OTHER FEATURES Real Time Clock	Option	Standard	Standard	Standard	Standard*	Option
Power Fail/Restart	Standard	Standard	Standard	Standard	Standard*	Option
Peripheral Device Options	DD MT, CT PT LP, CRT PLT	MT, CT PT LP, TP, CRT PLT	All Types	All Types	DD, MT, CT PT LP TP CRT	DD, MT, CT PT LP TP CRT
Software	Basic	Fortran	Fortran	Fortran	Assembler	Assembler
PRICE Computer with Basic Memory	$5 900	$3,700 (est)	$9,500 (est)	$18 000 (est)	$3,500 (10) $3 950 (20)	$5 650 (30) $6 850 (40)
Add On Memory Increment	$2 650/4K	$1,800/4K (est)	$3 600/4K (est)	$4 000/4K (est)	$995/1K	$2 950/4K
OTHER REMARKS					*Option on 20	*Standard on 40
Abbreviations	• DD — Disk Drives • DRD — Drum Drives • MT — Mag Tape Transports • CT — Cassette/Cartridge Transports • PT — Paper Tape Equip • PC — Punch Card Equip • LP — Line/Page Printers • TP — Teleprinters • CRT — CRT Displays • PLT — Plotters					

★Table obtained from Reference 17

TABLE 5-4.*
MINICOMPUTER CHARACTERISTICS, Contd

COMPANY	HEWLETT PACKARD			HONEYWELL		INTERDATA
MODEL	2114B	2116C	2100A	H316	DDP-516	Model 1
MEMORY Word Size (bits)	16	16	16	16	16	8
Memory Size (words)	4K to 16K	8K to 32K	4K to 32K	4K to 32K	4K to 32K	2K to 16K
Cycle Time (μsec)	2 00	1 60	0 98	1 60	0 96	1 00
Parity Check	Option	Option	Standard	Option	Option	Option
Memory Protect	———	Option	Standard	Option	Option	———
Direct Addressing (words)	2K	2K	2K	1K	1K	512
Indirect Addressing	Multi Level	Multi Level	Multi Level	Multi Level	Multi Level	Single Level
CPU Registers	2 Gen Purpose	2 Gen Purpose	2 Gen Purpose	2 Gen Purpose 1 Index	2 Gen Purpose 1 Index	1 Gen Purpose 8K Index
Hardware Multiply-Divide	———	Option	Standard	Option	Option	———
Immediate Instructions	———	———	———	———	———	Standard
Double Word Instructions	———	Option	Standard	Option	Option	Standard
Byte Processing	———	———	———	Standard	Standard	Standard
INPUT/OUTPUT I/O Word Size (bits)	16	16	16	16	16	8
Priority Interrupt Levels	56	48	56	49	49	255
Direct Memory Access Channel	Option	Option	Option	Option	Option	Option
I/O Maximum Word Rate (word/sec)	500 kHz	633 kHz	1 1 MHz	312 KHz	1 0 MHz	1 0 MHz
OTHER FEATURES Real Time Clock	Option	Option	Option	Option	Option	Standard
Power Fail/Restart	Option	Option	Standard	Option	Option	Option
Peripheral Device Options	All Types	All Types	All Types	All Types	All Types	All Types
Software	Algol Basic Fortran	Algol Basic Fortran	Algol, Basic Fortran	Basic, Fortran	Basic Fortran	Fortran
PRICE Computer with Basic Memory	$7 000	$14 000	$6 900	$8 400	$23 800	$3 750
Add-On Memory Increment	$3 500/4K	$8 000/8K	$3,500/4K	$3 500/4K	$8 000/4K	$900/2K
Abbreviations	• DD — Disk Drives • DRD — Drum Drives • MT - Mag Tape Transports • CT — Cassette/Cartridge Transports • PT — Paper Tape Equip • PC — Punch Card Equip • LP — Line/Page Printers • TP — Teleprinters • CRT — CRT Displays • PLT — Plotters					

*Table obtained from Reference 17.

ORIGINAL PAGE IS OF POOR QUALITY

TABLE 5-4.*
MINICOMPUTER CHARACTERISTICS, Contd

COMPANY	INTERDATA (Cont'd)	LOCKHEED ELECTRONICS		MODULAR COMPUTER SYSTEMS		
MODEL	Model 5	MAC Jr	MAC 16	MODCOMP I	MODCOMP II	MODCOMP III
MEMORY Word Size (bits)	16	16	16	16	16	16
Memory Size (words)	4K to 65K	4K to 65K	4K to 65K	4K to 16K	4K to 32K	4K to 64K
Cycle Time (μsec)	1 00	1 00	1 00	0 80	0 80	0 80
Parity Check	Option	Option	Option	Option	Option	Standard
Memory Protect	Option	Option	Option	————	————	Option
Direct Addressing (words)	65K	1K	1K	16K	32K	64K
Indirect Addressing	————	Multi Level	Multi Level	Single Level	Single Level	Single Level
CPU Registers	16 Gen Purpose 15 Index	1 Gen Purpose 4 to 16 Index	1 Gen Purpose 8 to 64 Index	3 Gen Purpose 3 Index	15 Gen Purpose 7 Index	15 Gen Purpose 7 Index
Hardware Multiply-Divide	Standard	Option	Option	————	Option	Option
Immediate Instructions	Standard	Standard	Standard	Standard	Standard	Standard
Double Word Instructions	Standard	Standard	Standard	————	Option	Standard
Byte Processing	Standard	Standard	Standard	Standard	Standard	Standard
INPUT/OUTPUT I/O Word Size (bits)	8/16	8/16	8/16	16	16	16
Priority Interrupt Levels	255	4 to 16	8 to 64	1	8	32
Direct Memory Access Channel	Option	Standard	Standard	Option	Option	Option
I/O Maximum Word Rate (word/sec)	500 kHz (bytes)	1 0 MHz	1 0 MHz	200 kHz	200 kHz	1 25 MHz
OTHER FEATURES Real Time Clock	Option	Option	Standard	Option	Option	Option
Power Fail/Restart	Option	Option	Standard	Option	Option	Standard
Peripheral Device Options	All Types	All Types	All Types	DD MT PT, PC LP TP	DD MT, PT PC, LP TP	DD MT PT PC LP TP
Software	Fortran	Fortran	Fortran	————	Fortran	Fortran
PRICE Computer with Basic Memory	$10 500	$7,900	$11 200	$5,200	$8,000	$13 500
Add On Memory Increment	$3,200/4K	$3 100/4K	$3,100/4K	$2 600/4K	$2 600/4K	$4 000/4K
OTHER REMARKS						*Option on 820/S
Abbreviations	• DD – Disk Drives • DRD – Drum Drives • MT – Mag Tape Transports • CT – Cassette/Cartridge Transports • PT – Paper Tape Equip • PC – Punch Card Equip • LP – Line/Page Printers • TP – Teleprinters • CRT – CRT Displays • PLT – Plotters					

*Table obtained from Reference 17.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 5-4.*
MINICOMPUTER CHARACTERISTICS, Contd

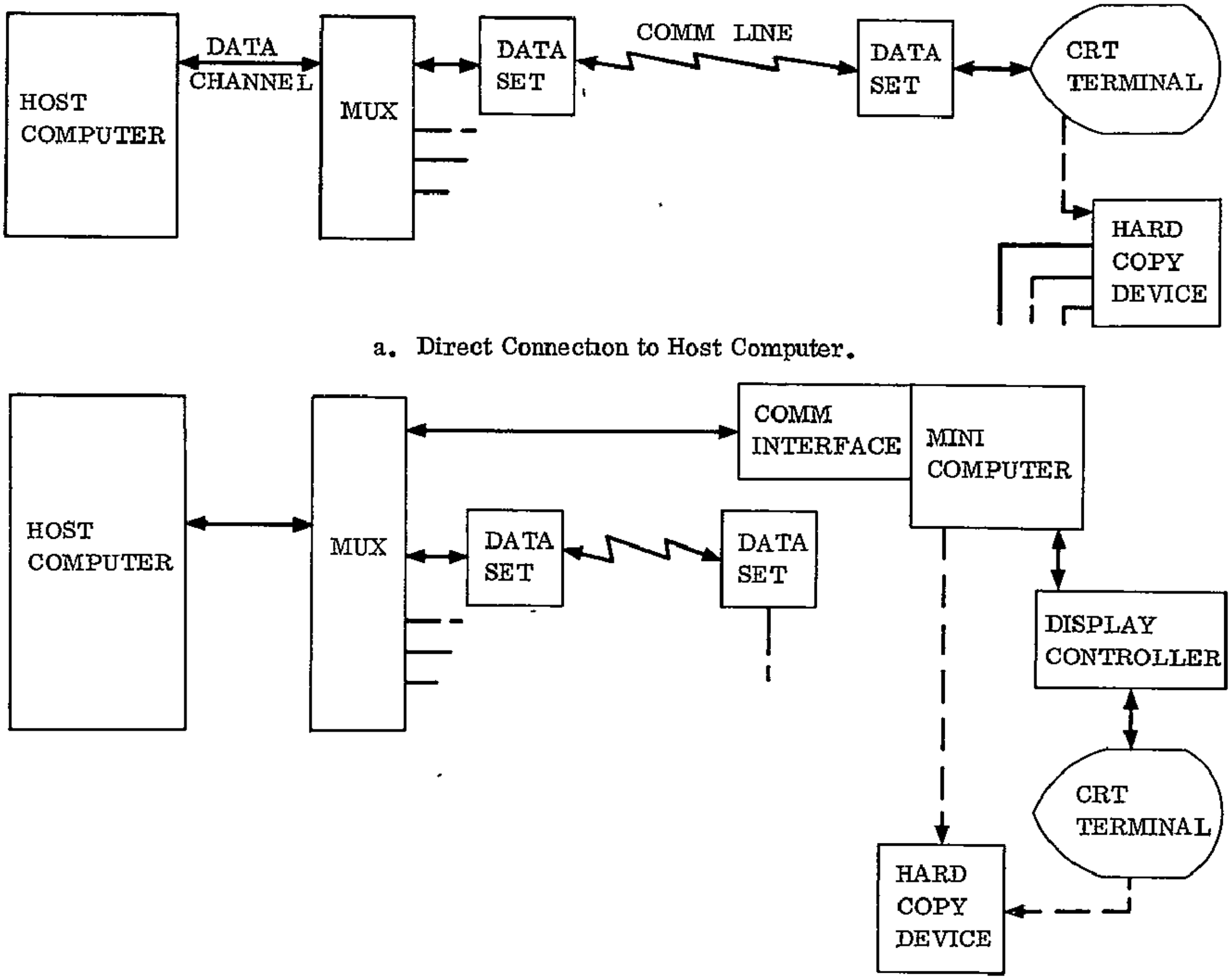
OMPANY	OMNICOMP COMPUTER	OMNITEC	PANASONIC		RAYTHEON DATA SYSTEMS		
IODEL	OMNUS-1	BIT 483	MC 7S	MC 7F	703	704	706
IEMORY 'ord Size (bits)	16	8	16	16	16	16	16
'emory Size (words)	2K/4K to 32K	8K to 65K	2K to 16K	4K to 16K	4K to 32K	4K to 32K	4K to 32K
ycle Time (μsec)	1 20	0 98	4 00	0 60	1 00	1 00	1 00
arity Check	———	Option	Standard	Standard	———	Option	Option
lemory rotect	Option	Standard	———	Standard	———	Option	Option
irect ddressing (words)	32K	512	256	256	2K	2K	2K
ndirect ddressing	———	Single Level	Single Level	Multi Level	———	———	———
PU Registers	2K Gen Purpose 2K Index	——— ———	1 Gen Purpose 3 Index	1 Gen Purpose 3 Index	1 Gen Purpose 1 Index	1 Gen Purpose 1 Index	1 Gen Purpose 1 Index
lardware Multiply-Divide	Option	Option	———	———	Option	Option	Option
mmediate nstructions	Standard	———	———	———	Standard	Standard	Standard
Double Word nstructions	Standard	Standard	———	———	Option	Option	Option
Byte Processing	Standard	Standard	———	———	Standard	Standard	Standard
NPUT/OUTPUT /O Word Size (bits)	8/16	8	16	16	16	16	16
Priority Interrupt Levels	32 to 256	8 to 32	———	———	16	16	16
Direct Memory Access Channel	Standard	Standard	Option	Standard	Option	Option	Option
I/O Maximum Word Rate (word/sec)	833 kHz	1 02 MHz	60 kHz	450 kHz	586 kHz	1 0 MHz	1 1 MHz
OTHER FEATURES Real Time Clock	Option	Option	Option	Standard	Option	Option	Option
Power Fail/Restart	Standard	Standard	Option	Standard	Option	Option	Option
Peripheral Device Options	DD, MT CT PT LP, TP, PLT	All Types	All Types	All Types	All Types	All Types	All Types
Software	———	Fortran	Basic Fortran	Basic Fortran	Fortran	Fortran	Fortran
PRICE Computer with Basic Memory	$5 300/2K $6 100/4K	$7 900	$7 500/4K (est)	$11,000/8K (est)	$12,750*	$8,000	$19 000*
Add On Memory Increment	$2,650/2K $3 450/4K	$4 500/8K	$3 000/4K (est)	$4,000/4K (est)	$5 000/4K	$3,500/4K	$6 600/4K
OTHER REMARKS					*with ASR-33		*with ASR 33
Abbreviations	• DD – Disk Drives • DRD – Drum Drives • MT – Mag Tape Transports • CT – Cassette/Cartridge Transports • PT – Paper Tape Equip • PC – Punch Card Equip • LP – Line/Page Printers • TP – Teleprinters • CRT – CRT Displays • PLT – Plotters						

*Table obtained from Reference 17.

TABLE 5-4.*
MINICOMPUTER CHARACTERISTICS, Concluded

COMPANY	VARIAN DATA MACHINES (Cont'd)			WESTINGHOUSE	XEROX DATA SYSTEMS
MODEL	620/f	620/i	620/L	2500	CF16A
MEMORY Word Size (bits)	16	16/18	16	16	16
Memory Size (words)	4K to 32K	4K to 32K	4K to 32K	4K to 65K	4K to 32K
Cycle Time (μsec)	0 75	1 80	1 80	0 75	1 60
Parity Check	Option	Option	————	Option	————
Memory Protect	Option	Option	Option	Option	Option
Direct Addressing (words)	2K	2K	2K	256	768
Indirect Addressing	Multi Level	Multi Level	Multi Level	Single Level	Multi Level
CPU Registers	2 Gen Purpose 2 Index	9 Gen Purpose 2 Index	4 Gen Purpose 2 Index	6 Gen Purpose 2 Index	2 Gen Purpose 1 Index
Hardware Multiply Divide	Option	Option	Option	Standard	Standard
Immediate Instructions	————	Standard	Standard	————	Standard
Double Word Instructions	Standard	Standard	Standard	Standard	Standard
Byte Processing	————	————	————	————	Standard
INPUT/OUTPUT I/O Word Size (bits)	16	16/18	16	16	16
Priority Interrupt Levels	64	64	64	120	66
Direct Memory Access Channel	Standard	Standard	Standard	Option	Option
I/O Maximum Word Rate (word/sec)	1 33 MHz	200 kHz	200 kHz	————	625 kHz
OTHER FEATURES Real Time Clock	Option	Option	Option	Option	Option
Power Fail/Restart	Option	Option	Option	Option	Option
Peripheral Device Options	Any Type	Any Type	Any Type	DD, MT, PT, LP, TP	DD, DRD, MT, PT, LP, TP, CRT, PLT
Software	Basic Fortran	Basic, Fortran	Basic, Fortran	Basic, Fortran	Fortran
PRICE Computer with Basic Memory	$10,500	$9,950	$5,400	$9 950	$7,990
Add On Memory Increment	$4 500/4K	$4,500/4K	$2 300/4K	$3,800/4K	$3 800/4K
Abbreviations	• DD – Disk Drives • DRD – Drum Drives • MT – Mag Tape Transports • CT – Cassette/Cartridge Transports • PT – Paper Tape Equip • PC – Punch Card Equip • LP – Line/Page Printers • TP – Teleprinters • CRT – CRT Displays • PLT – Plotters				

*Table obtained from Reference 17.



a. Direct Connection to Host Computer.

b. Connection to Host Computer via a Mini Computer.

Figure 5-4. Typical Type 4 Terminal Configurations

TABLE 5-5
TYPICAL TYPE 4 TERMINAL CONFIGURATION WITHOUT MINICOMPUTER

		Approx. Purchase Price
TEKTRONIX 4002A	Graphic Terminal	$8800
TEKTRONIX 4601	Hard Copy Unit	$3750
TEKTRONIX 4902	Interactive Graphics Unit	$ 750
TEKTRONIX 4951	Joystick	$ 600
		$13900

TABLE 5-6

TYPICAL TYPE 4 TERMINAL CONFIGURATION
WITH MINICOMPUTER

	Approx. Purchase Price **
IMLAC PDS-1 DISPLAY COMPUTER (4K memory)	$9620
HRC-1 High Contrast, High Resolution CRT	$ 290
LPA-2 Light Pen	$1645
HCY-1 Hard copy device	$5875
CBS-1 Cassette bulk store	$ 845
	$18275

Communications may be handled via specially conditioned telephone lines or direct connection to a data channel or multiplexer. (Refer to pertinent discussion in Subsection 4.3.1). At least 1200 baud data transfer rate is required and more than likely a 2400 or 4800 baud rate will be required to meet the demands of the amount of data to be transferred and the desired user response time. Oral reports from staff members at the Institute for Human Research at Stanford University indicate that response time effect on users running at 4800 and 9600 baud then reverting to a direct channel connection (> 50K baud) is such that the users don't want to go back to the "slow rates. Experience at Ford Philco in Newport Beach, and at Tektronix in Beaverton, Oregon, (as well as at Convair) indicates that 300 baud line rates allow only very simple displays to be put up and the slow response tends to frustrate the user even then.

Modems in the 2000 to 5000 baud rate range vary between $1000 to $6000, with median prices about $2000 for 2400 baud and $4500 for 4800 baud modems. These costs must be added to the terminal costs when remote locations are needed. The communication line costs for lines conditioned for the appropriate rates also must be added to the total system cost.

5.3.2 Typical Type 5 terminal configurations. - This type of terminal is the most sophisticated, most capable and, consequently, the most expensive. However, in situations where a remote job entry (RJE) capability is needed and several consoles or terminals can be supported together, the relative cost is reduced considerably.

** If manufacturer maintenance support is desired, the additional monthly cost is about $120.

A minicomputer (or buffer computer-controller) is always considered a part of this terminal because of the large amount of required display modification and manipulation plus the fact that some tasks, like the design/drafting function, can actually be done stand-alone (without recourse to the computational capability of the host computer). A typical configuration for this type terminal is shown in Figure 5-5.

Two distinct hardware implementations of the configuration are listed in Tables 5-7 and 5-8.

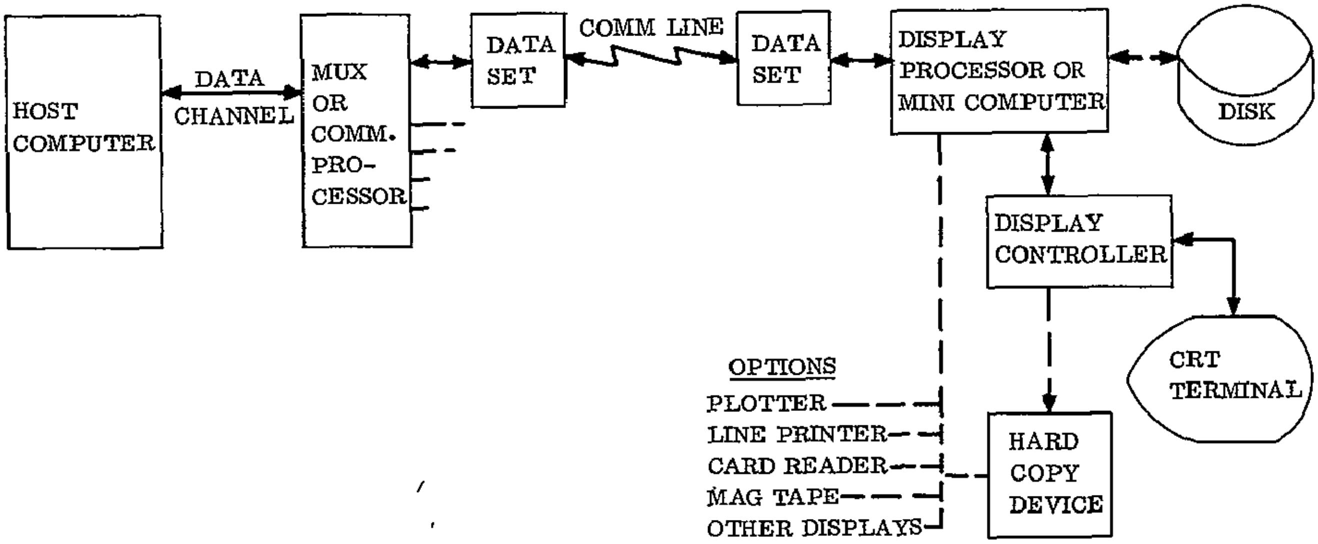


Figure 5-5. Typical Type 5 Terminal Configuration

TABLE 5-7

TYPICAL HARDWARE FOR A TYPE 5 TERMINAL CONFIGURATION: VECTOR GENERAL WITH A PDP-11 MINICOMPUTER

		Approx. Purchase Price
PDP 11/20	4K Memory, RT Clock, Auto Priority Interrupt	$10,800
	16K Additional Memory	$11,800
	Hardware Multiply/Divide	$ 2,650
	Full Duplex Modem Adapter	$ 700
2315	Disk Pack and Controller (3.5 M words)	$12,500
9 Track	Magnetic Tape Drive & Controller (25 IPS)	$ 6,500
DD3	High Speed Dual DAC Graphics Display	$25,000
DI1	Digital Device Interface	$ 500
KB1	Alphanumeric Keyboard	$ 1,200
LP2	Light Pen	$ 2,000
EC1	Expansion Chassis	$ 1,500
SS1	Subroutine/Stack Facility	$ 3,300
CG1	Character Generator	$ 4,000
		$82,450

TABLE 5-8

TYPICAL HARDWARE FOR A TYPE 5 TERMINAL CONFIGURATION:
CDC'S OFF-THE-SHELF 6000-1700 274 INTERACTIVE GRAPHICS SYSTEM

	Approx. Per Month Lease Price (1 yr)	Approx. Per Month Maintenance
S-C 1700 Computer (24K)	\$ 785	\$355
Interrupt Data Channel	\$ 47	\$ 17
Direct Storage Access	\$ 37	\$ 17
Buffered Data Channels (2)	\$ 460	\$ 66
TTY	\$ 125	\$ 38
Disk Controller	\$ 311	\$ 38
Disk Drive	\$ 281	\$ 52
Disk Pack	\$ 48	-
Graphics Controller (8K buffer)	\$1190	\$260
Graphics Display Console	\$1040	\$108
Data Set Controller	\$ 375	\$ 65
	\$4699	\$1016

Communications requirements for Type 5 terminals need not be much higher than for Type 4 terminals. An advantageous trade is possible here by providing more central memory and disk storage on the satellite minicomputer, lower data transmission rates and consequently lower costs. For an IPAD Type 5 installation where all of the display processing is to be done on the satellite mini, communication rates between 4800 and 9600 baud is sufficient for all but very large amounts of data transfer.

Modem and line costs are additional to the terminal costs shown in the tables. Modems for 4800 baud run about \$4500 purchase; 9600 baud modems may run as high as \$23,000. Where the distances are on the order of 1,000 feet or so, direct cable connections can be used and these may often be extended with line drivers up to three miles.

The CDC system (Table 5-8) requires a 40.8K baud line since a lot of graphic processing is done on the host computer. Typical cost for this broad bandwidth communications line and related modems over a distance of about ten miles is about \$500/month.

5.3.3 <u>Target costs for interactive terminal configurations.</u> Figure 5-6 summarizes the configurations discussed together with similar Type 4 and 5 systems. Low demand keeps production (and research) of interactive terminals low which means high prices. Conversely, the high cost of hardware plus the knowledge and lead time required for developing applications inhibits the demand for and use of these devices.

MANUFACTURER	CDC	CDC	IBM	VECTOR GENERAL	IMLAC	TEKTRONIX
MODEL	274	GPGT	2250	3D2	PDS-1	4002A
TERMINAL "TYPE"	5	5	5	5	4	4
TYPE OF CRT	REFRESHED	REFRESHED	REFRESHED	REFRESHED	REFRESHED/DVST	DVST
SCREEN SIZE (IN)	20	20	12 x 12	13 x 14	7 5 x 8 5	7 5 x 5 5
SHAPE	CIRCULAR	CIRCULAR	SQUARE	RECT	RECT	RECT
RASTER X RASTER	4,096 x 4,096	4,096 x 4,096	1,024 x 1,024	4,096 x 4,096	1,024 x 1,024	1,024 x 760
INTERACTIVE TOOLS						
A/N KEYBOARD	X	X	X	X	X	X
LIGHT PEN	X	X	X	X	X	
JOY STICK, MOUSE, ETC				X	X	X
ANALOG TABLET				X	X	X
FUNCTION KEYBOARD	X	X	X	X	X	X
MINICOMPUTER	CDC 1700	CDC SC1700	NONE	PDP-11, VARIAN 620, ETC	BUILT IN	NONE
SYSTEM COST (PURCHASE)	$165K	$100K	$150K	$85K	$25K	$15K

Figure 5-6. Representative Interactive Graphics Terminal Configurations, A Summary

The presentations in symposia and conferences over the last two years have pointed out the very clear, quantitative economic advantages and other benefits of using interactive computer graphics. These discussions have begun to have some effect. There are more ad hoc (as opposed to integrated) application, and smaller computer and peripheral equipment manufacturing companies have begun to produce acceptable devices and are consequently introducing competition into this market.

The IPAD feasibility study is helping to add impetus to this trend by showing the additional benefits to be derived from an integrated approach. (In fact showing that many IPAD objectives cannot be achieved without these interactive devices and techniques.) The wide industry and government exposure to these results is bound to be influential.

Based on Convair's experience in the cost effective use of interactive graphics and the knowledge of other industrial uses, the cost figures in Table 5-9 represent bogies or target goals for "acceptable" prices for interactive terminals. (Acceptable prices are those necessary to allow for wide-spread, cost effective use.)

It should be noted that cost effective use can be obtained within the aerospace community (Reference 18). However, to get the necessary sustained usage level, a broad base of users in various disciplines is needed. The techniques to be implemented in IPAD are appropriate and needed in many other industries (e.g., shipbuilding, farm machinery, electronics, and construction.)

TABLE 5-9

DESIRED INTERACTIVE TERMINAL PRICES

Type 5: Large vector drawing CRT with light pen, A/N keyboard, digital display processor (minicomputer) and display controller. Display area to be greater than 169 in^2 (preferably on the order of 200 in^2 to 300 in^2). Minicomputer with minimum of 16K 16 bit words, hardware arithmetic and greater than 1.5M words of mass storage (disk). A local hard copy device per one or two CRTs.

Purchase:	$40K to $50K
Lease (including maintenance):	$1500 to $2000 per month

Type 4: Medium size vector-drawing CRT with keyboard and light pen (or joystick, mouse, or analog tablet driven cross-hair cursor). A digital display processor with minimum of 8K 16 bit words, and a display controller. A local hard copy device per two to four display consoles.

Purchase:	$15K to $25K
Lease:	$500 to $1000 per month.

Type 4: Small size vector drawing CRT with positionable cross-hair cursor and keyboard. Display controller, and communication interface. Local hard copy device per two to four CRTs.

Purchase:	$2K to $5K
Lease:	$100 to $200 per month.

5.4 Scoping the Host Computing System

There is little doubt - with the background developed thus far - that the host computing system must be one of the large scale scientific computers. There are several factors that influence the actual choice of the candidates:

1. Computing power to support the terminal requirements developed in Section 5.2, e.g.:

   a. Central memory cycle time (which directly affects total job throughput).

   b. Central processor add time (which directly affects computational throughput).

c. Size of supporting central memory.

2. Peripheral supporting hardware, e.g.:

   a. Disk storage.

   b. Line printers.

   c. Magnetic tape units.

3. Operating system (software), e.g.:

   a. Batch entry subsystem.

   b. Timesharing subsystem.

Only currently marketed, large-scale, scientific computing systems are considered. All candidates have to additionally be supported by fully developed system software. As was done in Section 5.2 on interactive terminals, the host computing system hardware is sized for a single IPAD project.

5.4.1 System baseline. - As a basis, the computing system being utilized at the time of the OM Questionnaire (Section 4) was selected as a "minimum-adequate" system:

1. CDC 6400 computer with:

   a. 65,000 60-bit word central memory.

   b. 1.0 μs major (central memory) cycle.

   c. 1.1 μs binary 60-bit floating add.

2. CDC 6638 mass storage disk subsystem featuring:

   a. 1.3 M 60-bit word capacity CDC 808 disk.

   b. 24576 tracks (bands) accessed by two movable head groups.

   c. 25 - 110 ms initial access time.

   d. 50.8 ms per full rotation.

   e. 168 K word/sec transfer rate (dual channel).

3. Eight magnetic tape units.

4. Four line printers.

5. Two card readers.

6. Two card punches.

Engineering support of Space Shuttle following the Phase B study demonstrated that this system was not adequate in handling the computing environment envisioned for IPAD. At that time, NAMESIM (see Figure 2-5 and related discussion in Subsection 2.2.1) had two of three active - and fairly demanding - simulation applications going concurrently on two CDC 274 graphic consoles (the Type 5 configuration presented in Table 5-8 and depicted in Figure 5-5.) At times, the opposing graphics task was a printed circuit board packaging program making heavy demands on the mass storage subsystem. In these circumstances it was necessary to run at night and essentially dedicate the computing system to these two tasks with minimal (if any) batch background. The system response was nevertheless often poor (the "slowest" presented in Figure 2-7 of Subsection 2.2.1).

The factors that contributed to this poor system performance under heavy interactive loading will be evident and resolved in the discussions which follow.

5.4.2 Central processor performance. - Table 5-10 presents a performance comparison between the baseline computing system (the baseline system of Subsection 5.4.1) and four similar large scale scientific computers. The FORTRAN benchmark consisted of a collection of actual batch jobs in frequent use. The benchmark took just under 4 hours to complete on the baseline system beginning with an empty machine.

TABLE 5-10
PERFORMANCE COMPARISON

	Throughput (Min)	Improvement (Percent)
CDC 6400 (65,000 word CM)	239	BENCHMARK
CDC 6400 (98,000 word CM)	155	35
CDC 6600 (131,000 word CM)	99	59
IBM 370/155 (1 megabyte CM)	170	29
IBM 370/165 (1 megabyte CM)	103	57

A 35 percent improvement resulting solely from a 33-1/3 percent increase in central memory demonstrates that multiprogramming can only be achieved when a reasonable number of programs can be worked on simultaneously. The table places the IBM 370/155 as roughly equivalent to the CDC 6400, and the IBM 370/165 roughly equivalent to the CDC 6600. (Note that one megabyte is bit-equivalent to 133,000 words on a CDC 6000 Series system. However due to differences in architecture, the equivalency is not this simple.)

With the indicated improvement in performance, the CDC 6400 with the larger central memory can be considered to offer barely adequate support to IPAD. The minimum hardware requirements to support IPAD can now be established as a CDC 6400 with a minimum 100,000 words of central memory or (equivalently) an IBM 370/155 with one million bytes of central memory. Table 5-11 compares the prinicpal features that established the computing power of these two systems together with the two more powerful systems presented in Table 5-10. (The data for Table 5-11 comes from Reference 19.)

TABLE 5-11
PRINCIPAL FEATURES CONTRIBUTING TO PERFORMANCE

	CDC		IBM 370	
	6400	6600	155	165
Central Memory cycle:				
Time (μs)	1.0	1.0	0.35	0.08
Size (bits)	60	60	64	64
Binary Add:				
Time (μs)	1.1	0.3	0.12	0.08
Size (bits)	60	60	32	32
Processor Features:				
Double Precision?	Yes	Yes	Yes	Yes
Floating Point?	Yes	Yes	Yes	Yes
Multiply/Divide?	Yes	Yes	Yes	Yes
Timesharing Features:				
Base Address Relocation?	Yes	Yes	Yes	Yes
Clock?	Yes	Yes	Yes	Yes
Program Interrupt?	Yes	Yes	Yes	Yes
Memory Protection?	Yes	Yes	Yes	Yes
Dynamic Page Relocation?	No	No	No*	No*
Supervisor Mode?	Yes	Yes	Yes	Yes

What we have in Table 5-11 is - in effect - a criterion of selection for an IPAD target system, viz. that it have a minimum of 100,000 words of central memory (or the equivalent) and meet or exceed the "specifications" of Table 5-11. However the selection process is not that simple.

* This comparison - in conjunction with Table 5-10 - does not consider IBM's recently announced Virtual Storage/Virtual Machine capability.

It is not clear why the IBM systems - with their significantly faster storage cycle (read/restore) time - did not outperform the corresponding CDC systems. It is suspected that the available software for the 370/155 and 370/165 had something to do with this since these systems were in pre-delivery test when this benchmark testing was performed. Further, it is not known exactly what peripherals were utilized on each system other than they were "roughly equivalent".

What is clear however is that none of the systems tested with the benchmark is completely adequate for IPAD. IPAD (as envisioned in the conceptual design, Section 2) will operate principally as an interactive system - more specifically as an interactive graphics system. This will place harsh demands on the hardware to memory-share jobs (see Subsection 2.2.2.1) in a central memory of realistic size. (If central memory could be arbitrarily large, this problem wouldn't exist; just add more memory along with more jobs.) Memory sharing is typically accomplished via mass storage.

5.4.3 Mass storage requirements. - There are two approaches to obtaining extra-fast memory-sharing response in support of IPAD interactive users: job swapping and virtual memory.

5.4.3.1 Job swapping: Job swapping - as practiced by current CDC 6000 Series systems - is the removal of the total job from central memory and replacement of that job with another job awaiting service. Since scientific jobs are typically quite large (the mean central memory requirements for the OMs surveyed in the OM Questionnaire was 24,268 CM words), the mass storage device would need to possess unusually stringent access time and transfer rate specification to support the required swap rate. To circumvent this, CDC provides Extended Core Storage (ECS). ECS is essentially a rapid-access, high transfer rate, mass storage device. Its principal uses are (1) residence for frequently used system programs, (2) job swapping, (3) buffer between central memory and slower peripheral devices, and (4) storage of large data arrays.

Given an appropriately-sized ECS, jobs may be queued in ECS for core-to-core transfers between ECS and CM. This speeds up swapping by about two orders of magnitude. (Swapping is also termed swapin/swapout or rollin/rollout, although the latter is usually reserved for computer-operator initiated, temporary removal of a job from active processing.)

ECS and Distributed Data Path (DDP) are optional hardware on CDC computers.

5.4.3.2 Virtual memory: Virtual memory or virtual storage - as practiced (for example) by the IBM 360/67 - is a method of maintaining in central memory only those small, contiguous blocks of code required for processing in any small timeframe.

IBM describes the concept as it relates to their System/370 (Reference 20):

> In conventional systems, jobs are loaded into partitions or regions of limited space provided by the processor's main storage. In the new approach, jobs will be loaded into partitions or regions of a very large virtual storage space, up to 16 million bytes, that are primarily reserved on disk. During processing, small blocks or pages of instructions and data are transferred between disk storage and real storage (the processor storage) according to the momentary needs of each job.
>
> With virtual storage, the computer appears to have a "main" storage with an address space of up to 16 million bytes. The available address space is no longer limited by real processor storage size. Hence, the name virtual storage.

Actually, virtual memory or virtural storage is not all that "new" an approach. The BURROUGHS 5000 Series has had such a system since delivery (1962) as has IBM (Reference 21):

> IBM introduced the 360 model 67 to provide a product competitive with the GE 645 in the time-sharing market. The model 67 is a typical 360, which can run the standard 360 operating systems but which also has paging and segmentation hardware and so can support a sophisticated virtual memory operating system. The ambitious time-sharing system TSS 67 was announced in the spring of 1965...

Although paging of code segments still uses disk I/O, the program "size" being transferred between central memory and disk (paged) is ideally only that required for the "momentary needs of each job" which is typically quite small.

5.4.3.3 Mass storage capacity: The mass storage capacity required for a single IPAD project can be obtained from the OM Questionnaire results (data items are listed on Figure 4-4):

1. Storage for "active" OMs: data item 123. — 3.95 M words
2. Storage identified to the MDB: $\sum_{124}^{125}$ (data item j). — 3.91 M words
3. Storage identified to the UFs: $\sum_{132}^{138}$ (data item j). — 6.28 M words

4. "Average" storage during job execution: $\sum_{116}^{122}$ (data item j). 1.22 M words

Total: 15.36 M words

Thirty-five typical OMs obtained from the Questionnaire were assumed "active" and available on disk (item 1); ten "average" OMs were assumed to be in the execution process at any one time (item 4).

The calculation for storage identified to the Multidisciplinary Data Bank (MDB) was actually more complex than shown. If data item 124 was less than

(data item 50)*(data item 116)/100

this calculation - which represents that portion of the input which came from other programs - replaced data item 124. Similarly

(data item 60)*(data item 117)/100

replaced data item 125 if larger. This provided two somewhat independent estimates of the MDB storage requirements.

The maximum estimate of mass storage capacity was accepted as the minimum requirement.

5.4.3.4 Data transfer rate: The data transfer rate cannot be ascertained from the OM Questionnaire since there is no way to determine the actual required rate of transfer of data to-and-from mass store. An indirect approach to this problem is required.

Figure 5-7 illustrates the degradation in system performance with various hardware options (Reference 22). The data was generated using a CDC 6000 Series computer, the SCOPE 3.3 operating system and the INTERCOM 2.0 timesharing subsystem. Two groups of 5 and 15 interactive TTYs were simulated superimposed on a typical scientific batch (background) job mix. Two disk mass storage systems were investigated which differ principally in data transfer rate: The CDC 808 features 168,000 words/sec and the CDC 841 features 42,000 words/sec. ECS (Extended Core Storage) was used exclusively for INTERCOM.

Figure 5-7 demonstrates the performance dependence on disk data transfer rate as well as the improvement attained through the use of ECS. It should be noted that the system tested did not have the Distributed Data Path (DDP) hardware and loaded/unloaded ECS from/to disk via the central processor and central memory.

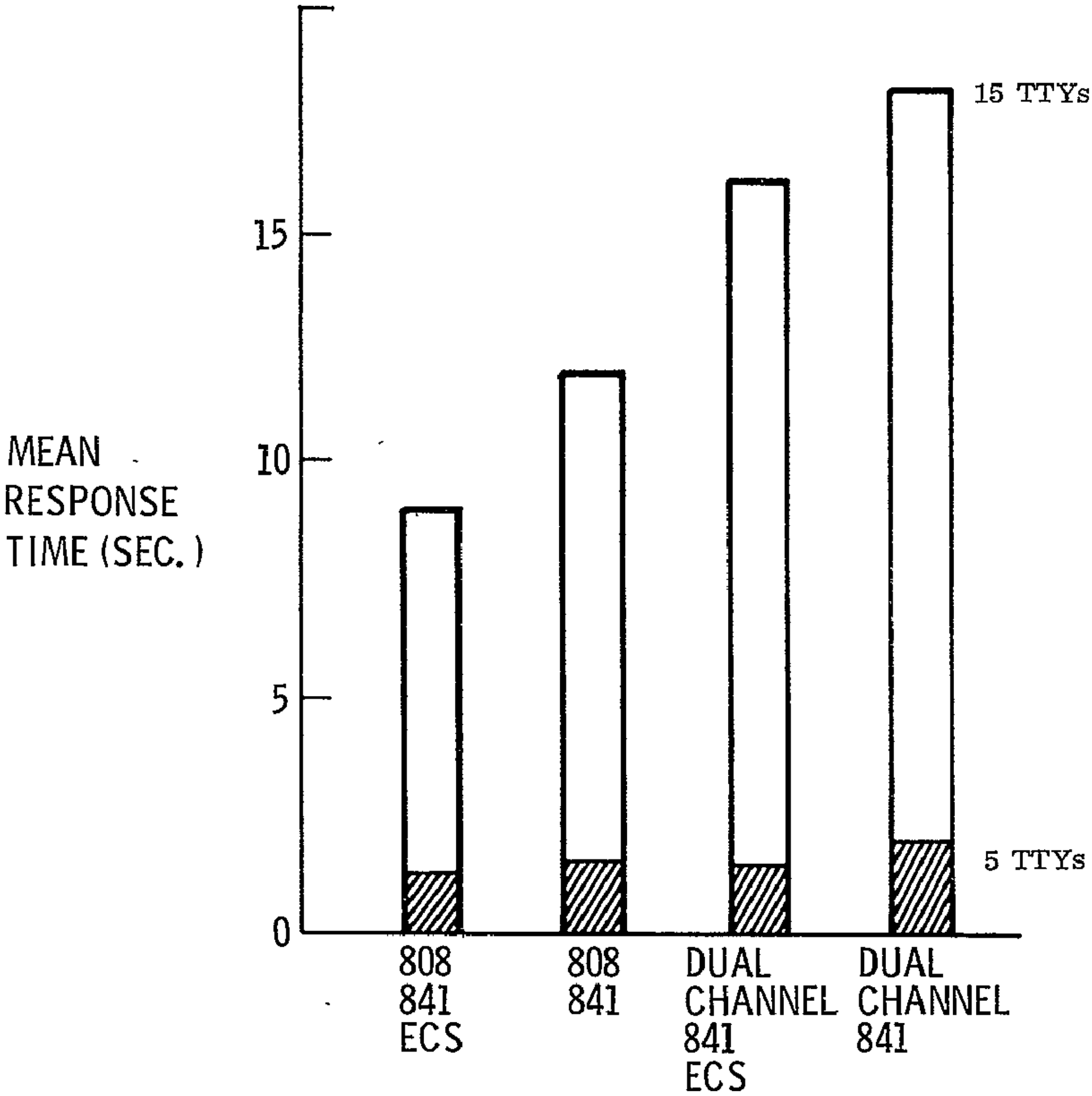


Figure 5-7. Degradation of Performance with Various Mass Storage Devices: 5 TTY vs 15 TTY INTERCOM Operation

(Systems with both ECS and DDP are expected to show a much better improvement than that shown for just ECS).

Since terminal loading for IPAD is expected to be 20 Type 4 terminals and 8 Type 5 terminals (Subsection 5.2.2), data transfer rate is a sensitive issue since the system response shown in Figure 5-7 for 15 TTYs is totally unacceptable.

Some relief will be achieved by spreading out the disk system over multiple head/ spindle systems. The 15-million word system is equivalent to one CDC 808 or four

CDC 841's. This will, in effect, quadruple the effective transfer rate of the 841 system over an 808 system (the 6638 with its controllers) by quadrupling the availability of the device.

All of the above - plus the consideration of paged virtual memory systems versus job swapping systems - are not definitive in establishing a data transfer rate "requirement". However with the prevailing trend towards a larger availability of disk units, it is envisioned that 100,000 words per second (one million characters per second) would produce a marginal system for one project.

5.4.4 Operating system software.- The operating system must provide these basic essentials to support IPAD:

1. Executive or supervisory subsystems:
   a. Batch entry.
   b. Timesharing.
2. Compilers:
   a. FORTRAN.
   b. COBOL.

IPAD - as envisioned in Section 2 - will exploit the host operating system software which must possess both a batch and timesharing subsystem (see Section 2.2.2). Further, to be effective, both a FORTRAN and COBOL compiler must be available. FORTRAN is required to support the heavy investment the scientific community has in existing OMs. COBOL similarily supports the business and management communities. To be viable, IPAD must support existing engineering management OMs (generally written in COBOL) just as it supports existing FORTRAN OMs. This is especially true in view of the accelerated schedules predicted in an IPAD environment (Section 5.1). This in effect is no problem since most candidates provide both FORTRAN and COBOL compilers.

These requirements are not meant to preclude additional compilers (e.g., ALGOL and PL/1) in support of existing OMs supported by a given computing system (e.g., BURROUGHS and IBM). The data management system supporting IPAD must be able to convert data as required so that a COBOL OM may provide data to a FORTRAN (or ALGOL) OM as might be required.

5.4.5 Summary. - Figure 5-8 summarizes the estimated requirements placed on the host computer. In addition to those requirements discussed in the preceeding subsections, a minimal complement of magnetic tape units, card reader/punches, line printers, and recorders has been presumed. It is noted that in an IPAD environment, most data is envisioned to originate and remain within the system on mass storage devices. The mass storage capacity shown in Figure 5-8 (i.e., 150M single-precision words) has been arbitrarily selected one order of magnitude larger than that derived in Section 5.4.3.3 to account for the more extensive storage needs of a full-fledged design activity including extensive data libraries, design data for various subsystems, and so on. Data viewing, job submittal, etc. is to be accomplished via interactive devices so that the requirements for conventional job entry and I/O are minimal.

It must be recognized that the requirements on central memory is in addition to the requirements for the IPAD system support software. Typically, efficient host system software supporting the IPAD user is written as re-entrant code, supporting many users and residing in central memory. The data management system support to IPAD may significantly increase central memory residency. It is suspected that the 100,000 word requirement to support IPAD OMs may expand to the order of 130,000 for total system support.

MINIMUM REQUIREMENTS

- MAIN FRAME HARDWARE

MAJOR MEMORY CYCLE	$\leqslant$ 1 0μS
TYPICAL BINARY FLOATING ADD	$\leqslant$ 1.5μS
CENTRAL MEMORY SIZE*	$\geqslant$ 100,000 SINGLE PRECISION "WORDS"
JOB ROLLIN/ROLLOUT OR SWAPIN/SWAPOUT	"PAGING" OR HIGH-SPEED TRANSFER TO EXTERNAL (LOW-SPEED) CORE

- PERIPHERAL HARDWARE

MASS STORAGE CAPACITY	> 150 M SINGLE PRECISION "WORDS"
MASS STORAGE TRANSFER RATE	> 1M CHARACTERS PER SEC.
MAGNETIC TAPE UNITS	>3
CARD READER/PUNCH	1
HIGH-SPEED PRINTERS	1
MICROFILM RECORDER	1 (CAN BE REMOTE)
TERMINALS (WITH HARDCOPIERS)	RECOMMENDED IPAD MIX (SECTION 5.2)
PAPER TAPE READER/PUNCH	IF REQUIRED

CANDIDATES (ALL ARE LARGE-SCALE SCIENTIFIC COMPUTERS)

IBM	CDC	UNIVAC	HONEYWELL	BURROUGHS
370/145	CYBER 70/72 (6200)	1108	6000/6030/6040	B6500
370/155,158	CYBER 70/73 (6400)	1110	6000/6050/6060	B6700
370/165,168	CYBER 70/73-2 (6500)		6000/6070/6080	B7700
	CYBER 70/74 (6600)			
	CYBER 70/74-2 (6700)			

*IPAD WILL INCREASE CENTRAL MEMORY RESIDENCY

Figure 5-8. Estimated Minimum Host Computer Requirements: Single IPAD Project

The hardware candidate computing systems (bottom of Figure 5-8) were obtained from Reference 19 by liberally applying the "requirements" of Table 5-11 and Subsection 5. 4. 4 except that the binary floating add time was relaxed to 1. 5 μs as noted on Figure 5-8; e. g., to be a candidate, the computing systems central memory had to be expandable to an excess of 100,000 "words" or the equivalent 1.0 megabytes. However the "requirement" for paging or high-speed transfer of programs to-and-from mass storage was not enforced except to limit the number of candidates to the best available from a manufacturer.

For example, the IBM candidates have been limited to those systems supported by their recently announced Virtual Machine facility (Reference 23):

> The IBM Virtual Machine Facility/370 (VM/370) is a System Control Program (SCP) that has been designed specifically for the IBM System/370. VM/370 manages the IBM System/370 in such a way that multiple remote terminal users appear to have a dedicated computing system at their disposal. Within this "virtual machine" the user may run the operating system of his choice subject to the [following] restrictions.
>
> VM/370 and its associated component, the Conversational Monitor System, is based on the CP-67/CMS system and is designed especially for the IBM System/370. Dynamic address translation provides the same facilities as did dynamic address translation (the DAT box) on the System/360 Model 67, but differs in design detail and implementation. Consequently, CP-67/CMS will not run on a System/370 and VM/370 will not run on a System/360.
>
> The Conversational Monitor System of VM/370 functionally extends the Cambridge Monitor System of CP-67/CMS.
>
> The Conversational Monitor System includes a batch job facility which accepts input streams in CMS command formats.

With VM/370, each IPAD user could elect to operate under the operating system of his choice (e.g., CMS) and the system would automatically provide the time-sharing and memory-sharing required to meet the demands of other users, (Reference 23).

The resulting candidates (Figure 5-8) are generally the top of the line systems of each manufacturer which are in current production. As can be seen, a reasonable number of candidates are available that can - at least minimally - support IPAD.

5.5 Selection of the Target Computing Systems for IPAD

Having established the available candidates (Figure 5-8), it is possible now to select several as "target systems" to support IPAD. This selection is required to provide a broad basis of systems to test the efficacy of a proposed IPAD system design.

The issue of what three candidate computing system should be selected as the target systems is easily resolved. Clearly IBM - in its position of dominance in the computer market - must be one of the choices. CDC - because of convenience for conducting this study and being represented at the customer's facility (NASA LRC) - is also a logical choice. The obvious other choice is UNIVAC which together with CDC and IBM account for most of the aerospace design application, both within government and private industry.

Figure 5-9 is an update of Figure 2-8 of Subsection 2.2.2.1, adding IBM's recently announced Virtual Machine Facility (VM/370). With VM/370, all three systems offer equivalent capabilities. But there are different software capabilities on each system, and differing ease of use of the software.

SOFTWARE FEATURE	AVAILABLE UNDER CDC SCOPE? (VERSION?)	AVAILABLE UNDER IBM VM/370?	AVAILABLE UNDER UNIVAC EXEC8?
• RANDOM ACCESS FILES (FORTRAN)	YES(3 1 2)	YES	YES
• INDEX SEQUENTIAL FILES (FORTRAN)	YES(3 3)	YES	YES***
• PERMANENT FILES	YES(3 1 6)	YES	YES
• UPDATE UTILITY	YES(3 2)	YES	YES
• INTERACTIVE COMMUNICATIONS	YES	YES	YES
• INTERCOM 2.0	YES (3 3*)	–	–
• INTERCOM 3 0	YES(3 3)	–	–
• INTERCOM 4 1	YES (3 4)	–	–
• TIME SHARING?	YES	YES	YES
• MEMORY SHARING?	YES	YES	YES
• INTERACTIVE GRAPHICS SYSTEM	YES**	YES**	YES**
• IGS VERSION 1	YES (3 1 2)	–	–
• IGS VERSION 2	YES(3 3)	–	–
• TIME SHARING?	YES	YES	YES
• MEMORY SHARING?	YES	YES	YES

*We Retrofit INTERCOM 2 0 Into SCOPE 3 2
**Recognizing That Graphics Language & Architectures Differ.
***But Difficult With Fortran (Best Use Assembly Language)

Figure 5-9. Summary of Currently Available Software Features Deemed Important to IPAD

Under VM/370 (as discussed in Subsection 5.4.5), the user may run the operating system of his choice with few restrictions. The UNIVAC EXEC 8 operating system supports both the 1108 and 1110 and offers the same capabilities to each. However, with CDC several distinct "current" operating systems are available for consideration:

1. SCOPE 3.4, CDC's standard CYBER 70 operating system.
2. KNONOS 2.0, CDC's special timesharing operating system.
3. LRC SCOPE, NASA Langley's special operating system for their unique CDC computing complex.

These are described in the subsections which follow.

5.5.1 SCOPE 3.4. - The SCOPE 3.4 operating system services the CYBER 70 line (excluding the Model 76, the CDC 7600). SCOPE (Reference 24):

1. Provides simultaneous control of the job input stream, output stream, and execution of jobs.
2. Controls a wide variety of assemblers, compilers and utility routines.
3. Supports a large range of peripheral devices.
4. Provides automatic job scheduling.
5. Provides sophisticated file management.
6. Keeps a chronological record on the system log of all jobs run and systems activity.
7. Offers a variety of random and sequential access methods for data retrieval.
8. Provides automatic tape scheduling; writes and verifies tape labels.
9. Provides checkpoint and restart procedures.
10. Provides batch, remote batch and interactive job processing.
11. Supports permanent files.

A schematic example of CYBER 70 communications and control paths appears in Figure 5-10. Components of the SCOPE operating system are distributed among the central memory, the peripheral memories, ECS and the system disk unit. One peripheral processor, containing the monitor, is in permanent supreme control of the system. A second peripheral processor, under control of monitor, is permanently assigned to the console keyboard and displays; the remaining processors are available for assignment as required.

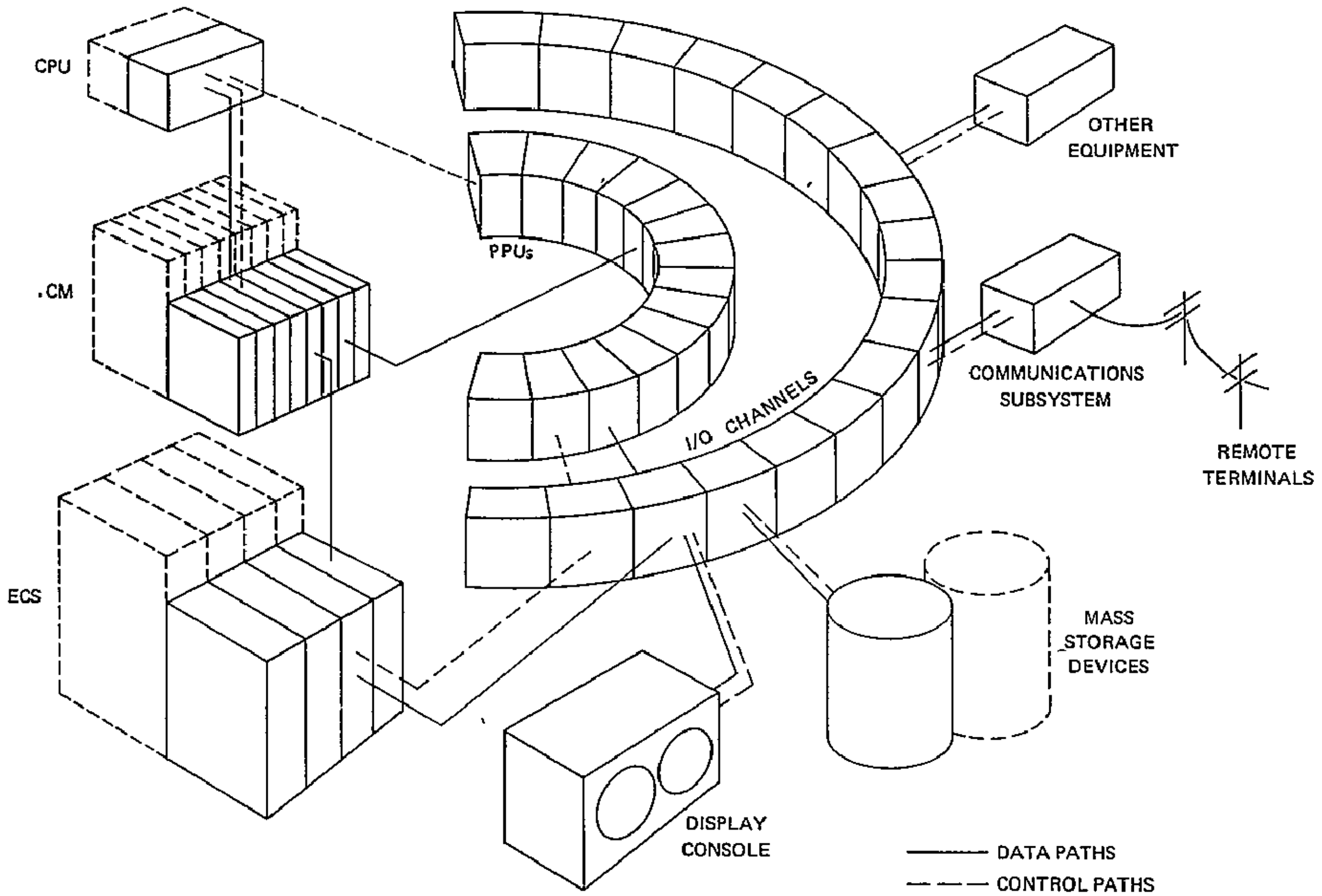


Figure 5-10. CYBER 70 Data and Control Flow

The monitor distributes work among the peripheral processors, communicating with them through an area in central memory reserved for this purpose. The peripheral processors contain routines which continually examine this communication area for requests. When a request appears, a peripheral processor carries it out. When the task is complete, the peripheral processor informs the monitor and returns to the idling routine until another request is made.

This extensive use of the independent peripheral processors achieves a minimal use of central memory and the central processor, substantially reducing overhead. High overall computing speed produced by the multiple operating modes of all segments of the computer is a significant feature of the CYBER 70 series. SCOPE employs the multiprocessing capability to the best advantage, controlling simultaneous job execution, directing input/output, and ensuring an efficient system throughout.

There is currently a wide base of SCOPE installations (53 of the 66 CDC installations reporting in Reference 25 use SCOPE; 39 of the 53 using the latest version).

5.5.2 KRONOS 2.0. - The KRONOS timesharing operating system services the CYBER 70 line (excluding the Model 76). KRONOS (Reference 26):

"....was developed by Control Data to provide interactive job processing capabilities in addition to the local and remote batch processing capabilities of the CYBER 70 computer system. This capability makes the speed and computing power of the CYBER 70 available to more users. The remote time-sharing terminal also takes advantage of the cost/performance ratio of Control Data's CYBER 70 computers."

The KRONOS Time-sharing System concurrently provides four types of job processing:

1. Local batch processing: Jobs can be entered and executed at the central site using all of the peripheral equipment attached to the central computer.

2. Remote batch processing: Jobs can be submitted from a remotely located 200 User Terminal.

3. Deferred batch processing: Jobs can be entered into the batch queue from an interactive terminal with the output routed to the appropriate peripheral equipment.

4. Interactive terminal processing: Jobs can be entered from a teletypewriter or teletypewriter-like terminal. The terminal responds as if it were the console of a small computer to which the user has sole access. However, the computer allocates only a small portion of its time to process the requests from each terminal. This process, known as time-slicing, is a significant factor in the design of KRONOS.

Figure 5-11 shows the general KRONOS system configuration.

KRONOS is designed to concurrently service up to:

1. 512 active time-sharing terminals, or

2. 16 remote batch processing terminals and a number of time-sharing terminals (the exact number depends on the work load and the system resources).

KRONOS - being specifically designed as an timesharing system - might first appear as a logical choice for IPAD. But several factors make SCOPE 3.4 a better choice:

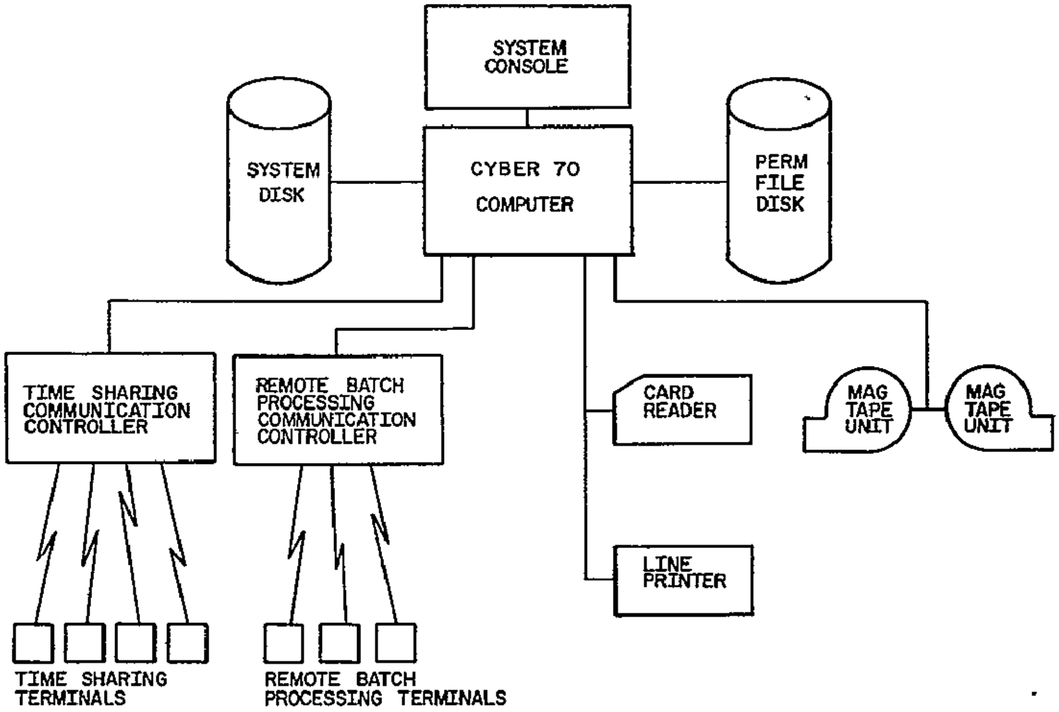


Figure 5-11. KRONOS System Configuration

1. There are currently fewer users of KRONOS than SCOPE by a factor of 10 (53 SCOPE vs. 6 KRONOS, see Reference 25).
2. SCOPE contains more advanced operating system features, principally in the data management area (e.g., advanced CDC file access techniques like SDA and SAK).
3. KRONOS does not support Type 5 terminals whereas SCOPE does via CDC's Interactive Graphics System (IGS).

5.5.3 LRC SCOPE. - LRC SCOPE is a "heavily modified" version of SCOPE 3.0 (Reference 25). (It is of interest to note that of the 66 domestic installations reporting in Reference 25, six in addition to LRC have special systems.) Special features have been added to support Langley's computing complex consisting of three CDC 6600's and two CDC 6400's (Reference 27):

> The contract for the Langley Research Center Computer Complex was awarded to the Control Data Corporation in June 1966. This contract called for the incremental delivery of several large scale computer systems, a

shared core memory* for inter-system communication and a variety of standard and special purpose peripheral equipment organized into a shared peripheral pool. The contract also called for the delivery of operating systems and supporting systems software to mold this collection of equipment into a unified, multi-computer complex, capable of supporting the LRC's computational requirements in the areas of scientific batch processing, remote computing, on-line research data reduction, and real-time flight simulation.

The LRC system does not contain an interactive timesharing subsystem nor an interactive graphics capability except that six CRT consoles provide a display of certain items and there are plans for interactive graphics (op.cit.):

> In order to maintain sufficient communication and control during the progress of a simulation, a high performance CRT (cathode ray tube), with keyboard, function switches, etc. and a central hardcopy recorder will be available to the test conductor. The CRT will be used to provide a dynamic display of selectable key parameters in graphic and/or tabular form for monitoring purposes. It will also provide real-time control features which will allow the test conductor to START, STOP, or HOLD the simulation, perform detailed analysis of historical information collected during the course of the simulation and resume or reinitialize the simulation at various selectable times.
>
> Development activity is currently underway in ACD which will provide a more general purpose use of CRT's. Out objectives are to provide, thru software development, the necessary communication, control, and graphic tools to support interactive problem solving and analysis via the CRT.

5.5.4 Target systems. - In consideration of the above, the three target system selected as a basis for the subsequent design feasibility study were:

1. IBM 370/145, 155, 158, 165, 168 with VM/370.
2. CDC 6000 Series with SCOPE 3.4.
3. UNIVAC 1108, 1110 with EXEC 8.

---

* Further developments led to a decision to integrate the complex using a "front-end" computer rather than a shared core. The complex contains no ECS. The functional description is still applicable.

5.6 The Role of Computer Networks

There is little question that IPAD will place heavy - perhaps excessive - demands on present day computing systems. This follows directly from two trends:

1. There will be a reduction in design time for each design phase (Section 5.1).
2. There will be an increase in computing in each design phase as tasks formerly done by hand are automated.

This will result in a sharp increase in computing per unit time.

What is the role of computer networks (e.g., the network being developed by ARPA)? It is clear from the discussion on communications (Subsection 4.3.1.4) that teleprocessing will be expensive if at all practical. The IPAD user is not a conventional timesharing user dealing with a minimal amount of data, but rather more like an interactive graphics user. IPAD will require communication rates from 2000 to 5000 baud for Type 4 terminals (Subsection 5.3.1) and from 4800 to 9600 baud for Type 5 terminals (Subsection 5.3.2). (Up to 50 kilobaud may be required if the Type 5 terminal is used without a minicomputer.) Further the terminal loading consideration (Subsection 5.2.2) require that up to 28 interactive "ports" be available for a single project. Such demands are unlikely to be met until an effective microwave and/or satellite network is in operation. However this is soon to be a reality (e.g., Western Union's communications satellite plans). In the not too distant future, an IPAD system could be based at the nodes of a vast computer network, each node being a complex of sophisticated large-scale scientific computers.

What is at issue is the development plan for IPAD. There are two distinct ways to develop an IPAD system:

1. Develop IPAD as code transferable on all of the major (i.e., "target") computing systems.
2. Develop IPAD for installation only at the various nodes of a large computer network.

The advantage in development costs for a network are obvious if the system can be built for a single computer or complex of computers of essentially one architecture. However the problems associated with the utilization of IPAD in such a system are diverse.

It is unlikely that industry would accept such an environment voluntarily. There are the problems associated with proprietary code and proprietary data. There is the inevitable "standardization" and associated loss of control over "their" computing complex. Use of such networks could be written into contracts but this is unlikely to foster the desired commitment by industry necessary for widespread adoption and use of IPAD.

There is also the very real problem of classified data. DOD regulations are very uncompromising in this area. For example, all classified computing is done within a restricted area with posted security guards. Teleprocessing is acceptable providing it is within that restricted area. All communication equipment going outside that area must be <u>physically</u> disconnected. All personnel within that area must have the appropriate (e.g., SECRET) clearance and be directly associated with that project. If the computer is also to be used for unclassified project work, all classified data must be physically removed (tapes, disks, cards, listings) from the computer area. All memory devices remaining (e.g., fixed disks, drums, core) must be overwritten several times. Line printer ribbons must be removed and destroyed. Compliance must be assured before any communication equipment is reconnected. In short, if classified computing is to be done, it is only practical on a dedicated system.

Who then will use IPAD on a computer network? Smaller companies who cannot afford a maxicomputer are likely candidates. Larger companies who are using leased code or have computing requirements which cannot adequately be met by their own computers are also candidates. Companies under contract to use these facilities are candidates. If these contract requirements read only that the data repose within the computer network, it is optional that the network be used for computing and it is likely that many companies would elect only to use the network as a data collector.

It is further unlikely that the computers of the various nodes of the network could be standardized to those of a given manufacturer or even of a given architecture. So the issue of development may not be an issue at all.

The recommendation is to develop IPAD for the three target configurations as a minimum.

5.7 What About the Supercomputers?

What about the future of hardware that will revolutionalize computing? The laser memory that promises to make practical ultra-fast gigabit memories. Or the soon-to-be-reality of inexpensive, mass produced nanosecond memories based on LSI technology. When practical, these will find their way into the computers of their day just as past "breakthroughs" are taken for granted in contemporary computing systems. In this way they will benefit IPAD.

What about the future computing system? What role do parallel processors (e.g., the University of Illinois' ILLIAC IV or Goodyear's STARAN) and the pipeline processors (e.g., CDC's STAR-100 or TI's Advanced Scientific Computer) play in the IPAD development? Each of these new developments - by their very nature - employ revolutionary techniques which obsolete existing software development. Most are released with primitive system software which is slow in development and generally lags behind the state of the art. Most - due to their radically different architecture - are difficult to exploit, e.g.,:

1. Computational speed depends heavily on program code.

2. Programmers usually need to understand machine architecture to achieve the promised benefits.*

3. Existing OMs would derive little benefits unless reprogrammed* to adapt to this revolutionary architecture.

4. Reprogramming* to produce optimum code is probably only justifiable on frequently used (e.g., multi-user) OMs.

It is undoubtedly not justifiable to saddle IPAD's development with the requirement to accommodate the supercomputers. In particular there is often not much of a host operating system to "exploit."

Does this means that IPAD will not take advantage of these giants when they become available? Not at all. The recommendation is:

1. Don't explicitly provide for the Supercomputers in IPAD's design approach. Let "upward compatibility" of the Supercomputer's system software eventually provide the framework for IPAD.

2. Until then, "front-end" the Supercomputer with a more sophisticated maxi that can delegate candidate tasks. Design IPAD to reside on the maxi.

---

* At least until sophisticated, machine-adaptive, optimal compilers are produced which can suitably reorganize the code to suit the host hardware.

6 TRANSFERABILITY, SOME PRELIMINARY* CONSIDERATIONS

Any consideration of transferability of IPAD system software must distinguish between the concepts of transferability and tranparency.

Transferability was defined for the purposes of the IPAD User Survey (Section 3) as "the degree to which IPAD software can be transferred from one hardware and/or software installation to another without requiring modification."

Transparency may be defined for IPAD purposes as "the degree to which a user utilizes the same command language when moving from one IPAD implementation to another." Attaining this objective was not specifically addressed in the IPAD User Survey, but "Standardization" (see Section 3.2) contained information which overlaps the concept of transparency. The degree of familiarization required with the new computing system as an IPAD user moves from one computing system (machine) to another diminishes as the degree of transparency increases. The tutorial programming and user manuals (documentation) remain essentially the same for all systems if transparency is sufficiently attained.

This section treats "transferability" from both aspects.

6.1 Transferability

Any discussion of transferability must consider the trade between how much of the software system must be transferable and the level of efficiency required. If a low efficiency is acceptable, total transferability is achievable by simulation (or emulation) of the original machine on the host computer. If a very high level of efficiency is required, maximum use of efficient facilities within the host computer must be exercised; and, therefore, less of the system is truly transferable.

Widespread use of FORTRAN as a universal language has created a competitive environment among computer manufacturers to best accommodate the language. This has nourished a degree of transferability of programs written in this language which is not present in the use of some other languages. It is reasonable to expect that widespread use of software systems such as IPAD will promote interest among hardware manufacturers to accommodate the system in an efficient manner. The transferability of the system would increase if this phenomenom occurs.

* This is additionally discussed in more detail in Part II, Section 9, of Volume V.

6.1.1 Factors affecting transferability. - The trade between degree of transferability and level of efficiency is dependent upon a number of factors. In the case of languages, a compiler must exist on the host computer before even considering the efficiency of the language and its transferability. Also among these factors are the type of software, the degree of program modularity, the programming and coding techniques employed, and the desired level of efficiency. These factors are discussed in the following subsections.

6.1.1.1 Language selection: The choice of languages for maximum transferability must be a language combining the least number of machine-dependent features with the greatest number of features useful to IPAD. FORTRAN and COBOL are two good languages if properly used, and PL/1 (a powerful, declaration-based language) is perhaps even better. A compiler must be made available, however, to employ PL/1 since these are principally available only on IBM systems.

Assembly languages are completely machine dependent, highly efficient if used properly, but essentially non-transferable.

There are specialized languages such as SNOBOL, a character manipulation language (Reference 28), which are highly transferable because they depend on interpretive execution. If interpretive execution is tolerable, even assembly languages can be transferred; but the costs are prohibitive in most cases.

6.1.1.2 Software's function: The intended use of the software being implemented determines to a large extent the desirability of machine or implementation dependency. An ordinary mathematical application program does not become overly involved in the dependency factors. Software which must interface closely with the machine or its operating system - such as the IPAD EXEC - must have a relatively high dependence no matter how well it is programmed.

6.1.1.3 Program modularity: Modularity tends to improve overall transferability because machine dependencies can be kept concentrated in a few relatively independent modules instead of being scattered throughout an entire program. These few modules can be recoded completely, if necessary, when the program is transferred. Modularity also simplifies all types of program modification, including transference, because the different modules can be modified, debugged and tested individually. Unfortunately, efficiency conflicts with modularity just as it conflicts with most other things; a monolithic program can be made more efficient than a modular one.

6.1.1.4 Programming skills: Skillful and thoughtful programming can greatly ease transferability, especially if the program is coded initially with the idea in mind that it is to be transferred eventually. Thorough and meaningful comments are particularly valuable, especially those that point out areas of machine dependence and explain who a machine-dependent approach was used. These comments can be used as a guide in reprogramming software packages; they are an IPAD programming requirement.

6.1.1.5 Exploiting machine specifics: Efficiency is highest when all the advantages of a particular machine are being employed. Such a program may have to be coded in assembly language which makes it practically non-transferable. No matter which language, exploitation of many of the advantages of any one computing system tends to make the program machine-dependent.

A quest for high efficiency can also impact transferability in another, more subtle way because it involves program structure rather than just coding methods. Consider, for example, a program that involves extensive looping. On a CDC 6400 (as on most other machines) it would be most efficient to design it to use as few separate loops as possible, to minimize the amount of loop-closing overhead. On a CDC 6600 - even though it is upward compatible with the 6400 - it would be much more efficient to use more loops and keep each one small enough to fit into the machine's 8-word, high-speed, discrete component instruction stack (16 to 31 instructions). On a UNIVAC 1108 it would again be most efficient to minimize the number of loops, unless array searches or moves are involved; these should be broken out as separate loops to take advantage of the machine's repeat-mode search and block move instructions.

What makes this sort of thing a special case of "transferability" is that, in one sense, it doesn't impair transferability at all; even when a program has been specifically structured for efficiency on one machine it can still be transferred to another machine and run there correctly. It just won't run as efficiently as it could if it were restructured to take advantage of the new machine's characteristics or, perhaps, never structured for exploitation in the first place.

6.1.2 Conclusions. - Many factors have been briefly treated which influence - to some degree - the transferability of IPAD system code. No specific conclusion can be reached at this point since the trade between degree of transferability and efficiency cannot be answered without considering, for each module of code:

1. Language to be used.
2. Function to be served.
3. Modularity to be achieved.
4. Programmer skills envisioned (considering language and function).
5. Host system exploitation to be sought.

These considerations must await identification of the specific modules.

6.2 Transparency

It is important to distinguish between two degrees of transparency in this discussion:

1. Transparency limited to IPAD itself with the commands unique only to IPAD being similar on all machines.
2. Complete transparency with all commands, both IPAD and non-IPAD, the same on all machines.

Inexperience of the user is a factor to be considered, but this study of transparency assumes user familiarization with at least one system and touches upon aspects which are concerned only with moving from one machine to another.

6.2.1 Transparency limited to IPAD. - Transparency limited to IPAD implies that user commands - such as those that control one of IPAD's General Purpose Utilities - would be the same on all machines; non-IPAD commands would be provided by the respective standard operating systems in each machine. Most time-sharing systems use different command structures in their own different subsystems such as the Text Editor, interactive debugger, BASIC compiler, etc., and a set of commands structured for IPAD will not be likely to overcomplicate the system from a user standpoint. Table 6-1 illustrates the forms of a number of the commands likely to be needed by IPAD users on each of the three target computing systems:

1. The CDC Cyber 70 or 6000 Series with SCOPE 3.4 and INTERCOM 4.1.
2. The UNIVAC 1106, 1108 or 1110 with EXEC 8.
3. The IBM 370 with VM 370 and the Conversational Monitoring System (CMS).*

As noted in the table, some commands are the same or quite similar on all three systems, other commands are quite dissimilar - even to the extent of accomplishing somewhat different functions. This puts the burden on the user to accommodate these dissimilarities - particularly the more subtle ones - in moving from one system to another.

The set of tutorial aids and user manuals required in the "IPAD only" approach would have some effect in reducing the effort in changing machines, and completeness

---

* Documentation for VM/370 and the Conversational Monitoring System (CMS) on the IBM 370/145, 158 and 168 has not yet been officially released, but the command language is said to be very similar to that for the 360/67 with CP-67 and the Cambridge Monitoring System (also termed CMS).

TABLE 6-1

COMMAND LANGUAGE COMPARISONS

SESSION INITIATION	CDC	LOGIN
	UNIVAC	@ RUN
	IBM	LOGIN (CP-67 COMMAND)
SESSION TERMINATION	CDC	LOGOUT
	UNIVAC	@ FIN
	IBM	LOGOUT
ALLOCATE OR DE-ALLOCATE MEMORY	CDC	EFL, REDUCE
	UNIVAC	(NOT NEEDED-AUTOMATIC)
	IBM	(NOT NEEDED-AUTOMATIC)
ALLOCATE PERIPHERAL EQUIPMENT	CDC	REQUEST (ALLOCATES DISK ONLY-TAPES NOT AVAILABLE TO TIMESHARING JOBS)
	UNIVAC	@ ASG
	IBM	LINK (LINKS DEVICE TO JOB), LOGIN (CMS COMMAND - "LOGS IN" DISK SPACE FOR CMS FILES. NOT THE SAME AS CP-67 LOGIN, ABOVE.), FORMAT (FORMATS DISK SPACE), FILEDEF
DE-ALLOCATE PERIPHERAL EQUIPMENT	CDC	RETURN, UNLOAD
	UNIVAC	@ FREE
	IBM	DETACH
SAVE LOCAL FILE AS PERMANENT FILE	CDC	CATALOG, STORE (SHORT FORM)
	UNIVAC	@ ASG (SUBFIELD ON @ ASG COMMAND INDICATES FILE IS PERMANENT)
	IBM	N/A - FILE IS TEMPORARY IF ASSIGNED TO TEMPORARY DISK SPACE, OTHERWISE IT'S PERMANENT
CONNECT PERMANENT FILE AS LOCAL FILE	CDC	ATTACH, FETCH (SHORT FORM)
	UNIVAC	@ ASG
	IBM	N/A - SEE ABOVE
DELETE PERMANENT FILE	CDC	PURGE, DISCARD (SHORT FORM)
	UNIVAC	@ DELETE
	IBM	ERASE
FILE POSITIONING	CDC	REWIND SKIPF, SKIPB (SKIP RECORDS, FORWARD OR BACKWARD) BKSP (BACKSPACE FILES)
	UNIVAC	@ REWIND, @ MOVE
	IBM	TAPEIO

TABLE 6-1
COMMAND LANGUAGE COMPARISONS, Concluded

FILE COPYING, ETC.	CDC	COPYBR (BINARY RECORDS) COPYCR (CODED RECORDS) COPYBF (BINARY FILES) COPYCF (CODED FILES)
	UNIVAC	@ COPY @ COPIN, @ COPOUT (WITH REFORMATTING FOR PROGRAM FILES)
	IBM	COMBINE (CONCATENATE FILES) SPLIT (SPLIT UP A FILE) PRINTF (COPY FILE TO TERMINAL) COMPARE, UPDATE, SORT
CALL TEXT EDITOR	CDC	EDITOR
	UNIVAC	@ ED
	IBM	EDIT
CALL ASSEMBLER OR COMPILER	CDC	COMPASS (ASSEMBLER) RUN, FTN (FORTRAN) BASIC, ALGOL, COBOL
	UNIVAC	@ ASM (ASSEMBLER), @ FOR (FORTRAN) @ COB (COBOL), @ BAS (BASIC)
	IBM	ASSEMBLE, FORTRAN, SNOBOL, PLI(PL/1) BRUIN (BROWN UNIV. INTERPRETER)
LOAD AND EXECUTE A PROGRAM	CDC	XEQ
	UNIVAC	@ XQT
	IBM	LOAD, START, $ (EQUIVALENT TO LOAD + START), IPL (SIMULATES BOOTSTRAP LOAD)
ACTIVATE COMMANDS FILE	CDC	NOT AVAILABLE
	UNIVAC	@ ADD
	IBM	EXEC
SUBMIT BATCH JOB FROM TERMINAL	CDC	BATCH
	UNIVAC	@ START
	IBM	USE IPL TO LOAD BATCH SYSTEM
SEND MESSAGE TO MACHINE OPERATOR	CDC	MESSAGE, M (SHORT FORM)
	UNIVAC	@ MSG
	IBM	MSG

of the set would be relatively simple to achieve (albeit at higher costs) within the limits of the requirements.

6.2.2 Complete transparency. - Complete transparency, which may appear at first to be advantageous, has some significant practical disadvantages. These are discussed separately in the subsections which follow.

6.2.2.1 Providing a universal job control language: Either an existing operating system control language must be adopted or a new language must be developed specifically for IPAD. Use of the existing language would be unnatural to all but the machine for which it was developed. It is concluded that a new operating system control language must be developed.

The American National Standards Institute (ANSI) formed an ad hoc committee (X.3.4.2F) on Operating System Control Language (OSCL) for this express purpose in October of 1968. This group met from February 1969 through July 1971 investigating the proposed "industry-standard input language for top-level control of the computing process" (Reference 29). They concluded "that there is a need for a standard, there is clear need for further work on OSCL [and] a formal study committee [should] be formed" (ibid). Their results were published in July of 1971 (Reference 29, a summary of which is presented in Appendix C). Although ANSI's SPARC standing committee of ANSC X3 authorized a study committee (X3/SPARC/OSCL) to investigate potential standardization of OSCL, this committee has remained dormant until quite recently (circa November 1972) when it was activated under a new chairman. (See Appendix D for the realtionship of SPARC, the Standards Planning and Requirements Committee, to ANSI's ANSC X3.) In continuing X3/SPARC/OSCL following publication of their study report (Reference 29), ANSC X3 evidently recognized "the essential study [is] required to make order out of this enormously complex area" (Reference 30).

It is reasonable to assume that X3/SPARC/OSCL - upon further studying the problem - will recommend standardization. Such a universal Job Control Language (JCL) which would result from standardization could then be imposed on the manufacturers for orderly implementation. Although this would nearly circumvent the transferability issue since all operating system languages would then be "standard", this could take many years. It is doubtful whether this could be accomplished in time to support IPAD. The resulting universal JCL however could (if available) be adopted for IPAD; this leads to further difficulty.

6.2.2.2 Implementing a universal JCL for IPAD: A major programming task would have to be undertaken to give the IPAD EXEC all the power and flexibility to:

1. Intercept user commands.

2. Analyze commands syntactically and semantically.

3. Generate equivalent commands in the host system's language.

4. Pass the commands to the host system to be executed.

The task of integrating the required code into the existing host operating system would be difficult to program and check out. (However, the conceptually simpler approach of writing an entirely new operating system to implement the JCL on each machine is entirely unrealistic due to the magnitude of the rework involved.) Presuming that a universal JCL were implemented, the system overhead would be significantly increased due to the increased size of the EXEC (instructions, syntax and semantic tables, etc.). The core storage available to all users would be correspondingly reduced.

Reduction of power, flexibility, adaptability, and open-endedness is also likely to occur. It would probably be necessary as an interim measure to exclude some features of practically every operating system on the grounds that they are incompatible with the other systems. Whether implementing IPAD on a new machine or phasing in a new operating system release, useful but unique new features could not be used without degrading transparency - unless they were somehow incorporated into all systems.

Further, complete transparency is unattainable - unless implemented via an ANSI standard by the manufacturer - due to the inherent, existing incompatibilities between systems and between machines. The differences in memory allocation methods, memory structures, loading features, word sizes, and other machine and system characteristics are too many and too varied to permit development of a truly identical operating system command language for all machines and all systems. From any such unilateral attempt, there would be remaining small differences, and these remaining small differences would be likely to cause more confusion than would significantly different languages since they would tend to go unnoticed in the overall similiarity causing subsequent problems.

6.2.3 Conclusions. - It is doubtful that ease of moving from one system to another (complete transparency) is really a valid or worthwhile objective. Personnel experienced on their own systems would probably view (at least initially) modification made in the name of complete transparency as a nuisance. Inexperienced users would undoubtedly experience difficulty when going from IPAD tasks to non-IPAD applications. Note however that implementing a universal JCL as IPAD's command language initially places all users in an inexperienced-user category; in this event non-transparency may actually be preferred.

A restricted level of transparency (that limited to IPAD itself) has none of the disadvantages of complete transparency. Since IPAD is totally undeveloped at the offset, there will be no loss of power, efficiency, or flexibility through use of a common language and a saving in design effort will result.

6.3 Recommendations

The recommended approach is to develop IPAD's control language to control IPAD's own utilities and functions the same on all implementations as nearly as is practicable. For all other functions, the host system's native operating system's command language should be used. If and when ANSI develops and implements an Operation System Control Language (OSCL) Specification and achieves compliance, the issue of transparency will vanish since all computing systems will be designed around this specification (or nearly so).

The issue of transferability is not so neatly dispatched. Many factors affect transferability and each must be carefully examined on its own merits. The onus falls on the people involved in the preliminary design of the IPAD system to identify and resolve areas in which transferability can be achieved effectively, and areas where the results are not worth the costs. In the IPAD system design, the objective of minimum system code is consistent with maximum practicable transferability. This issue will be revisited repeatedly in the evolution of the design.

7 CONCLUDING REMARKS

This volume began with an outline of the general characteristics desired of the IPAD system, so as to focus on the underlying requirements.

In Section 2, the role of digital computing was discussed to provide a basis for projecting the computing environment surrounding an IPAD system. There was no mention of hybrid (analog/digital) computing or other real-time applications in the preceding sections. This was not an oversight. The role played by real-time computing is specialized to that application and already possesses many of the features sought for IPAD (e.g., the interactive interface). To include these within the IPAD system would complicate the system design without offering any appreciable advantages to real-time users.

A conceptual design was then evolved (Section 2) to provide a set of goals and requirements for the IPAD design team. The information exchange objectives were outlined in a series of figures intending to portray what an IPAD user might see on his CRT. The feasibility of incorporating an existing OM into IPAD without modification was discussed in some detail in Subsection 2.3.4 and found meriting further consideration.

Section 3 reported on a survey of existing and potential large-scale computer users so as to determine their evaluation as to the desired principal objective of such a system. Although the sample represented a spectrum of job classifications, the results conveyed a surprising uniformity in what was considered the principal objectives and provided additional emphasis (weighting) to the design goals.

Section 4 reported on a comprehensive survey of existing computer applications, again supporting a spectrum of job classifications. Information obtained from this questionnaire was used to substantiate the validity of the sample and provide a realistic data base with which to extrapolate the hardware and software requirements for a fully operational IPAD system. These requirements were then delineated in Section 5 for the interactive terminals (Section 5.2 and 5.3), the host computing complex hardware and software (Section 5.4), and finally the selection of three different manufacturers' computing systems for the design "target" (Section 5.5).

Having established IPAD's conceptual design, some definitive design requirements, and the target computing systems upon which IPAD should operate, there remained to discuss the intent and extent of the design requirement for transferability of the resulting code. This was done in Section 6 where - from a user's standpoint-it was found that "transparency" was the issue of principal concern and even this objective should be moderately restrained.

Upon reflection, what is proposed for IPAD is not an "integrated program for aerospace-vehicle design" but rather a programming "structure (or framework) within which aerospace vehicle design (or any procedurally oriented task sequence for that matter) can be accomplished with speed, efficiency and confidence" (Subsection 2.3.7). Further the term "automation" may be inappropriate to describe IPAD as envisioned.

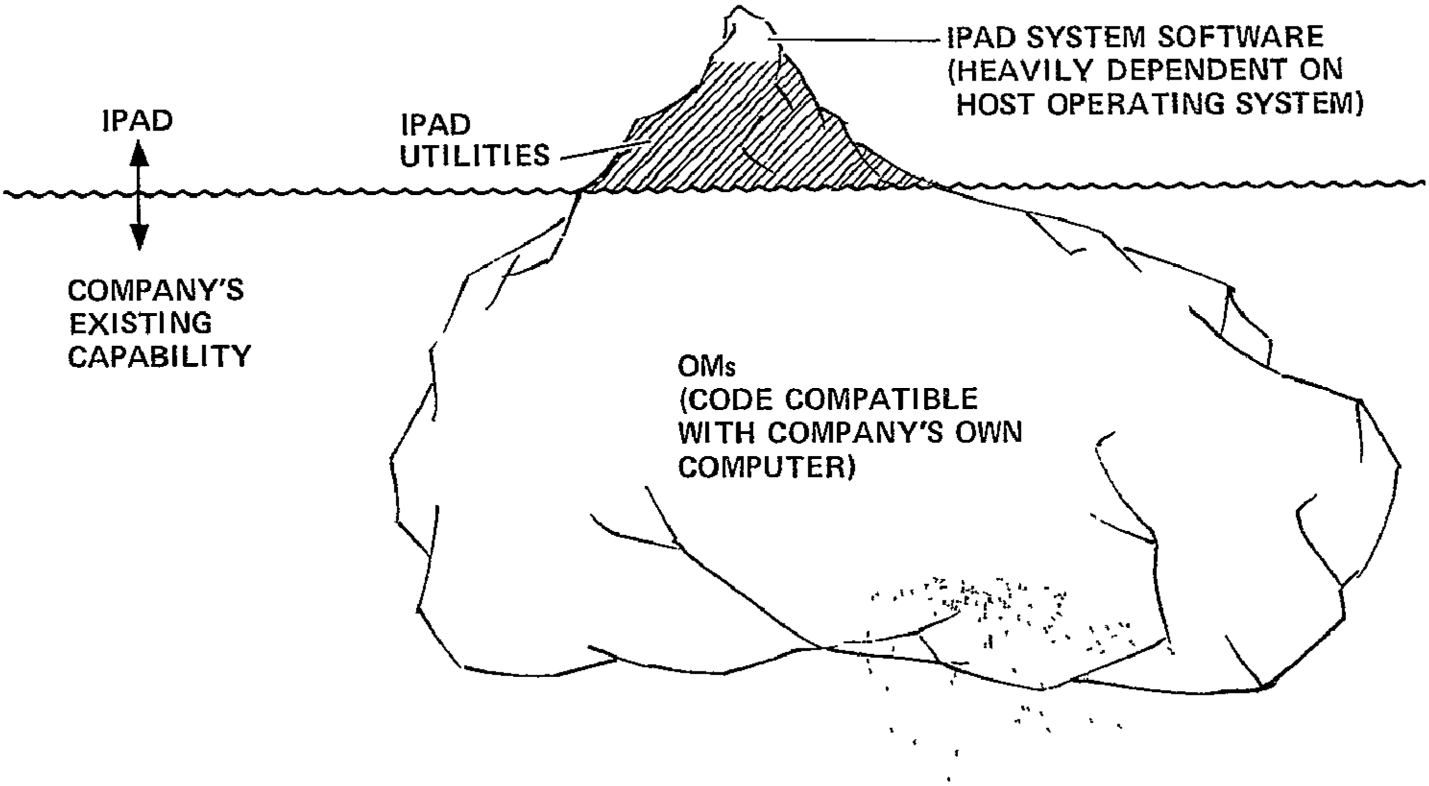


Figure 7-1. Software Iceberg of a Fully Implemented IPAD System

Figure 7-1 illustrates the envisioned fully-implemented IPAD system consisting of:

1. IPAD system software (viz the Executive) which ties directly into the host computer operating system and provides for exploiting this system's

features. This code is likely to be heavily dependent on the various host computer operating systems for which it is designed. As implied by the figure, there is a minimal amount of this host-dependent system code.

2. The IPAD system utilities, e.g. the I/O Formatter Utility (IOF) and the Graphical Plotter Utility, which are envisioned to be fully transferable. The utilities provide the interface between the user and the IPAD system, and augment the capabilities provided by the user's OMs. As implied by the figure, the major IPAD investment lies in the area of utilities.

3. The OMs which represents the company's (and hence the engineering community's) existing capability. As implied by the figure, there is a tremendous investment in existing (principally FORTRAN) OMs. The OM code is compatible with the individual company's own computer (and is often proprietary code).

The IPAD conceptual design has delegated to the computer many tasks which are historically routine management. The typical user is being facilitated substantially. Management must also exploit IPAD if they are going to be able to maintain control over this design process. This means more management OMs and new adaptive management structures.

Towards this end, a track* of each user within the system must be available to project management. This should provide the answers to questions such as:

1. What is your current status?
2. What data was used in this study?
3. Did you account for variation in this parameter?
4. Is the study completed?, and so on.

Volume V picks up where this volume leaves off, reporting on the evolution of the IPAD system design (Part II) and its supporting utilities (Part III).

* This was subsequently coined the "User's Task Trajectory" (UTT).

REFERENCES

1. Lee, V. A.; Ball, H. G.; Wadsworth, E. A.; Moran, W. J.; and McLeod, J. D.: Computerized Aircraft Synthesis. J. Aircraft, vol. 4, no. 5, Sept. - Oct. 1967, pp. 402-408.

2. Herbst, W. B.; and Ross, H.: Application of Computer Aided Design Programs for the Management of Fighter Development Projects. Presented at the AIAA Fighter Aircraft Conference, St. Louis, Mo., March 5-7, 1970, AIAA Paper No. 70-364.

3. Wennagel, G. J.; Mason, P. W.; and Rosenbaum, J. D.: IDEAS, Integrated Design and Analysis System. SAE Paper 68-0728, presented to SAE Aeronautics and Space Eng. Manufacturing Meeting, Los Angeles, California, Oct. 1968.

4. Hurley, M. J.: Report of Committee to Investigate Computer Utilization Within the Applied Mechanics Section, Appendix A, Dynamics Group. Convair Aerospace Division of General Dynamics Internal Memo CAM-9-23, Nov. 1959.

5. Anon: Feasibility Study of an Integrated Program for Aerospace-Vehicle Design (IPAD). Statement of Work 1-15-2183 Exhibit A, National Aeronautics and Space Administration, Langley Research Center, Oct. 4, 1971.

6. Salzman, R. M.: State of the Art in Computer-Aided Design. Paper presented to Panel No. 3 during Panel Reports for DOD/Industry on Computer-Aided-Design and Computer-Aided-Manufacturing/Numerical Control, Davenport, Iowa, Oct. 1969.

7. Kopp, C. G.; and Wertz, J. P.: Designer and Computer: An Engineering Partnership. Battelle Technical Review, Feb. 1968.

8. Anon: Engineer's Manual for Running DYNES. Convair Aerospace Division of General Dynamics, Report GDC-DDE66-007, Jan. 1966.

9. Reed, T. E.; Tonelli, W. C.; and Neuharth, E. R.: DYNAMO 70, DYNamics Assembler MOdeling System. Convair Aerospace Division of General Dynamics, Report SD-70-070, Revision B, July 1971.

10. Hurley, M. J.: Instruction Packet for Completing the IPAD OM Questionnaire. Convair Aerospace Division of General Dynamics, IPAD Internal Memo, May, 1972.

11. Prince, M. D.: Interactive Graphics for Computer-Aided Design. Addison-Wesley Publishing Company, 1971.

12. Worley, A. R.: Practical Aspects of Data Communications. Datamation, Oct. 1969.

13. Telecon between Jim Shneer, Systems Manager, Control Data Corporation, COMSOURCE Division, Los Angeles CYBERNET Center and Mike Hurley of Convair Aerospace Division of General Dynamics, 22 Sept. 1972.

14. Anon: Technical Statement of Work, Space Shuttle Booster, Program Definition (Phase B). Convair Aerospace Division of General Dynamics, Report GDC-DCB70-030, May, 1970.

15 Hurley, M. J.; and Carrie, G. W.: Stage Separation of Parallel-Staged Shuttle Vehicles: A Capability Assessment. Journal of Spacecraft and Rockets, Oct. 1972.

16. Murphy, J. A.: Interactive CRT Terminals, Part I - Full Graphic CRT Terminals and Systems. Modern Data, June 1971, pp 44-55.

17. Murphy, J. A.: Minicomputers. Modern Data, June 1971, pp 58-72.

18. Bonine, Kenneth C.: Economic Considerations of CAD/CAM. Paper presented at SME International Conference on CAD/CAM (Atlanta, Georgia), Feb. 1972.

19. Anon.: Computer Characteristics Review. Volume 11, No. 3 Publication of GML Corporation, 1972.

20. Anon.: Key to Enhanced Application Development. Data Processor (an IBM Publication), Sept. 1972.

21. Rosen, S.: Programming Systems and Languages, 1965-1975. Communications of the ACM, July 1972

22. Anon.. INTERCOM Performance. CDC's PSI Excerpts, April, 1972, pp 42-18, 42-20.

23. Anon.: IBM Virtual Machine Facility/370: Introduction. IBM publication GC20-1800-0, July, 1972.

24. Anon.: CYBER 70, Model 72, Model 73, Model 74 Computer Systems: SCOPE, General Information Manual. CDC publication 60327700A, March, 1971.

25. Anon : VIM-16 Installation Reports. VIM, Inc. publication (published by CDC), VIM-16 Conference, April, 1972.

26. Anon : CYBER 70 Computer Systems General Information Manual. KRONOS. CDC publication 59156600, Feb., 1971.

27. Anon.: Computer Program Manual. National Aeronautics and Space Administration, Langley Research Center. Volume I, Section 2, May, 1969.

28. Sammet, J. E.: Programming Languages: History and Fundamentals. Prentice-Hall, Inc., 1969.

29. Perstein, M. H.: Report to SPARC from AD Hoc Committee on Operating System Control Language. Report X3/SPARC/OSCL 71-4, American National Standards Institute (ANSI), July 6, 1971.

30. Perstein, M. H.: Organizational Structure of X3. Attachment 2 to Committee Correspondence MHP/3.4.2F-128, American National Standards Institute (ANSI), June 7, 1971.

APPENDIXES

APPENDIX A
GLOSSARY OF IPAD ACRONYMS AND SELECTED TERMINOLOGY

APPENDIX A

GLOSSARY OF IPAD ACRONYMS AND SELECTED TERMINOLOGY

Throughout the IPAD feasibility study - as concepts were formulated and designs envisioned - acronyms and special terminology were evolved to represent these concepts in a concise and easily recognized form. These acronyms soon became inseparable from the concepts they represented and found their way into all discussions, presentations and documentations concerning IPAD. Although unfortunate from the casual reader's standpoint, the use of acronyms is a tool which the system designers - especially those involved in conceptual design - tend to rely on and incorporate into their thought processes.

It is with these apologetic thoughts in mind that this appendix, which contains most of the acronyms and special terminology used throughout this report, is presented.

APPENDIX A - Continued

GLOSSARY OF IPAD ACRONYMS AND SELECTED TERMINOLOGY

AED — AUTOMATED ENGINEERING DESIGN...(ALSO KNOWN AS ALGOL EXTENDED FOR DESIGN)...AN ALGOL-LIKE CAD LANGUAGE DEVELOPED AT MIT CIRCA 1959.

ALGOL — ALGORITHMIC LANGUAGE...A PROBLEM-SOLVING COMPUTER LANGUAGE THAT USES THE CONCEPT OF THREE LANGUAGE TYPES...PUBLICATION, REFERENCE AND HARDWARE.

A/N — ALPHANUMERIC...REFERS TO BOTH ALPHABETIC AND NUMERIC CHARACTERS.

ANSC — AMERICAN NATIONAL STANDARDS COMMITTEE...ANY STANDING COMMITTEE OF ANSI

ANSI — AMERICAN NATIONAL STANDARDS INSTITUTE...AN INSITIUTE FOR INDUSTRY AND GOVERNMENT COORDINATION IN THE FIELD OF STANDARDIZATION (APPENDIX D).

APL — A PROGRAMMING LANGUAGE...A PROGRAMMING LANGUAGE ORIGINALLY DEVELOPED BY IVERSON AND PUBLISHED IN 1962. APL IS CHARACTERIZED BY ITS LARGE CHARACTER SET AND PROLIFERATION OF UNUSUAL SYMBOLS.

APT — AUTOMATICALLY PROGRAMMED TOOLING...APT IS A LANGUAGE(OR LANGUAGE TRANSLATOR) FOR GENERATING PROGRAMS FOR AUTOMATICALLY CONTROLLING THE OPERATIONS OF NUMERICALLY-CONTROLLED MACHINE TOOLS.

ARPA — ADVANCED RESEARCH PROJECT AGENCY.

BASIC — BEGINNER'S ALL-PURPOSE SYMBOLIC INSTRUCTION CODE...BASIC IS A SIMPLE COMPUTER LANGUAGE TO LEARN. IT WAS DESIGNED AS A STEPPING-STONE FOR STUDENTS TO LEARN PRIOR TO ONE OF THE MORE POWERFUL LANGUAGES SUCH AS FORTRAN OR ALGOL.

BATCH — BATCH MODE PROCESSING...THE SUBMITTAL OF JOB (GENERALLY VIA CARDS) FOR DEFERED PROCESSING BY THE COMPUTER.

BAUD — A TERM DENOTING FREQUENCY IN SAMPLE DATA SYSTEMS. IN SYNCHRONOUS TRANSMISSION, A BAUD IS A SAMPLE PER SECOND.

BURROUGHS — A BRAND NAME REFERRING TO THE MANUFACTURER BURROUGHS CORP., BURROUGHS PL., DETROIT MICH.

CAD — COMPUTER-AIDED DESIGN...THE PROCESS OF USING A COMPUTER FOR ANALYSIS AND DESIGN WHEREBY THE USER INTERACTS IN A CONVERSATIONAL MODE. HIS PROBLEM-SOLVING PROCESS IS CONTINUOUS AND UNINTERRUPTED.

CALCOMP — A BRAND NAME REFERRING TO THE MANUFACTURER CALIF. COMPUTER PRODUCTS, 2411 W LA PALMA AVE, ANAHEIM CALIF.

CAM — COMPUTER-AIDED MANUFACTURING...USE OF THE COMPUTER FOR ONLINE INVENTORY CONTROL, MANAGEMENT INFORMATION SYSTEMS, AND DIRECT CONTROL OF MACHINE TOOLS.

CDC — A BRAND NAME REFERRING TO THE MANUFACTURER CONTROL DATA CORPORATION, 8100 34TH AVE S, MINNEAPOLIS MINN.

CM — CENTRAL MEMORY.

CMS — CONVERSATIONAL MONITORING SYSTEM...IBM 370/145,158,168 CONVERSATIONAL TIMESHARING SUBSYSTEM (SOFTWARE)....ALSO USED TO DENOTE CAMBRIDGE MONITOR SYSTEM...IBM 360/67 CONVERSATIONAL TIMESHARING SUBSYSTEM (SOFTWARE).

COMPUTEX — A BRAND NAME, REFERRING TO THE MANUFACTURER...COMPUTEX,INC.,CAMBRIDGE MASSACHUSETTS.

CP — CENTRAL PROCESSOR...CDC TERMINOLOGY FOR THE CENTRAL ARITHMETIC UNIT IN A COMPUTATIONAL SYSTEM.

CPU — CENTRAL PROCESSOR UNIT...SEE CP.

CRT — CATHODE RAY TUBE DISPLAY OF INTERACTIVE DEVICE. COULD BE THREE TYPES

- DVST — DIRECT VIEW STORAGE TUBE...RETAINS IMAGE WITH SLOW DECAY UNTIL REPAINT
- REFRESH — REFRESHED CRT...THE IMAGED IS CYCLED 30 TO 40 TIMES/SEC AND TO REFRESH THE CRT TO PREVENT FLICKER.
- TV — TELEVISION (CONTINUOUS RASTER SCAN)...RESOLUTION LIMITED BY RASTER.

CTS — CONVERSATIONAL TIMESHARING SYSTEM...UNIVAC CONVERSATIONAL TIMESHARING SUBSYSTEM (SOFTWARE).

DB — DATA BASE...THE TOTAL REPOSITORY OF IPAD RELATED DATA ON DISC OR OTHER ON-LINE MASS STORAGE WHICH IS UNDER CONTROL OF THE DBMS. MORE SPECIFICALLY, A DATABASE CONSISTS OF ALL THE RECORD OCCURRENCES, SET OCCURRENCES AND AREAS WHICH ARE CONTROLLED BY A SPECIFIC SCHEMA.

DEC — A BRAND NAME, REFERRING TO DIGITAL EQUIPMENT CORP.,MAYNARD,MASS.

DOD — DEPARTMENT OF DEFENSE.

DVST — SEE CRT, DVST.

APPENDIX A - Continued

ECS	EXTENDED CORE STORAGE...CDC TERM FOR CORE STORAGE ACTING AS A BUFFER TO DISC MASS STORAGE OR FOR PROGRAM USE FOR LARGE ARRAYS, ETC.
ERB	ENGINEERING REVIEW BOARD...THE BOARD OR PANEL, A WING OF MANAGEMENT, GIVEN RESPONSIBILITY FOR REVIEWING THE ENGINEERING DESIGN FOR ADEQUACY AND CONTROLLING THE TECHNICAL ASPECTS OF THE PROJECT.
EXEC	REFERRING TO THE SOFTWARE (CODE) WHICH PERFORMS THE IPAD EXECUTIVE FUNCTION.
FORTRAN	FORMULA TRANSLATING SYSTEM...A SCIENTIFIC COMPUTER LANGUAGE ORIGINALLY CREATED BY IBM FOR USE ON THEIR COMPUTERS.
GERBER	A BRAND NAME REFERRING TO THE MANUFACTURER GERBER SCIENTIFIC INSTRUMENT CO. 83 GERBER RD, WAPPING CONN.
GPGT	GENERAL PURPOSE GRAPHICS TERMINAL...CDC SUCCESSOR TO THEIR LARGE CRT 274 TERMINAL (EXPECTED DELIVERY,FALL 1973).
IBM	A BRAND NAME REFERRING TO THE MANUFACTURER INTERNATIONAL BUSINESS MACHINES CORP. OLD ORCHARD RD. ARMONK NY.
ID	IDENTIFICATION.
IDI	A BRAND NAME REFERRING TO THE MANUFACTURER INFORMATION DISPLAYS INC., MOUNT KISCO,NEW YORK.
IGS	INTERACTIVE GRAPHICS SYSTEM...CDC TERM FOR THEIR INTERACTIVE GRAPHICS SYSTEM SOFTWARE (FORTRAN CALLABLE). CURRENT VERSION IS 2.0.
IMLAC	A BRAND NAME REFERRING TO THE MANUFACTURER IMLAC CORP. 296 NEWTON ST., WALTHAM,MASS.
INTERCOM	THE INTERACTIVE (TIME-SHARING) SUBSYSTEM OF CDC'S SCOPE 3.0 AND ON.
I/O	INPUT/OUTPUT...REFERS TO BOTH INPUT AND OUTPUT (EG,I/O FILES).
I/ODEF	THE COMBINATION OF THE IDEF AND ODEF OF AN OM.
IPAD	INTEGRATED PROGRAM FOR AEROSPACE-VEHICLE DESIGN. USED 2 DISTINCT WAYS 1.FULLY IMPLEMENTED...CONTAINS FULL COMPLEMENT OF OPERATIONAL MODULES. 2.SYSTEM SOFTWARE...JUST SUFFICIENT SYSTEM CODE FOR IPAD (NO OM'S). IPAD IS A SOFTWARE FRAMEWORK INTENDED TO REDUCE THE TIME AND LABOR EXPENDED BY AN (AEROSPACE) ENGINEER IN ACCOMPLISHING HIS ENGINEERING TASK.
ISO	INTERNATIONAL ORGANIZATION FOR STANDARDIZATION...AN INTERNATIONAL STANDARDS ORGANIZATION. SEE APPENDIX D FOR DETAILS.
IS+R	INFORMATION,STORAGE AND RETRIEVAL...THE TECHNOLOGY FOR CODING INFORMATION FOR EASY STORAGE AND RETRIEVAL.
JOVIAL	JULES' OWN VERSION OF THE INTERNATIONAL ALGEBRAIC LANGUAGE...JOVIAL IS A PROCEDURE-ORIENTED, PROBLEM-ORIENTED AND PROBLEM-DEFINING LANGUAGE. IT SERVES SIMULTANEOUSLY AS A REFERENCE, PUBLICATION, AND HARDWARE LANGUAGE.
LSI	LARGE SCALE INTEGRATION...AN ELECTRONICS APPROACH TO MICRO'MINITURE CIRCUITRY.
LUNDY	A BRAND NAME,REFERRING TO THE MANUFACTURER LUNDY ELECTRONICS + SYSTEMS INC.,PARAMUS,NEW JERSEY.
MACRO	LARGE OR OF THE HIGHEST ORDER.
MAXI	USUALLY REFERRING TO A LARGE SCIENTIFIC COMPUTER SYSTEM (INCLUDING CONSIDERABLE SUPPORTING SYSTEM SOFTWARE).
MDB	MULTIDISCIPLINARY DATA BANK...THE DB AREAS RESERVED FOR 'BLESSED' DATA WHICH IS READ-ONLY FOR THE USER AND UNDER THE STRICT PROJECT CONTROL OF THE DBA AND HIS PEOPLE.
MENU	A TABLEAU OR LIST OF ITEMS ON A GRAPHICS TERMINAL,ONE OR MORE OF WHICH ARE MEANT TO BE SELECTED, GENERALLY BY A TOPOLOGICAL INPUT DEVICE, EG A LIGHT PEN.
MICRO	SMALL OR OF THE LOWEST ORDER.
MIDI	USUALLY REFERRING TO A MEDIUM COMPUTER, LARGER THAN A MINI BUT SMALLER THAN A MAXI. THE DIVISION SEEMS TO BE MORE ON PHYSICAL SIZE AND COST THAN ON COMPUTING CAPACITY.
MINI	USUALLY REFERRING TO A SMALL COMPUTER USED STANDALONE OR AS A PERIPHERAL TO A MAXI
MUX	MULTIPLIEXER...A DEVICE WHICH ALLOWS TWO OR MORE OPERATIONS TO BE CARRIED ON CONCURRENTLY.
NAVAIR	NAVAL AIR SYSTEMS COMMAND - WASHINGTON, D.C.
NAVSEC	NAVAL SHIP ENGINEERING CENTER (OFTEN CREDITED WITH STARTING CAD)-WASHINGTON, D.C.

N/C	NUMERICAL CONTROL...REFERS TO THAT CLASS OF MACHINES (EG A PROFILE MILL) WHICH ARE PROGRAMMED VIA DIGITAL MAGNETIC TAPE TO DEFINE THE PATH OF THE CUTTER.
NCR	A BRAND NAME REFERRING TO THE MANUFACTURER NATIONAL CASH REGISTER.
OM	OPERATIONAL MODULE...A FULLY FUNCTIONAL PIECE OF CODE WHICH CAN (HAS) RUN STANDALONE IN BATCH MODE OR IS A FULLY CHECKED OUT INTERACTIVE PROGRAM. AN OM IS USUALLY A FULLY OPERATIONAL EXISTING FORTRAN BATCH PROGRAM WHICH REPRESENTS A PORTION OF A DISCIPLINE'S CAPABILITY.
OSCL	OPERATING SYSTEM CONTROL LANGUAGE...THE LANGUAGE FOR CONTROLLING A COMPUTER'S OPERATING SYSTEM (SEE APPENDIX C FOR DETAILS)....ALSO AN AD HOC COMMITTEE REPORTING TO SPARC (SEE APPENDIX D).
PCB	PRINTED CIRCUIT BOARD...AN ELECTRONIC COMPONENT CONSISTING OF DISCRETE COMPONENTS MOUNTED ON A BOARD.
PDP	PREFIX TO THE NAMES OF COMPUTERS MADE BY DIGITAL EQUIPMENT CORP. (DEC)
PERT	PROGRAM EVALUATION REVIEW TECHNIQUE...A SYSTEM FOR SCHEDULING ACTIONS AND DEFINING TIME-CRITICAL PATHS.
PL/I	SEE PL/1
PL/1	PROGRAM LANGUAGE 1...PL/1 IS BOTH A PROCEDURE'ORIENTED LANGUAGE AND A PROBLEM ORIENTED LANGUAGE IN A BROAD SENSE. IT SERVES SIMULTANEOUSLY AS A REFERENCE, PUBLICATION, AND HARDWARE LANGUAGE. PL1 WAS INTRODUCED BY IBM IN OCTOBER, 1967.
PP	PERIPHERAL PROCESSOR...CDC TERMINOLOGY FOR A SEPARATE PROCESSOR USED IN CDC COMPUTER SYSTEM DESIGN TO HANDLE ALL I/O ACTIVITIES PERIPHERAL TO THE CENTRAL PROCESSOR.
RAT	A RISK ASSESSMENT TERM...A GROUP OF ENGINEERS ASSIGNED TO ASSESS THE CONSEQUENCES OF PROPOSED STRATEGY.
REDCOR	REDCOR, INC., NOW DEFUNCT, FORMERLY OF CALIFORNIA.
REFRESHED	SEE CRT, REFRESH.
RJE	REMOTE JOB ENTRY...ENTRY TO A CENTRAL COMPUTER, EG BY TELEPHONE LINES VIA A PERIPHERAL DEVICE LIKE A CARD READER.
SCOPE	THE HOST COMPUTER OPERATING SYSTEM FOR THE CDC CYBER 70 SERIES MODELS 72 THRU 74. (ALSO KNOWN AS CDC 6000 SERIES COMPUTERS.)
SPARC	STANDARDS PLANNING AND REQUIREMENTS COMMITTEE OF ANSC X3 (SEE APPENDIX D FOR THE RELATIONSHIP).
TAT	THREAT ASSESSMENT TEAM...A GROUP OF ENGINEERS ASSIGNED TO ASSESS THE THREAT OF POSTULATED ENEMY STRATEGY AND CAPABILITY.
TCS	TASK CONTROL SEQUENCE...ESSENTIALLY A COMMANDS FILE PROVIDING THE ABILITY TO EXECUTE JOB SEQUENCES AUTOMATICALLY. TCS FILES ARE CODED FILES WHICH CAN BE EDITED BY THE SYSTEMS TEXT EDITOR AND EXECUTED BY THE IPAD EXECUTIVE AS REQUIRED.
TEKTRONIX	A BRAND NAME REFERRING TO THE MANUFACTURER TEKTRONIX,INC.,BEAVERTON, OREGON
TI	A BRAND NAME REFERRING TO THE MANUFACTURER TEXAS INSTRUMENTS.
TTY	REFERS TO THE GENERAL TYPE OF INTERACTIVE CONSOLE USING THE STANDARD TELETYPEWRITER CHARACTER TRANSMISSION INTERFACE OR AN EXTENSION OF IT. ALSO, A BRAND NAME REFERRING TO THE WELL KNOWN TELETYPEWRITER.
TV	SEE CRT, TV
UNIVAC	A BRAND NAME REFERRING TO THE MANUFACTURER UNIVAC, A DIVISION OF SPERRY RAND CORPORATION, 1290 AVE OF THE AMERICAS, NEW YORK, NY.
USER	ANY IPAD USER, GENERALLY CHARACTERIZED AS AN ENGINEER NOT EXPERIENCED WITH COMPUTERS--AT LEAST TO ANY GREAT EXTENT--BUT INTIMATELY FAMILIAF WITH THE PROBLEM HE IS TRYING TO SOLVE.
VARIAN	A BRAND NAME REFERRING TO THE MANUFACTURER VARIAN DATA MACHINES, A VARIAN SUBSIDIARY, 2722 MICHELSON DRIVE, IRVINE, CALIF.
VECTOR GENERAL	A BRAND NAME REFERRING TO THE MANUFACTURER VECTOR GENERAL, INC., CANOGA PARK, CALIFORNIA.
VM	VIRTUAL MEMORY...ON A TIME-SHARING SYSTEM,THE STORAGE SPACE EACH USER APPEARS TO HAVE FOR HIS OWN USE. ALSO VIRTUAL MACHINE... AN IBM 370 CONCEPT IN WHICH EACH OM APPEARS TO HAVE ITS OWN COMPLETE MACHINE (INCLUDING A SPECIFIC OPERATING SYSTEM) FOR ITS OWN USE.
2D	TWO-DIMENSIONAL.
3D	THREE-DIMENSIONAL.

APPENDIX B
INDUSTRY EXPERIENCE WITH INTERACTIVE GRAPHICS, A LITERATURE SURVEY

APPENDIX B

INDUSTRY EXPERIENCE WITH INTERACTIVE GRAPHICS, A LITERATURE SURVEY

The use of interactive graphic systems (IGS) dates from the mid-fifties when the military used IGS in the SAGE air defense system; nonmilitary use dates from the early sixties when Ivan E. Sutherland developed "Sketchpad" at MIT (Reference B1). Although General Motors has been doing work in this area since 1959 (Reference B2), the first published data was Sutherland's in 1963. The use of these systems has grown steadily, finding application in computer-aided design, mathematical equation solving, simulation, management information systems and computer-aided-manufacturing and numerical control programming. Some typical applications and experience of other interactive graphic users are described in the following sections to illustrate how an interactive graphic system can be used and where it has been effective.

B.1 Applications in Analysis

The greatest diversity of application of interactive graphics is in the area of analysis. This is a consequence of the degree of computerization the analyst enjoys; most of his analytic tools are currently computerized and he is conditioned to exploit the state-of-the-art in computing for his own ends. The following are a few diverse applications in this fertile field.

B.1.1 Simulation programs. - Some of the problem-oriented, continuous system and discrete-event programming languages have been adapted to interactive graphics (Reference B3). Among these are MIDAS IV, 360 CSMP, MIMIC and GPSS (op. cit.). These programs are natural for this type of interactive use, expecially the ones which are block-diagram oriented. Electrical network analysis programs such as SCEPTRE, (Reference B4) could also be included here. Basically, these programs allow the user to build a model of the system he wants to simulate by linking together symbols, blocks and statements in a language suited to his problem. The computer then generates the machine code necessary for simulating the dynamic behavior of the system and, with appropriate parameter values entered by the user, generates a solution of the equations describing the system. These solutions may be selectively viewed by the user on the graphics scope for acceptance or modification, as he sees fit. Interactive graphics systems have also been used in connection with hybrid simulations with a man-in-the-loop type of study, and for ease of manipulating and viewing three dimensional output.

All aerospace companies use simulation techniques and most utilize one or more of the simulation languages (continuous or discrete systems). These techniques provide time savings in themselves of 2:1 to 4:1; multiply that by a factor of 2 to 5 as a result of interactive computing and the savings are really significant!

Interactive graphics techniques were used at MacDonnell-Douglas, Long Beach on the DC-10 as described by J. B. O'Neil (Reference B5).

> GRAPHIC CSMP (Continuous System Modeling Program) was written to give the engineers the capability of interacting with the CRT because it is used extensively for design and analysis work which, to be effective, requires interaction. ... The data can be changed during the execution phase while sitting at the CRT, allowing a homing-in process for design and analysis.

Lockheed-California used digital simulation techniques (CSMP) adapted to interactive graphics for the simulation of the Lockheed's L-1011 Tri-Star transport during take-off and landing. Three specific areas investigated were the handling qualities, dynamic loads and guidance and control. Mr. Sturcke comments on the interactive aspects were as follows (Reference B6).

> The main drawback to digital simulation is the lack of on-line solution monitoring and management. ... Digital simulation languages have all the advantages of a large digital computer, but still have the disadvantage that they do not have the man-machine interface that is so highly desirable in the simulation field. ... This disadvantage has been eliminated by adapting CSMP to the IBM 2250 graphic display [terminal].

A recent experience at General Dynamics Convair Aerospace (December 1971) is another example of the power of simulation languages when coupled with interactive graphics. The control dynamics group had three days to perform two complete engine gimbal angle studies after receipt of data; the normal batch method takes one week from receipt of the data. This was accomplished with an interactive graphics version of the batch simulation program. The task started on Thursday and was completed by Saturday noon with a full set of hard copy and viewgraphs of exactly what the controls analyst wanted to show; he left with these on Sunday for a Monday meeting at NASA in Houston where the data proved that one configuration was not feasible. In addition to doing a task which could not have been done in the time available, there was a 41 percent reduction in total cost.

B.1.2 Structural analysis. - Interactive graphic has proven to be a very useful and successful tool for structural analysis of aerospace vehicles. Existing analysis techniques involve large amounts of input data, most or all of which are variables depending on geometry, mass and stiffness. Lockheed-Georgia (References B7 and B8) began looking at computer graphics in 1964 to assist in the analysis of the complex structure of todays aerospace vehicles. Successful implementation of interactive graphics structural analysis programs was described by Lockheed-Georgia engineers (References B9 and B10) and resulted in a reported first year cost savings of $252,000 (Reference B11).

General modal analysis, where the user can vary the mass, mass distribution, geometry, stiffness distribution, etc., and produce mode shapes and generalized frequencies for the analysis of structures (e.g. via the techniques of forced response, frame analysis, stress analysis, flutter and control system interaction), are typical of problems being solved by structural/stress analysts at General Dynamics Convair Aerospace. When the size and running time of some of the large programs preclude their direct use interactively, a batch run may be done and the output written onto a disk file. Then perturbation techniques can be applied to obtain subsequent solutions in a short enough time to allow a user to observe the graphic (and sometime animated) display of the output on the CRT in an interactive fashion.

Describing the input bars, shear panels, and nodes to a finite element analysis program is a tedious job. A typical input deck may run 500 to 2,000 cards. Programs of this size and complexity offer many opportunities for input coding errors: misnumbered nodes, failure to right justify an exponent in an E-field, leaving out a shear panel, etc. Scanning the cards by eye (or even a listing of them) takes time, is boring, and is not likely to uncover all error's that may exist. The ability to view the model in its geometric presentation allows the engineer to detect errors in seconds that he might spend an hour looking for (and perhaps not find) on the cards themselves. Lockheed-Georgia reported saving 500 hours of aborted computer runs during the analysis of a single aircraft wing using this visual check capability (Reference B11). This same model building and analysis program was used at Convair in the batch mode and subsequently programmed for interactive graphics. It was used interactively on a project that had a 90 day schedule to meet with total cost savings of $23,490, a cost ratio of 5.4 to 1. Table B1 (from Reference B12) typifies the cost savings resulting from interactive graphics.

Convair's experience in developing and using interactive computer graphics programs has shown that they are useful tools for checking finite-element models prior to structural analysis and for displaying the results in a meaningful fashion. These techniques have been successfully applied to NASTRAN resulting in model checkout cost savings and increased flexibility in the use of NASTRAN. Mike Cronk

TABLE B1

COMPARISON OF CALENDAR TIME, MANHOURS AND COMPUTER COST BETWEEN BATCH AND INTERACTIVE GRAPHICS

	BATCH	INTERACTIVE GRAPHICS	BENEFITS
Calendar Time	3 months	3 weeks	Kept schedule that could not have been made using batch techniques. Also assisted other disciplines by providing results of structural analyses.
Manhours	13 manweeks ($8,840)	5 manweeks ($3,400)	Saved $5,440 in engineering labor cost. Reduced fatiguing rote work of engineer so he could concentrate on engineering problem rather than proof-reading key-punched data cards.
Computer Costs	$20,000	$1,950	Represents about 40 hours of aborted runs on a CDC 6400 computer for the batch work. For the graphics, there was about $1,200 of computer time and only $750 (25 hours) of CRT time.

stated at the NASTRAN users' colloguium in September 1971 (Reference B13):

> Interactive computer graphics offers several improvements over the normal digital mode of operation. Turnaround time is reduced by the ability to display "immediately" the results of computation, modify the data and recompute. Several iterations of the submit job, compute, look at printout, correct data and resubmit process can be compressed into one session at the display device. Judgement and insight to the problem are enhanced by using the graphic display. Rapid comparison of various designs or analyses, often essential during preliminary design or "panic" redesign, are readily accomplished.

B.1.3 Control system analysis. - Application of interactive graphic techniques to control system analysis is simple and straightforward. For the most part, the controls engineer knows what elements of his problem are germane to his decision process, and these must be made visible and controllable in the display on the CRT. Provision must also be made for data input at the terminal and program control (restart, terminate, selective output, etc.). This capability is coded into the applications program using subroutines provided by the graphic support package. The first step in using the interactive graphics terminal is usually to simply use it as a rapid display and program change device and continue to use the current methods of solving control problems. The insights gained through this mode of operation and the new capabilities provided however have stimulated innovations and changes in problem-solving methodology that may have as profound an effect on the state-of-the-art in design and analysis as the introduction of interactive computing itself. One of the control system analyst's primary tools or techniques, simulation, has been discussed above. Another and equally important tool is the use of transfer functions to describe frequency response and stability characteristics of a system.

An outstanding program of this type has been developed by Steve Bayer and Ken Yankeleivtz of Douglas Aircraft Co., Long Beach, California (Reference B14). Their program has a data base with standard transfer functions stored in it, and new ones can be added or read in at execution time from cards, or input via the graphic console. The user can select the transfer functions to work with; then from a menu, select the method or order in which they will be cascaded and what type of analysis is desired. The analysis selection consists of frequency response with Bode Plots, Root Locus with a choice of either a scan-type or root-solving solution, and Nyquist diagrams. The program computes the over-all transfer function and displays it and the appropriate plot on the scope. The coefficients of the Laplace operator, S, may be changed from the console and a new solution generated within a few seconds. The new solution may be displayed superimposed on the previous one and the scale factors on the plotting grid expanded or reduced, at the console, to suit the user. Immediate hard copy is available via a polaroid camera that swings down in a special hood over the face of the CRT. Also, a microfilm processor (SC-4020) may be used, however, this is deferred processing not immediately available. Mr. Bayer states (Reference B14):

> The direct-lift control method will be used to improve the conventional control characteristics of the DC-10. . . . The analytical difficulty results from the interaction between the spoilers and elevators. . . . [The] high speed of the program allows the engineers to easily observe this interaction and quickly design appropriate filters. The insight into the interaction effects gained from the use of the program enables the engineer to design better systems. . . . During the AWACS proposal development, it was necessary to obtain a quick analysis of the aircraft rudder response due to the unusual dutch-roll characteristics. . . .

> The entire program was solved short of three hours which included less than one hour at the graphics terminal. Without this program, the study would have required several days . . . through the use of conventional methods.

Convair, San Diego has a very similar interactive graphics program used for space vehicle and aircraft stability studies. Herb Greiner (Reference B15) writes:

> One of the most powerful tools applicable to the analysis and synthesis of launch vehicle flight control systems is the root locus technique. Its chief virtue is the graphical display of the motion of system roots as functions of various system parameters. This enables the control engineer to evaluate the effects of these changes directly (the root locations can be translated immediately into an estimate of the system time response) or if desired, its frequency response.

Nearly all of the 49 root locus figues of Reference B15 were produced by the graphics program (CSAP) in either an interactive or batch mode.

B.1.4 Aircraft predesign. - Interactive graphics has been successfully applied to the aircraft preliminary design process at Lockheed-Georgia, Lockheed California and to varying degrees of success at many other aerospace firms. Since details of implementation and design criteria and consideration are usually considered proprietary, very few have been published on this subject. A generalized description of an overall system is given by David Prince of Lockheed Georgia in Reference B16. Mr. R.Q. Boyles of Lockheed-Georgia's Preliminary Design states (Reference B17).

> The development of interactive, graphic display devices as interfaces between engineers and computers has opened a new era in man-computer synergetics. Merging the attributes of the digital computer with those of the human in an environment in which there is a freely flowing interchange of information promises great improvements in creativity and productivity. The development of a computer-aided preliminary design and performance estimation capability will open up a new dimension for the aircraft design engineer. In this environment, the man and the computer will form a partnership which has virtually unlimited potential for creative design.

In recent years Convair Aerospace has actively engaged in developing interactive computer graphics programs for use as tools in the design synthesis and analysis process. Significant benefits to be realized through application of interactive computer graphics include a large reduction in engineering manhours, substantial savings in computer time, and a reduction in turnaround time between job submittal and the resultant data. The synthesis and analysis process is continuous - conserving user mental momentum and stimulating innovative and creative thinking, which is often hindered by the tedium associated with offline or batch mode computer usage. One present application uses interactive graphics with a geometry and weight synthesis program (References B18 through B20) which is adapted for use with the more general Combat Aircraft Synthesis program (Reference B20). These programs (in everyday use at Convair Aerospace) are intended to facilitate design tradeoff studies of advanced vehicle concepts at the preliminary design level. Synthesis program input is initially submitted and the iterative computation is begun and controlled from the graphics display console by the user. Weight and geometry data of the resultant sized vehicle is displayed at the console. From the console, the user has the option of calling for a printout of the complete data output, making a change to the input parameters from the console and reinitiating the sizing process, calling a plot subroutine and plotting data saved through a series of sizing trials, or calling a planform subroutine to display the planform corresponding to the sized vehicle. A second graphics program, currently being interfaced with the vehicle synthesis program, is an aircraft weight and balance routine. The user "flies" a specified mission with a given aircraft configuration, fuel load distribution, and payload or weapon load distribution. The resultant shift of center of gravity location throughout the mission is displayed as a function of wing mean aerodynamic chord. The user may modify the mission or configuration and see the result of changes made. This extensive experience and everyday use of interactive graphic programs in preliminary aircraft design has resulted in a contract to the Air Force Flight Dynamics Lab at Dayton, Ohio (Reference B22):

> The objective of the contract is to provide a computer program with interactive graphic display capability that will rapidly accomplish the preliminary design phase of an advanced fighter, bomber, or cargo aircraft.

The existing program which will be used as a basis simulates the performance of an aircraft in various flight configurations and phases. A basic configuration - weight and balance data, engine models, atmosphere models, etc., - may be input via cards or called from a data bank previously stored on a mass storage device. The user can modify the geometry, shape, location of control surfaces, weights, etc., from the console. He then selects a phase of flight such as takeoff, climb, cruise, etc., and an evaluation criterion: Military or Civil. As the calculations are being performed, a plot of any parameter versus another may be displayed on the CRT. The results of the modified aircraft or engine model, etc., may be visually compared with the unchanged model, and changes made until a satisfactory result is obtained (Reference B23).

C-3

B. 1. 5 Equation solving. - Problems in fluid flow, radiative energy transport, reactor control and many other areas of practical interest lead to nonlinear partial differential equations to describe the system operation and the control laws. Development work along this line has been done at the University of Utah, sponsored by the Advanced Research Project Agency (ARPA). (See Reference B24). The developed graphical capabilities permit a rapid and complete study of the nonlinear systems response. The time history of the solution as well as relative and absolute computational errors are displayed on the scope. This allows the user to monitor the performance of the algorithm and modify it as required while watching the computed solution. In a batch mode, an unsatisfactory convergence criterion or step size could result in a bad solution or an aborted run; whereas, in the interactive mode, these parameters can be modified as soon as the improper behavior is detected and the run continued to an acceptable solution. Also, the ability to rotate the trajectories (solutions) so that their positions in three dimensional space may be viewed, provides additional insight into the problem.

B. 1. 6 Curve fitting. - When curve (or surface) fitting, graphic display is given of a curve to determine the best fit to displayed data. The user can typically edit the data by erasing unwanted points and adding new ones if desired. Several types of curve fits are generally available (least-squares polynomial, orthogonal, splines, smoothing spline, etc.) and precise numerical information is generally available for both error measurement and analytic expressions defining the curve. An interactive graphics curve fitter is a useful multidisciplinary tool for general data display and editing.

Variants have been tailored to allow for wind tunnel data reduction where the amount of data could not be handled by hand and batch processing with offline graphics took a long time and was very costly. Convair has developed such an interactive graphics program for the storage, retrieval and analysis of experimental wind tunnel data. Experience with this program has shown a cost saving of 5:1 over manual methods. Chuck Whitney states (Reference B25).

> With just the curve-fitting, crossplotting and differencing capability the majority of data can be reduced through its first analysis step. The use of this analysis program on the STAI contract has shown a significant reduction in both time and cost when compared to previous hand methods of analysis. With the addition of more sophisticated data analysis modules a further savings would be realized.

B. 2 Applications in Design

Although not as diverse as the applications in analysis, interactive design programs have the largest potential for cost reductions due to the limited computerization of these areas and the large number of manhours expended. The following subsections present a few applications in design.

B.2.1 Board design. - Interactive computer graphics enabled Lockheed's Tri-Star to meet flight schedule. A staff reporter of Product Engineering reports (Reference B26) Lockheed's graphics experience as follows:

> Although beset by problems not all of its own making, Lockheed's L-1011 Tri-Star has met its scheduled commercial flight date. A large part of the credit goes to the extensive use of computer graphics as used by Lockheed (California Co.). As an example of the versatility of the design package, [S.T.] Horn [Manager of Computergraphics staff] points to the family of 110 floor beams on the L-1011. On a see-what-the-difference-is basis, the design package was used to develop 15 of these beams on a computer graphics console. Result: Finished drawings, ready for production were delivered six times faster on those done by interactive graphics than on those done by conventional means. Another example, Horn says, occurred when design load changes cropped up early in the design schedule. The changes could have presented a scheduling problem, but by using their interactive computer graphics equipment, the engineers unplugged the bottleneck in three weeks; less time than conventional methods would have taken, and held to the original schedule. Still another example was 1300 wiring diagrams that are different for each of the airlines and literally required thousands of drawings in a format new to the company. Again, interactive computer graphics were used and saved an estimated half-million dollars compared to doing the job by conventional means.

The Sikorsky Aircraft Division of United Aircraft Corporation reports (Reference B27) that computer graphic terminals have been used to whet the designers effectiveness since 1967. Drafting, stress analysis, gear set design and numerical control (N/C) tape generation are but a few of the applications using interactive graphics. As the scope of computer graphics use has grown, engineering management has modified its original evaluation of why the investment should be undertaken. Sikorsky Aircraft further stated (Reference B27):

> Qualitative gains originally expected have been achieved. The interactive, dialogue-like process of using a computer through a graphic terminal has produced a marked improvement in design results. But in addition, even the first months of graphic-terminal operation began to show a significant cash return on investment, in the form of savings in engineering manhours and in elapsed time to completion of new designs. The interactive dialogue-like process of using a computer through a graphic terminal has

> produced a marked improvement in design results [and] savings in engineering man-hours and in elapsed time to completion of new designs . . . As the engineer's scope of experience broadened, his understanding of his design jobs increased . . . errors and the retracing of steps were minimized.

Structural analysis and design of planetary gears had accounted for the majority of Sikorsky design work at the date of the article (February 1971). However, as a result of this experience Sikorsky is revising existing analytical computed programs to include interactive graphic capabilities for the designer.

An overview of interactive graphics at Lockheed-Georgia over the past nine years is presented by Dave Prince (Reference B16) with special emphasis on design:

> . . . we begin to envisage the long-term impact of computer-aided design. Tedious and error-prone translation of data from manually compiled records to punched cards will be minimized. The use of a common engineering data base will all but eliminate the time delays of printing out computer listings, changing the form of the data for the next program and recording it . . . Many repetitive analyses performed today by frustrated engineers will be carried out in a tenth of the time, using graphics. Thus the engineer will be freed to spend a larger portion of his time developing and verifying more efficient and accurate analysis methods.

B.2.2 Airfoil synthesis. - The conventional design of wing flaps constitutes a relatively long and expensive process. Wind tunnel models are designed and then tested in two-dimensional test sections. The costs of obtaining one set of data for a given flap geometry generally involves tens of thousands of dollars and approximately half a year in model construction, test and data reduction time.

The introduction of interactive graphics has reduced the cost and time delay associated with high-lift system design and development. The operator becomes his own designer and changes high-lift section geometry on his display tube by means of a light pen. A high-speed digital computer calculates pressure distributions about the individual wing and flap sections and integrates the pressures to obtain forces and moments. The latter are displayed to the operator together with pressure distributions for each section. The operator considers the lift and drag levels obtained and examines the pressure distributions, expecially in reference to pressure peaks. He then has the option of changing section gaps, overlaps, deflections or contours. At the conclusion of his geometric manipulations, he can obtain another set of aerodynamic data. He compares the new set to the previous data. This permits him to draw certain conclusions about the effect of his geometric change on lift, drag, pitching moments and pressure distributions. After several cycles, the operator is able to arrive at the required lift and drag characteristics.

Lockheed-Georgia developed a program for an airfoil design which provides airfoil shape, pressure distribution on upper and lower surfaces, boundary layer conditions and lift and drag coefficients versus angle of attack. The user can change parametric data and can change the shape of the curves by drawing construction on the CRT with the light pen (Reference B28).

An interactive graphics system employed by Convair Aerospace permits the preliminary design of an airplane high-lift system at reasonable cost in a very short time period. It is not designed to eliminate wind tunnel testing, however, it is expected to reduce the time and cost of such tests by eliminating the obvious non-optimum configurations from tunnel test schedules. Recently these graphic techniques were used in the design of wing flaps for the high lift system of the Space Shuttle (Reference B29).

The major shortcoming of such systems - that of accuracy - is not related to the graphics or computer capability but is caused by inadequate aerodynamic (mathematical) models. With the projected advances in modeling of flow fields, this shortcoming should be overcome, and the basic procedure of developing aerodynamic configurations with the aid of a computer and interactive graphics will have a major impact on the development of future airplanes and space vehicles.

B.2.3 Printed circuit board packaging. - All the major computer manufacturers use printed circuit board (PCB) packaging processes involving interactive graphic techniques somewhere in the design, layout, production and testing cycle. The PCB packaging objective is to effect an optimum placement of components so as to minimize the wire lead lengths between components. Several other criteria are also involved, including priorities in placement, use of certain types of components, heating considerations, and exterior pin connections. Probably the most fully integrated IPAD-like operation is that used by IBM in the design and production of circuit cards and boards as described in Reference B30:

> The procedure begins with design engineers setting up logic requirements in the form of functional block diagrams. An engineer then turns to a computer that has been programmed to choose a suitable design based on program requirements and a file of previously prepared designs. . . . A second part of the system uses the design information to generate instructions for NC machines that produce and test the entire card or board. . . . The system also provides information on routing, stock, costs and administrative data.

The General Dynamics Electro Dynamics Division in Pomona initiated a five yea program in 1968 to automate the design and manufacturing of flexible harnesses and circuit cards used in the Block V Standard Missile and Standard Arm Projects (Reference B31) In 1970 Pomona design personnel working in conjunction with General Dynamics Convair Aerospace Division's interactive graphic specialists at San Diego develo ed a pilot model to demonstrate the feasibility of packaging printed circuit boards (PC] on an interactive graphics terminal. The pilot model indicated a potential 80 percent reduction in manhours and turnaround time for analog printed circuit boards. In addition to savings in cost and turnaround time in packaging, the interactive graphics data files are available to produce numerical control drill tapes and PCB artwork for hole layout drawings, component assembly drawings, and Gerber photoplot tapes, whi(would represent additional time and cost if done manually (Reference B26).

B.2.4 Circuit synthesis and circuit mask layout. - Motorola, Lockheed Corporation, IBM, CDC and others are actively using interactive graphic programs for circuit synthesis and circuit mask layout. The rapid graphic feedback to the designer significantly reduces the time required to do these jobs. One of these experiences reporte(in detail in Reference B32 concludes:

> A computer-graphic, hybrid microcircuit mask design program has been developed by the Lockheed-Georgia Co. to provide a rapid and economical means of designing and redesigning circuit masks for hybrid, thick film microcircuits used on the C-5A aircraft. The program provides the mask designers with the capability to transfer information directly from a schematic diagram to a digitized format by means of a cathode-ray tube graphical input/output device. The computer relieves the operator of routine chores and allows him to act essentially as a task manager, making decisions as to component placement. . . . manhour reduction ratio afforded by the computer system over manual design is better than 5:1.

Many papers have been written on the feasibility, specifications, and actual use of interactive graphics in conjunction with electrical network and circuit analysis. On of the earliest proponents of interactive graphics for network and circuit analysis was Michael L. Dertauzos of MIT. His works were used by industry as the basic guidelin in formulation of interactive graphic programs for network and circuit analysis. Mr. Dertauzos contends that although the process of circuit design and analysis is virtually identical when done manually or/and when done with the aid of a computer, the computer approach compresses the analysis process from several hours or days to a matter of seconds or minutes. By entering a schematic or input and receiving tables and/or plots of output the designer interfaces with the system in terms of his own language. As a net result, computer graphics provides the designer with a better job, completed in significantly less time (Reference B33, page 2-1).

B.3 Applications in Computer-Aided Manufacturing (CAM) and Numerical Control (N/C)

Interactive graphic systems have proven to be cost effective in Computer-Aided Manufacturing (CAM) and Numerical Control (N/C) applications The first programs, while cost effective, were limited to profiling (two-dimensional work) and usually replaced the use of the Automatic Programmed Tool (APT) language. Convair is presently working in a cooperative effort with other companies to produce an interactive graphic/APT system that will allow three-dimensional interactive parts programming.

B.3.1 Aerospace experience. - McDonnell-Douglas in St. Louis is using interactive graphic programs in these areas coupled to a common data base consisting of the mathematical description of the surfaces to be manufactured. This integrated approach produces a particularly powerful and versatile tool as beautifully described by Grant Christensen in Reference B34:

> Interactive Graphic Terminals are the basic means through which a complete computer-oriented, design-through-manufacturing system has been developed at the McDonnell Aircraft division of McDonnell Douglas Corp. in St. Louis. This approach already has made designers in some unique instances as much as 48 times more productive than when using conventional, manual methods .... Our graphics system allows the designer to work in a time-cycle environment never before possible. ... Now the designer sits at a CRT in his work area and defines the plane in which he wants the contour. ... To call up data from the files, the designer goes through a series of lists displayed on the tube and indicates with his light pen the name of the drawing, a copy of which he wants displayed on the CRT. ... If a designer with a CRT wants to input the image of a part already manually drawn - either to modify it, or to produce an accurate, full-sized layout for use with a line follower digitizer - he can produce in one hour at the CRT what would take one day at the board. ... He just deletes from his display that which isn't pertinent to his design task. The pertinent data that remains is the identical data originally laid out. He hasn't had to trace it or interpret it. The integrity of the interface is preserved and the structure fits. When layout data as basic as this is called up and used over and over again, productivity ratios multiply by orders of magnitude. ... This system is designer-oriented. A designer at the CRT thinks in the same terms as before and needs no reorientation of his past design experience. ...The designer defines the part accurately, indicates applicable tolerances, but need not note the digital dimensional data, and the computer automatically labels all surfaces and features of the part shown. As a result, by eliminating tedium the system keeps the designer thinking and encourages his creativity. This is unquestionably one of the greatest benefits in terms of increased productivity, although it always will be the

most difficult to measure. ...This approach enables the designer to produce almost as fast as he can think. ...One thing is already clear: the power of the system is limited only by the creativity of the designer using it.

B.3.2 Automotive experience: General Motors, Ford, and Chrysler have all been using interactive graphics for mechanical design, simulation, body geometry and othe manufacturing related processes in varying degree for the last thirteen years (Reference B11). Donald E. Hart describes General Motors' approach as follows (Reference B2):

> Our approach was to provide a new set of tools which would make it possible for designers and draftsmen to use the computer directly to solve their problems. In order to do this certain basic capabilities would be required. These included:
>
> 1) A means for direct graphical interaction between the designer and the computer,
> 2) A set of tools (programs) inside the computer which would be the computer equivalents of the draftsman's T-square, compass, and french curve (3-dimensional french curves for manipulating free-form lines and surfaces), and
> 3) A means of storing the results in the form of "mathematical drawings" inside the computer.
>
> In 1960 we launched a major project to establish a laboratory to learn how to do this.

This approach has resulted (in part) in an interactive graphic system described by Ronald Khal in Reference B30:

> Fisher Body Division of G.M. uses clay models to define the design envelope. Coordinate points from the clay model are recorded and smoothed and refined by a designer at a CRT terminal. This basic data base is then available for designers at CRT terminals to define the working surfaces for forming dies for external surfaces. For interior structures this dimension base defines the geometric constraints for design of interior panels and structural support.

Benefits to the automobile industry have been reduced lead time, improved accuracy and the opportunity to explore more alternatives and thus meet tight yearly schedules, and the ease and speed with which experienced designers and manufacturing engineers who had no knowledge of computers could use the interactive graphics terminals. A side benefit is the fact that the use of the terminal with a light pen, etc. proved to be analogous to the drawing board experience of designers. A specific example cited by D.E. Hart of General Motors was for windshield design (Reference B2):

> For each windshield, styling has been completed in a day that used to take several weeks. The resultant windshield model on the computer was subsequently used by Chevrolet for design of the windshield wiper system. This job previously took thirty days and is now completed in five. ...This lead time reduction is a result of a significant increase in the productivity of scarce design manpower.

Another example of CAM-N/C interactive graphic application is at Battelle Institute as described in Reference B35 by Akgerman and Altan:

> Battelle Columbus Laboratory is designing forging dies from a rough sketch with key points, plane and geometric surfaces identified by number. APT is used to describe the lines, planes and circles for the math model of the forging die which becomes the data base that supplies information for all design, manufacturing and quality control functions within the production system. To use the system a designer sits at a CRT display and calls up the profile from the APT math model along with forging characteristics of different metals, forge shop restrictions, die steel requirements, etc. As he fine tunes details concerning fillets, corner radii, and web and rib ratios he sees an appropriate section develop on the CRT. Appropriate NC tapes are then generated, along with analytical stress verifications and detailed drawings if required.

B.4 Applications in Management

Many firms, including Boeing, Martin, Systems Development Corporation (SDC), insurance companies, brokerage houses, and television broadcasting corporations have installed interactive graphics for management support. Typical functions provided by the system are the display of current status information, scheduling PERT displays, and modeling programs. The interactive display terminals allow the manager to ask "what if" type of questions in regard to altered program workloads, changing prices and interest rates, and other factors involved in operational decisions. Because the computer is instantly available, computations, predictions and extrapolations for future trends from current data can be readily made and depicted graphically. These inquiries and manipulations may be accomplished at the display terminal in a user-oriented language that does not require him to be a computer specialist to use the system.

The Japan Broadcasting Corporation (NHK) has implemented an elaborate Management Information System (MIS) called TOPICS (Total-On-line Program and Information Control System) which uses interactive graphics (Reference B36). All production and broadcasting activity of NHK's two television and three radio networks is coordinated via this system. TOPICS maintains working schedules and budgets and helps administer the simultaneous production of some 1,800 programs by 1,000 directors and 2,700 technicians at work on 26 television and 33 radio stations and on location. About 184 alphanumeric CRTs and eight vector drawing CRT display terminals are in use in the system. Corporate president, Yoshihon Maeda observes (Reference B36):

> NHK is not using the computer simply in individual applications, such as information retrieval, simulation, computation or process control, but for all, simultaneously; and in a completely integrated way. ... NHK has made the computer the technological underpinning of our entire complex of operations. The new system fulfills my ambition to reorganize the corporation in such a way that the mechanics of running the organization would be looked after by machines so that our people could do human work.

Business and engineering management decision making can be expedited and accomplished with more confidence through the use of interactive graphics. Direct cost savings can be achieved as well as the more important significant cost avoidance (Reference B37).

B.5 Summary

Figure B1 succinctly summarizes the wide range of applications of interactive computer graphics within industry, not all of which have been discussed in this Appendix. It is apparent from the figure that industry - particularly the aerospace industry - has made heavy investments in interactive graphics software.

○ CURRENTLY IN USE/DEVELOPMENT IN INDUSTRY ● CURRENTLY IN USE/DEVELOPMENT AT GENERAL DYNAMICS

	AEROSPACE	SHIPBUILDING	AUTOMOTIVE	ELECTRONICS	CIVIL ENGR	BUILDING & CONSTRUCTION	MILITARY
SURFACE DEFINITION	○● LOFTING MASTER DIMENSIONING	○ LINES FAIRING HULL DESIGN	○ BODY STYLING DIE DESIGN	ANTENNA BEAM PATTERNS	○ TOPOGRAPHY CONTOUR MAPPING	○ SHELLS	BLAST EFFECT ENVELOPES
STRUCTURES ANALYSIS	○● FINITE ELEMENT ANALYSIS FRAME ANALYSIS	○ STRUCTURAL DETAILING	○ SUSPENSION SYSTEM DESIGN	ANTENNA STRUCTURES	STRUDEL STRESS	MEMBER SELECTION FRAME DESIGN	○ ROTOR ANALYSIS
CONFIGURATION LAYOUT	○● PACKAGING INTERFERENCE ANALYSIS WT & BALANCE	○ ARRANGEMENTS SPACE RESERVATION, PIPING, CABLING, WIRING, ROUTING		○● PC BOARD LAYOUT IC CHIP DESIGN	ROADS ALIGNMENT SHIPYARD CRANE TRACKS	○ OFFICE LAYOUT PIPING DESIGN	
DRAFTING & N/C	○● PART DESIGN, WIRING, ENGR DWG MAINTENANCE, PART PROG	○ STRUCTURAL DETAILING FLAME CUTTER PROGRAM	○ PART DESIGN TOOL & DIE DESIGN	● BACK PLANE WIRING PC BOARD ASSEMBLY FROM TO LISTS	○ HIGHWAY PROFILE CROSS SECTIONS PLAT LAYOUT		
MECHANISMS	○● LINKAGES FOR DOORS, LANDING GEAR, ETC KINEMATICS	PROPULSION PIPE SIZING PIPE STRESS	○ WINDSHIELD WIPERS DESIGN	ANTENNA VIBRATION RF INTERFERENCE		HEATING, VENTILATION, AIR CONDITIONING SYSTEMS	RECOIL ANALYSIS
PRELIMINARY DESIGN	○● A/C PERFORMANCE AIRCRAFT DESIGN SPACE VEHICLES WING FLAP DESIGN	ISDS COGAP		LOGICAL DESIGN		○ ARCHITECTURAL LAYOUT PROSPECTIVE VIEW GENERATION	WEAPON SYS ANALYSIS
ENGINEERING ANALYSIS	○● CONTROL SYSTEMS ANALYSIS DATA REDUCTION OPTIMIZATION CURVE FITTING TRAJECTORY ANLY AERODYNAMIC ANLY	○ HULL CHARACTERISTICS, HEAT, BALANCE	○ SAFETY FACTOR PLANNING STRESS ANALYSIS	○● CIRCUIT ANLY ECAP, ETC			○ DAMAGE ANALYSIS DETONATOR ANALYSIS STRUCTURAL ANALYSIS
MANAGEMENT	○● DATA BASE INQUIRY FORWARD PRICING PROPOSAL MONITOR- ING EDITING & PREPARATION			LEARNING CURVE ANALYSIS FUNCTIONAL ANAL- YSIS & FLOW DIAGRAMMING PERT CHARTING			
PROGRAMMING	○● ON LINE DEBUGGING ON LINE PROGRAM- MING			TEXT EDITING INSTRUCTION			

Figure B1. Interactive Computer Graphics Applications

B.6 References

B1. Sutherland, I. E.: Sketchpad, a Man-Machine Graphical Communication System. Proceedings of the Spring Joint Computer Conference, Detroit, Michigan, May 21-23, 1963.

B2. Hart, D. E.: Interactive Graphics at G. M. Research. Paper presented at Keydata Corporation Symposium, Boston, Massachusetts, May 1969.

B3. Bonine, K. C.: Design Considerations for Continuous System Simulation Languages. Automatica, vol. 8, 1972, pp. 647-651.

B4. Sedore, S.R.; and Wright, A.I.: SCEPTRE Support II. Air Force Weapons Lab Report AFWL-TR-69-77, September 1972.

B5. O'Neill, J. B.: Computer Graphics as an Operational Tool. Douglas Paper 5672, Computer Graphics 1970 International Symposium, Uxbridge, England, April 1970.

B6. Sturcke, E. H.: Interactive Graphics in Aircraft Landing and Take-Off Studies. Conference on Applications of Continuous System Simulation Languages, San Francisco, California, June 30 and July 1, 1969.

B7. Coale, C. W.; and Forsberg, K. J.: Development of Improved Structural Dynamics Analysis. AFFDL-TR-66-187, vol. 1, October 1966.

B8. Forsberg, K. J.; and Ferriera, S. K.: Development of Improved Structural Dynamic Analysis. AFFDL-TR-66-187, vol. 2, March 1967.

B9. Sayer, R. B.: Computer-Aided Aircraft Structural Design. AIAA Paper No. 68-326, AIAA/ASME Conference, Palm Springs, California, April 1968.

B10. Batdorf, W. J.; Kapur, S. S.; and Sayer, R. B.: The Role of Computer Graphics in the Structural Design Process. AIAA Paper No. 68-1012, AIAA Conference, Philadelphia, Pennsylvania, October 1968.

B 11. Salzman, R.M.: State of the Art in Computer-Aided Design. Paper Presented to Panel No. 3 during Panel Reports for DOD/Industry on Computer-Aided-Design and Computer-Aided-Manufacturing/Numerical Control, Davenport, Iowa, Oct. 1969

B 12. Bonine, K.C.: Economic Considerations of CAD/CAM. Paper presented at SME International Conference on CAD/CAM, Atlanta, Georgia, February 1972.

B13. Cronk, M.: An Interactive Computer Graphics Program for NASTRAN. NASA Technical Memo 2378, NASTRAN Users Colloquium, Langley Research Center, September 13-15, 1971.

B14. Bayer, S. E.: Graphic Control Systems Analysis. Summer Computer Simulation Conference, vol. 1, pp 802-807. June 14-16 1972.

B15. Greiner, H. G.: Influence of Vibration Modes on Control System Stabilization for Space-Shuttle Type Vehicles. NASA CR-112202, November 1972.

B16. Prince, D. M.: Interactive Graphics for Computer-Aided Design. Addison-Wesley Publishing Co., 1971.

B17. Boyles, R. Q.: Application of Computer Graphics in Aircraft Design. Paper presented at International Automotive Engineering Congress, Detroit, Michigan, January 1969.

B18. Bachman, W.: Geometry and Weight Synthesis Adapter for Use in the Combat Aircraft Synthesis Program. Convair Aerospace Division of General Dynamics Report GDC-ERR-1572, December 1970

B19. Reed, T. F.: A Graphics Program for Plotting Weight vs. Center of Gravity (Percent MAC). Convair Aerospace Division of General Dynamics Report GDC-ERR-1532, December 1970.

B20 Reed, T. F.: Interactive Graphics Program for Aircraft Weight Sizing. Convair Aerospace Division of General Dynamics Report GDCA-ERR-1644, December 1971.

B21 Bulinski, R. J.: Combat Aircraft Synthesis. Convair Aerospace Division of General Dynamics Report GDC-ERR-1503, December 1970.

B22 Anon.: Aircraft Design Synthesis. Air Force Flight Dynamics Lab (AFFDL) Contract F33615-72-C-1925.

B23 Anon.: Aircraft Design Synthesis. Convair Aerospace Division of General Dynamics Report GDCA CM72-2196, April 1972.

B24 Evans, D. C.: Graphical Man/Machine Communications. Technical Report No. RADC-TR-68-90, Rome Air Development Center, May 1968.

B25 Whitney, C. A.: Wind Tunnel Data Analysis Using Interactive Graphics. Convair Aerospace Division of General Dynamics Report TN-72-DP-05, October 1972.

B26. Anon.: Computer Graphics Enables Tri-Star to Meet Flight Schedule. Product Engineering, May 1972, pp. 13-14.

B27. Anon.: Graphic Computer Terminals Whet Designers' Effectiveness. Product Engineering, February 1971, pp. 18-19.

B28. Haas, M. E.; and Van Ingen, J. L.: Computer Graphics Techniques Applied to Airfoil Design. Software Age, September 1967.

B29. Karemaa, A.: Airplane High-Lift System Design by Interactive Graphics. AIAA Integrated System Conference, Palo Alto, California, February 17-18, 1971.

B30. Khal, R.: The Inevitable Marriage of CAD and CAM. Machine Design, June 1972.

B31. Ruggio, S.; McBride, R.; Perrill, W.; and Tyler, K.: Computer Aided Packaging of Printed Circuit Board and Electrical Engineering. Electro Dynamics Division of General Dynamics Report GDPO-ERR-71-057-1, 1971.

B32. Meadows, D. M.; Robertson, J. E.; and Kragh, E. C.: Computer-Aided Hybrid Microcircuit Mask Design. 1968 International Electronic Circuit Packaging Symposium, vol. 9, section 314.

B33. Anon.: Electrical Network Analysis. N68-16206, Computer Graphics: vol. 1, NASA-MSFC Digital Computer Graphics Program, December 1967.

B34. Christensen, G. D.: A Design Revolution with Graphic Terminals. Optical Spectra, June 1972.

B35. Akgerman, N.; and Altan, T.: NC's Growing Impact on Design. Machine Design, July 1972.

B36. Anon.: A New Wrinkle in Broadcasting. Data Processor (IBM), vol. 12, no. 3, July 1969.

B37. Miller, I. M.: Economic Art Speeds Business Decision Making. Computer Decisions, July 1972, pp 18-21.

APPENDIX C
REPORT TO SPARC FROM AD HOC COMMITTEE ON OPERATING SYSTEM CONTROL LANGUAGE

APPENDIX C

REPORT TO SPARC FROM AD HOC COMMITTEE ON OPERATING SYSTEM CONTROL LANGUAGE

This appendix presents the first five sections of the draft version of Reference C1. It is presented here as a convenience to the reader because succinctly presents the history of ANSC X3/SPARC/OSCL, their investigations and conclusions.

C.1 Introduction

C.1.1 History. - At the USASI X3.4.2 (Committee on Programming Languages) meeting of 1967 June 27-28, M. H. Perstein presented a proposal that "X3.4.2 shall initiate action to provide an industry-standard input language for top-level control of the computer process, ...". The proposal was presented following a suggestion developed by an internal technical committee at System Development Corporation, Mr. Perstein's employer.

The proposal did not come to a vote at that meeting, but the chairman of X3.4.2, P. Z. Ingerman, appointed Perstein a committee of one to investigate community interest in the matter and to report back to X3.4.2. In addition to direct mailings to persons who might be interested, Perstein set a news release to the industry and professional press. Because of the inept paraphrasing of the release by one news medium, the propriety of the wide distribution and the use of a news release were questioned at the next meeting of X3.4.2, despite the specific purpose of determining community interest imposed on the Perstein committee of one.

At the next meeting of X3.4.2, 1967 August 22-23, Perstein reported that four favorable responses had been received. At the meeting on 1967 November 1-2, he reported that there was some interest in evidence, but not enough to form a committee. The question remained open.

At the meeting on 1968 January 22-25, Perstein reported a continuing display of interest and requested approval of a news release incorporating a strong letter of support from N. J. Ream, Special Assistant to the Secretary of the Navy. The release was approved by C. A. Phillips, Chairman of X3. Following this meeting, Perstein issued the news release and received many assurances of support in response.

At its meeting of 1968 June 18-19, following some controversy on procedural and jurisdictional matters, X3.4.2 voted to form an ad hoc committee on standard operating system control language (X3.4.2F).

The scope and program of work for X3.4.2F were prepared at the 1968 August 14-15 meeting of X3.4.2 and recommended for approval by X3.4. Perstein noted that he had four volunteers for X3.4.2F, when, as, and if it be formed.

X3.4 approved the scope and program of work of X3.4.2F with a slight change. At the X3.4.2 meeting of 1968 October 21-23, Perstein reported ten volunteers, in addition to himself, were willing to work on X3.4.2F. Ingerman appointed Perstein Chairman of X3.4.2F.

The first meeting of X3.4.2F was held 1969 February 4-5 with 35 people in attendance. Much of this meeting was in the nature of a seminar on various aspects of the need for standards in this area. Work began on the delineation of the functions and the users of an operating system control language. This was reported to X3.4.2 at its meeting 1969 March 5-6.

Since then, USASI has become ANSI, X3 has been reorganized, with X3.4.2F reporting to SPARC, and its name changed to OSCL. Although the reorganization did not dissolve X3.4.2, it has not met since 1969 March 6.

The further history of OSCL (nee X3.4.2F) is contained in the minutes of meetings, attached to and made a part of [the complete] report [viz, Reference C1].

C.1.2 Semantics. -

C.1.2.1 Operating System Control Language (OSCL): This is the language for controlling the operating system. There must exist, in order for the term to be meaningful, an operating system to be controlled by OSCL or some sort of artificial language. Such system exhibit a very broad range over the spectrum of sophistication. Somewhere in the middle part of the range there must lie a set of OSCL functions of broad general interest which is narrow enough to be meaningful but broad enough to be useful. If the user approaches a bare machine, one endowed with electricity but devoid of software, he must control the operation of the system by manipulating console switches and loading and unloading peripheral equipment in the proper sequence as well as providing a program to control all aspects of the desired computation. This is probably the low end of the sophistication range. At the other end is the elaborate hardware and software system which approaches the capacity of humans. This is a system in which the manipulation of the system may be very loosely defined and in fact may be arrived at by a conversation between the user and the system to agree on the level of control. Here at the high end, the control language may devolve to the vernacular and may approach natural language in freedom and complexity.

In another sense the spectrum of operating systems control languages covers a range of systems which may be categorized as small to large in size. This range of systems is orthogonal to the sophistication range and the combination of the two ranges leads to a set of systems which vary in at least two dimensions. It is clear that the OSCL requirements of the very small sized systems are probably quite different from the very large sized systems.

It is not clear just where, over these ranges, are the boundaries, within which the proper sphere of control language is to be drawn. Nevertheless, OSCL is the language for controlling a computation system which includes, for the purpose of such control, software characterized by some point in the vast range of sophistication and designed to interpret and respond to such language.

Several aspects of such a language may be candidates for standardization. Such standardization cannot be contemplated until the semantics of control of computation systems can be categorized and defined. It may be that further study will be able to decide what parts of the responses to a control language may also be standardized.

In order to approach this area intelligently certain hypotheses may be posed. These hypotheses, described below in C.1.2.2, C.1.2.3, and C.1.2.4, form the basis which the committee feels must exist if a standard is to be developed. Unless these hypotheses can be clearly validated, standardization cannot exist.

C.1.2.2 Elementary functions: There is a feeling in the committee that there exists a set of units of work or of response which represent the repertoire of the computation system. It is not proposed that these elementary functions be quantized or indivisible, but rather these are hypothesized for convenience only.

It is assumed that there is a three step process involved in breaking down the universe of control function into elementary functions. The test of this hypothesis is that the process can actually take place. The steps envisioned are:

1. The universe of function can be broken down into specific functions done in response to a single OSCL command.
2. That there are some responses which occur for more than one command. All the set intersections are considered as candidates for a single elementary function each. All the set differences are equally candidates.
3. For purely esthetic reasons these sets may be further subdivided. In this case the separability of the sets will have to be maintained.

Each of the sets arrived at at the end of step 3 is defined to be an elementary function. It is a further requirement that the set of elementary functions shall be bounded and in fact the total number should be relatively small.

That this can be done for an existing system seems obvious on the surface. That it can be done for all existing systems (or even a fair number of these) is not at all clear. The real test is whether this operation can be extrapolated from existing systems to satisfy the needs of a wide range of users of computation systems.

C.1.2.3 Functional group: In the context of operating systems it is possible to hypothesize the existence of a related group of elementary functions to be invoked when the user issues a single command. The proof of this hypothesis must rest on the ability to be able to transform non-command type control functions such as keys, buttons and other physical action devices into the same context as more usual commands.

It is further assumed that it is possible to structure the semantics of commands into the familiar action symbol (sometimes called a verb) and a set of contextual modifiers (usually called operands).

C.1.2.4 Structure of commands: This topic refers to the syntax of commands. The hypothesis to be tested here is that when the set of functional groups is known semantically, there can be developed a syntax to incorporate all of them. This includes all of the usual periphery of such syntaxes such as command words, parameters, internal delimiters, terminators and character sets.

If the hypothesis C.1.2.3 is proved correct, then this hypothesis can be tested by producing at least one potential structure which satisfies the hypothesis.

C.1.3 Conclusions. - We have concluded that there is a need for a standard. We have also concluded that there is a clear need for further work on OSCL. We propose that a formal study committee be formed.

It is the strong feeling of the current committee that a standard is much needed and that work should proceed with deliberate speed to do the necessary groundwork. After the study work has been completed, it is expected that there will be a recommendation for one or more standards in this area.

There are a large number of potential OSCL candidates. In fact the current committee did not have sufficient resources to complete exhaustive surveys of OSCLs. None of the OSCLs surveyed so far is considered sufficient as a basis for a standard, and it may be that a more complete survey will not alter this conclusion.

C.1.4 Recommended scope of the study committee. - The committee should be charged with codifying and organizing the existing information about operating systems control language with the goal of recommending specific standardization activity. Based on the work of the last two years, the study committee must extend into those areas not yet considered. In particular, the requirements and peculiarities of networks of operating systems must be fully exposed. The interactions of Data Descriptive Languages and Data Base Control Languages with OSCLs must be examined. The relationships of privileged users to the integrity and reliability of the system may well have implications with respect to the OSCL and should be studied. The interactions of hardware and OSCLs must be considered.

C.1.5 Recommended program of work of the study committee. - Consider the elementary functions that operating systems might perform. Consider those functions illuminated by existing surveys and as many others as are deemed necessary to truly represent all functions that a system might perform for users. Since many alternative sets of elementary functions will be considered, it will be necessary to evaluate different sets to find the most appropriate set.

Is it possible to subset the collection of elementary functions and treat these subsets as whole sub-languages of OSCL? Is there some set of nucleus functions which are mandatory in any OSCL?

Define a way to describe these elementary functions with a minimum of ambiguity and inconsistency. In particular, is there a way of describing these functions such that various systems can successfully offer the same functions?

In particular, examine the relationship of the elementary functions of a Data Base Control Language to the OSCL elementary functions. Also examine the report of the ANSI DDL Study Committee as it relates to OSCL.

Complete the study of the implications of hardware characteristics on control languages. It is apparent that every system in existence has external differences. Are these differences purely cosmetic or does the fundamental underlying system have a real impact on the form of OSCL associated with a system?

Define a method of measuring conformance with the eventual OSCL standard. Such a measure is probably required in the future so that there can be no question as to the requirements that conforming systems must meet.

Define and describe the impact that the external media (typewriters, scopes, etc.) may have on the forms of an OSCL. Consider the necessity or desirability of device dependent language forms.

Is it possible to divorce certain system responses from the OSCL or should all responses be categorized and regularized? Definition in this area is much needed.

In light of the above studies can the following hypotheses be proved or disproved'

1. There exists a set of elementary functions.
2. There exists a reasonable set of user-oriented groupings of these functions.
3. There exists a reasonable user-oriented syntax for commands to invoke these groups.
4. There exists a reasonable set of commands following this syntax.

When all of this study has been accomplished with sufficient thoroughness, is it possible to recommend that one or more standards in the OSCL domain should be promulgated and specifically in which areas?

Determine the impact on users of having to convert to the eventual standard OSCL. This should be estimated for various size systems from a very small to a very large and for a wide variety of applications areas.

C.2 Language/Function Survey

C.2.1 <u>What was surveyed (historical)</u>. - In an attempt to satisfy point 1 of the program of work and to determine if an existing control language might be suitable for a standard, the committee developed surveys of the following operating systems and their control languages:

1. GECOS III
2. EXEC VIII
3. Honeywell Mod 4 OS
4. MULTICS
5. PDP - 10 Monitor
6. 360 OS (TSO)
7. 360/DOS
8. 360/TSS
9. Sigma 5/7 BTM

These surveys may be found in their entirety in Section 6.4 [of Reference C1].

The control languages of these particular operating systems were chosen as candidates for the survey because it was felt they presented a reasonable sampling of existing systems and were familiar to the surveyors. In addition the majority suppo both the batch and interactive modes of control in both uniprogramming and multiprogramming environments.

C.2.2 <u>How surveyed.</u> - An attempt was made to standardize the method of surveying each operating system by having the surveys follow a common outline. General information was provided for the following sections:

1. User/System Interface
2. The Internal Components of the System
3. Initial State of the System
4. Syntax of the Control Language
5. Normal Behavior of the System
6. Interruption of Normal Behavior
7. Privileged Use
8. System Functional Diagram (Beech Tree)

The selection entitled "Normal Behavior of the System" contains the lexical structure of the command language. Subsections are provided for:

1. Process Management
2. Program Management
3. File Management
4. Control Language Modification
5. Enquiries

Each subsection contains the pertinent individual commands. Although in most cases no attempt was made to specifically identify associated parametric data, the general functions performed by each command were specified as well as their modes of operation (interactive, non-interactive or both).

In an attempt to provide accuracy and comprehensiveness, each survey was, where practical, prepared by an individual who was familiar with the system being surveyed either through vendor or user association. In general the surveys are a summary of information contained in manuals or other customer documentation.

Although the surveys are reasonably comprehensive in scope and content, each is an independent document. It was immediately obvious to the committee that some means of crossreferencing the data contained in them was necessary. After several methods had been investigated a systems and functions matrix was devised which correlated a composite list of functions against the functions provided by eight of the operating systems surveyed. The resultant matrix revealed some rather interesting information. This will be covered in Sections C.2.4 and C.2.5.

C.2.3 <u>What was not surveyed and why</u>. - There were two limiting factors governing the breadth of the systems surveyed. First it was necessary that there be someone who had the time and energy to make the survey. Second the committee felt that the larger systems would provide the most information about OSCL.

As a result only existing systems that were well documented, reasonably large, and known to the surveyors were chosen. Of course the surveys were not exhaustive in this area as a number of manufacturers' OSCLs were not included. Network systems, sensor based systems and elaborate multiprocessing systems were also omitted

C.2.4 Commonality of functions. - Prior to applying the data contained in the survey to the systems and functions matrix, a complete function list was produced. This function list was subject to several iterations during the earlier committee meetings and the list used for the matrix was somewhat abbreviated from earlier versions.

After all of the surveys were applied to the matrix the following points were noted:

1. There was a good deal of commonality of functions among the systems surveyed especially in regard to commands relating to program and file management.

2. There were a number of functions contained in the systems surveyed which were not reflected in the functions list. Apparently earlier versions of the functions list would have been more suitable for the matrix.

C.2.5 Diversity of languages. - Regardless of the commonality of functions demonstrated by the matrix, there was considerable diversity in the language implementation methods of the functions within operating systems both syntactically and lexically. In general the surveys and matrix pointed out that there are a finite number of functions for which an operating system can provide control regardless of its complexity. However, within the industry today there is an almost infinite variety of ways of combining and implementing these functions from a language standpoint.

C.3 Types of Users of Command Languages

C.3.1 Non-program oriented users. -

C.3.1.1 Primitive users: In this category are grouped all those users of systems who do not have any control over the system. In this sense, they are zero function users or non-users. Nevertheless, they represent by far the largest body of system users. These are the kind exemplified by the watcher of the ticker tape display. He is aware of the system and is cognizant of its output from minute to minute, but does not have any control over what is contained in the display.

Another of this type is the race track goer who watches the Tote Board. Here he will react to the displayed information which is dynamic but will not be able to directly affect the values shown.

C.3.1.2 Circumscribed users: In this category are included all those, who although they make inputs to the system and receive outputs, in no way affect the progress of the system. As an example of this category, consider the airlines reservation clerk. The clerk inputs data, receives answers from the system, but nothing the clerk can do will cause the system to alter its processes. In cases like this the users are reacting with the data contained in and controlled by the system rather than the system itself.

Another instance of this category is the payroll clerk. In many cases he feeds the data bank of the system and receives in return summaries and listings showing the current state of his payroll. In no way can he directly affect the operation of the system running the payroll packages, and even may not be allowed to alter the progress of the payroll programs themselves.

A last example of this kind of user is the desk calculator. Here the control that the user exerts is over the application package providing the desk calculator facility and not the operating system. This user as with others in this category is isolated from the operating system which exists behind his application. In fact this type of user is unconcerned with the operating system and its controls and will not be able to tell when the character of the operating system changes, so long as his package of applications still runs.

C.3.2 Program oriented users. - Under this broad category are included all those who are aware of the operating system. These are the users who have some degree of control over the operating system. They may be distinguished from the preceding categories in that they are somehow program oriented. They think in terms of programming and programming systems, and are to some degree aware of the underlying hardware. This general group has some well defined sub-categories.

C.3.2.1 General use: This class of user is characterized by the fact that he does a wide variety of operations on the operating systems. He uses several of the programming facilities including the high level languages to achieve solution to his problems. He may also use application packages for specific purposes. In general, however, he may be categorized as applying the resources of the operating system to solve his problems and arrive at answers to his questions.

C.3.2.2 Application programmer: This category of user may use more than one application package as well as almost all high level languages to construct new programming sub-systems to be used by circumscribed users. He makes free use of the control language of the system to create these packages. The only real distinction between this category and the general use category is that this user is an indirect one. This user is not writing programs to solve his own problems but to construct the tools for the solution of someone else's problems. It is expected that this category of user may be more sophisticated than those in the general category. It may be the case that certain parts of the control of the system is permitted to him and not to the general users

C.3.2.3 Operations use: This is a broad category of user engaged primarily in keeping the operating system running for others. Primary among these is the familiar system operator. The traditional concept of the system operator is being replaced in part by a whole range of specialists. Among these are the systems administrator, the Data Base Manager, the Security Officer, and others. This category is characterized by the fact that these users are not solving problems by using the operating system. They do not tend to use high level languages to any great extent. They do tend to modify the system to provide better or more reliable service. They also tend to be the principal people engaged with the physical handling necessary around the outsides of the system. As a general rule, many of the functions performed by this category of users is quite distinct from other program oriented categories. Yet frequently these users tend to 'wear two hats' and only perform these operations functions part of the time. At other times they are quite likely to be general users.

C.3.2.4 Systems programming: Although in many senses this category is a logical extension of the applications programmer, it is separated in this case because there is a unique element to this category. These are the users who are expected to modify the operating system itself. Included in this category is the classical systems programmer who constructs operating systems or parts of them. Also included is the system maintenance user, the trouble shooter, the system debugger or fixer. These users, although they do everything that an applications programmer might do, have additional power to affect the system at a more profound level. The access these users have to the insides of the operating systems makes them at once both very dangerous to the reliability of the system and at the same time essential to its continued existence.

Another reason for separating these users from all others is that they must to some degree be dependent on the internal details of the operating system and its underlying hardware. In this, they can not be entirely system independent. This category of user is probably so close to the system that their functions are shaped by the system they deal with.

C.4 Hardware Considerations

C.4.1 <u>Hardware studies.</u> - Although considerable discussion of the hardware implications of OSCL was undertaken, very little concrete was realized. The committee did undertake to recommend a 'break-in key' standard and the balloting and final text of that recommendation is included in Section 6 [of Reference C1] .

C.4.2 What remains to be studied in this area. - It was the opinion of the committee that further work in investigating the impact of new hardware upon OSCL as well as the impact of OSCL upon existing hardware must be undertaken. The committee did not have the time nor the expertise to undertake such study in depth. It is not clear that such a study can ever be completed. New hardware devices will continue to appear and each of these may have an impact on OSCL.

One of the areas of particular concern is new media as input for OSCL. In the past most commands have been entered by card readers and typewriters. It is certain that new media such as displays and audio input devices will have some effect on command structure.

C.5 Recommendations

C.5.1 Need for a standard. - Although many different programming languages exist for the legitimate purpose of writing computer solutions to problems in a great diversity of application areas, it is not reasonable to expect that a large number of control languages need exist. The purpose of OSCL is single, to control the operations of the system. The advantages of being able to move from system to system without having to relearn control language is akin to the ability to use COBOL on any system without having to learn a completely new language.

The need for a standard is pressing and its attainment should be possible. In order to get to the standard, work should proceed with the necessary preliminaries.

C.5.2 Scope of operating system control functions. - It is the feeling of the committee that there are several capabilities which are not necessarily germane to the executive control of a computer system. We recommend that the study group should examine but not feel constrained to include the following features as part of OSCL:

1. editing of documents, programs, or OSCL procedures,
2. formatting of output data or control of output devices, and
3. data base creation, manipulation, or interrogation.

C.5.3 No candidate. - There appears to be no candidate among surveyed OSCLs to be selected as the basis for the standard. All the languages have good and bad features. Therefore, we are not recommending a candidate language.

C.5.4 No multiple standards. - We urgently recommend a single standard for OSCL. Any user once having mastered the necessary knowledge of OSCL should not be required to learn another language.

As far as can be determined, the only possible excuse for multiple standards might occur if the differences in user media (i.e., decks of cards vs. scopes) might make it possible to achieve a single standard.

C.5.5 Piecemeal standards. - To minimize problems of conversion and transition, every attempt should be made to avoid piecemeal standardization.

C.6 References

C-1. Perstein, M. H.: Report to SPARC from AD Hoc Committee on Operating System Control Language. Report X3/SPARC/OSCL 71-4, American National Standards Institute (ANSI), July 6, 1971.

APPENDIX D
AMERICAN NATIONAL STANDARDS
INSTITUTE (ANSI)

APPENDIX D

AMERICAN NATIONAL STANDARDS INSTITUTE (ANSI)

The following text is copied from the handbook "The World of EDP Standards, written by Marjorie F. Hill (Control Data Corporation Technical Memo TM 4, September 1972). It is presented as a convenience to the reader with the kind permission of CDC.

D.1 History

ANSI was originally organized in 1918 by five engineering societies and was known as the American Engineering Standards Committee (AESC). The founding organizations were the American Institute of Electrical Engineers, American Society of Mechanical Engineers, American Society of Civil Engineers, American Society of Mining and Metallurgical Engineers and American Society for Testing and Materials.

The AESC's initial purpose was to provide a means for coordinating the standards issued by its founders. Three Federal Government departments were invited to join and work with the founding societies, the War Department, the Navy Department, and the Department of Commerce accepted the invitation. In 1920, trade associations as well as several technical and professional societies were invited to join. The need for a more workable structure resulted in the organization of the American Standards Association (ASA) in 1928.

In 1966 the name of the organization was changed to the United States of America Standards Institute (USASI). In 1969, the present name, American National Standards Institute, was adopted. At that time ANSI was reorganized and more recently has undergone several modifications to its structure. The effect is to broaden the membership base and encourage user involvement.

D.2 Objectives

Five of the major purposes of the American National Standards Institute are:

To serve as the national coordinating institute for the development of national standards so as to insure the development of needed standards.

To provide the mechanism for approval and promulgation of voluntary national standards.

3. To provide a focal point for industry and government coordination in the field of standardization.
4. To provide the mechanism for developing programs of ANSI Standards.
5. To represent the USA in international standardization organizations of a non-governmental nature.

D.3 Membership

The American National Standards Institute is a federation of trade and professional associations, companies, and government departments and agencies.

Membership is divided into six classes: organizational member, governmental member, company member, individual members, sustaining member and honorary member. Each organizational, governmental and company member has a vote on Institute matters; sustaining and honorary members do not. An individual member attending meetings of the Consumer Council may have a vote at such meetings.

An organizational member is a non-profit technical, professional, scientific, trade or other membership organization which is national in scope and which is organized so that it can participate in the development of standards.

A governmental member may be a department or agency of the United States Government, or of any of the states, or an interstate or regional authority or agency, or any subdivision of these.

A company member is an individual company engaged in industrial or commercial enterprise or professional educational, research, testing or trade activities. Any affiliate, or division of a company may, at the discretion of the Board of Directors, be eligible for membership as a company member.

Sustaining members are those individuals or organizations not otherwise eligible for membership but which are interested in standards development.

An individual member is a person interested in development of standards or certification. An individual member may attend meetings of any of the councils or boards with the permission of its chairman.

Honorary memberships are conferred upon individuals by action of the Board of Directors.

D.4 Organization

The principal officers are the President, who serves as the chairman of the Board of Directors and three Vice Presidents. The Board of Directors is the governing body of ANSI. It may delegate any part of its authority for the conduct of ANSI's business.

The Board of Directors designates a Managing Director who serves as Secretary of the Institute and is its chief administrative officer.

Membership is composed of not less than fifteen and not more than sixty members. Members include the President, the immediate past President and the three Vice Presidents. Ex-officio members are the Director of the National Bureau of Standards, the Chairman of the Board of Standards Review, the Chairman of each Council, the Chairman of the Certification Committee and the Chairman of the Government Liaison and Support Committee.

Also serving on the Board are twelve directors nominated by company members; twelve nominated by organizational members; twelve representing governmental and consumer interests, nine of whom are nominated by the Government Liaison and Support Committee and three of whom are nominated by the Consumer Council; and three directors-at-large nominated by the President and approved by the Board.

Committees of the Board include the Executive Committee which is empowered to act for the Board between meetings of the Board, and the Government Liaison and Support Committee. The latter committee develop policies and programs to improve liaison with governmental agencies at the Federal, state and local level. The membership consists of from seven to ten members appointed by the President with the approval of the Board of Directors.

Five Councils, the Board of Standards Review and the Certification Committee make up the operating arms of ANSI (Figure D-1). The Councils are: Executive Standards Council, International Standards Council, Organizational Member Council, Company Member Council and Consumer Council.

D.4.1 Board of Standards Review. - Approval of ANSI standards is delegated to the Board of Standards Review by the Board of Directors. Essentially the Board's function is a judicial one of determining whether or not a consensus has been reached. If a concensus has been reached by accepted procedures, the board formally approves the standard.

The Board consists of nine to eighteen members appointed by the president of the Institute in consultation with the chairmen of the Councils, and with the approval of ANSI's Board of Directors. Members of the Board of Standards Review serve as individuals, and not as members or representatives of any organization.

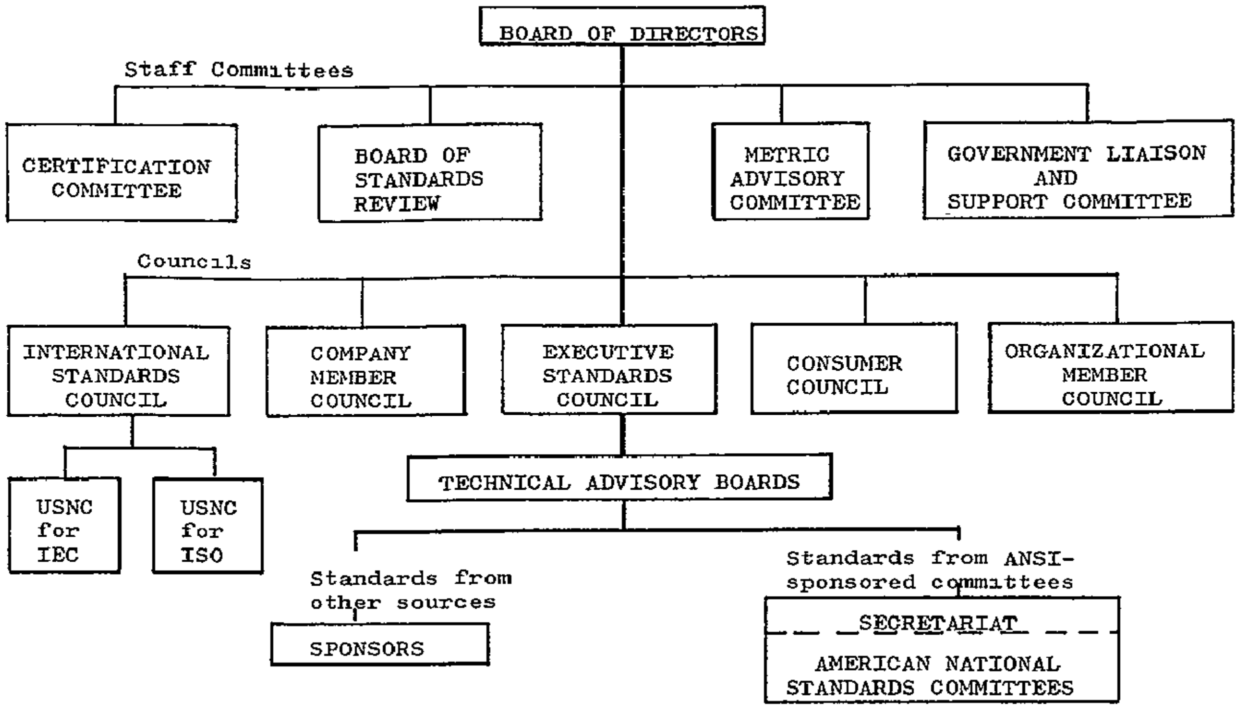


Figure D1. Organization of the American National Standards Institute

The primary responsibilities of the Board are to:

1. Implement procedures for the approval and withdrawal of American National Standards and adjudicate questions or conflicts that develop in the approval procedure.
2. Determine that all substantially concerned parties have had an opportunity to express their views, and have had them carefully considered.
3. Ensure there is evidence of use of or potential use of a proposed American National Standard.
4. Guarantee that before final approval, any recognized significant conflict with another American National Standard has been resolved.
5. Validate that the proposed standard is in accord with the public interest.
6. Scrutinize evidence of the technical quality of the proposed American National Standard.

D.4.2 Certification Committee. - The Certification Committee is responsible to the Board of Directors for the administration of all national activities in the field of certification. The membership of not less than six nor more than fifteen is appointed by the President with the approval of the Board of Directors.

The primary responsibilities of the Certification Committee are to:

1. Foster the development of certification programs by others in response to a demonstrated need recognized by the Committee or a Council of the Institute.
2. Collaborate with government and private organizations in the development of criteria and systems for accrediting certification programs.
3. Advise the International Standards Council on participation by the United States of America in international certification activities concerned with civilian safety, trade and commerce.

D.4.3 Company Member Council. - The Company Member Council represents the interests of commerce and industry in the activities of the Institute. The Council is responsible to the Board of Directors for obtaining an adequate and representative membership and the needed financial support.

It is also the responsibility of this council to stimulate action through the appropriate channels to initiate standards development activities, and to expedite completion of standards projects.

D.4.4 Consumer Council. - The Consumer Council is responsible for the representation and protection of consumer interests in ANSI's work. The Council is composed of ANSI members who choose to be represented in the Council. In addition five persons experienced in the consumer field are designated by the President with the approval of the Board of Directors. The Council will also accept applications for membership from other sources; these are subject to its approval and that of the Board of Directors.

The Executive Committee of the Consumer Council handles the administrative work of the Council.

The primary responsibilities of the Consumer Council are to:

1. Determine, through studies and surveys, where standardization can generate improvements in consumer goods and services.
2. Serve as the Institute contact between the general public and industry in matters concerning standards affecting the public.
3. Make the consumer aware of the impact on the U.S. economy of well-designed and coordinated voluntary standardization programs, and how these programs serve his interests.

D.4.5 Executive Standards Council. - The Executive Standards Council is concerned with the general administration of standards programs and coordinates the technical aspects of the programs. In addition, the Executive Standards Council has the responsibility for encouraging expeditious development of standards.

More specifically, the ESC has the responsibility to develop and promulgate procedures and criteria for the development and approval of standards as American National Standards. Also, the ESC is responsible for defining the scope of proposed standards projects and assigning administrative responsibility for them.

The Executive Standards Council is composed of six representatives of organizational members, six company members, four governmental, two consumer organization members and three members-at-large. Members are appointed by the president of the Institute with the guidance of the categories represented.

D.4.6 International Standards Council. - The International Standards Council (ISC) is primarily an advisory body which counsels the Board of Directors on ANSI membership in international standardization organizations and on ANSI budget requirements for international standardization. The ISC recommends basic policies and procedures for ANSI participation in international standardization and certification activities in liaison with the Certification Committee.

The ISC consists of not less than fifteen nor more than thirty members who represent the broad interests of the Institute.

The ISC responsibility includes technical and administrative policy for the Institute's activities involving the International Electrotechnical Commission (IEC), International Organization for Standardization (ISO), the Pan American Standards Commission (COPANT) and other international standardizations with which the Institute is or may become affiliated.

D.4.7 Organizational Member Council. - The Organizational Member Council collaborates with the Executive Standards Council in identifying the need for new standards and the re-examination of existing standards in the light of changing conditions. The Council also represents the interests of the organizational members to the Board of Directors on the policies and programs of the Institute. Each organizational member is represented on the Council.

D.5 Relation to Other Organizations

ANSI is the United States Member Body of the International Organization for Standardization (ISO) and serves as the focal point for organizing United States participation in international standardization.

For international standardization in the electrical field, ANSI provides services to the United States National Committee of the International Electrotechnical Commission (IEC) and is a member of the Pan American Standards Commission (COPANT).

ANSI holds the Secretariat for several international technical committees, among them, ISO/TC97, Computers and Information Processing, ISO/TC28, Petroleum Products and ISO/TC85, Nuclear Energy.

D.6 Finance

ANSI derives its income from the sale of published standards and from membership dues.

ANSI is responsible for the national dues to ISO, IEC and COPANT. As the secretariat for several ISO and IEC technical committees it contributes additional financial support to international standardization.

D.7 Technical Work

ANSI does not, in itself, develop standards, its only function is to provide the organization through which standards can be approved. It has Technical Advisory Boards to review the technical content, a review board to determine that consensus has been reached, and an approval board (see Section D.4).

To aid in the technical review cycle ANSI has 18 Technical Advisory Boards among which are: construction, electrical and electronics, highway traffic safety, materials and testing, information systems, mining, nuclear, textile, safety and other topics.

D.7.1 Technical Advisory Boards (TABs). - The work of the Information Systems Technical Advisory Board (ISTAB) includes responsibility for standards programs for office machines (X4), computers and information processing (X3), vocabulary for automatic control (C85), and library sciences and documentation (Z39).

The establishment of a TAB may be requested by any responsible party, group or organization, or by a member of the Executive Standards Council. The founding of a TAB is authorized by action of the Executive Standards Council.

The initial chairman and one or two vice chairmen are appointed by the Executive Standards Council. Thereafter they are elected by the TAB membership for one year terms but may not be re-elected for more than six consecutive terms.

The membership of the TABs is not limited in number and all classes of members are eligible. The TAB members serve because of their knowledge of a particular technical field, and their judgments are based on technical issues. The Executive Standards Council may appoint exceptionally qualified individuals to serve on a TAB.

Each TAB is responsible for the following functions according to its assigned scope:

1. Foster, maintain and coordinate standardization projects.
2. Initiate standards activities as required.
3. Harmonize conflicts in standards-making activities and if required, establish an American National Standards Committee (ANSC) to develop a proposed standard. In this case, the TAB may serve as secretariat.
4. Supervise the maintenance of standards.
5. Supervise the establishment, organization, maintenance and activities of ANSCs and review their progress.
6. Advise on participation in international standardization projects.
7. Identify areas in which certification programs are needed.

Each TAB is responsible for submitting an annual report to the Executive Standards Council summarizing the activities under its jurisdiction. The report includes the work of the standards committees as well as an evaluation of the performance of the secretariats.

D.7.2 Technical Committees. - Three of the standards committees for which the Information Systems Technical Advisory Board has responsibility are described in the subsections that follow.

D.7.2.1 Computers and Information Processing (ANSC X3). At an ISO meeting in early 1960, Sweden recommended that a new ISO Technical Committee be formed for standards for information processing. Additionally it was suggested that the United States accept the Secretariat.

Upon return from the ISO meeting the heads of manufacturing concerns in the United States and officials of Business Equipment Manufacturers Association (BEMA) - then the Office Equipment Manufacturers Institute - were invited to a meeting. At this special meeting it was recommended that an organization be formed to develop standards in the computing field. As a result of this recommendation the X3, X4 and X6 Sectional Committees were formed. In 1965 the X6 committee was disbanded and the work taken over by an Electronic Industries Association (EIA) group.

The announcement of the formation of X3 was made in September of 1960 and at that time COBOL and codes were emphasized as the standards to be developed. At the first organizational meeting of X3 which was held in February 1961, seven major topic were identified as.

1. Optical character recognition
2. Magnetic ink character recognition
3. Data transmission
4. Programming languages
5. Terminology
6. Problem definition and analysis
7. Codes.

The activities on keyboards and other office machines standards became a separate committee known as X4 (see Subsection D.7.2.2). BEMA accepted the Secretariat (at that time this function was called a sponsorship) of both committees.

At a Round-Table Conference organized by International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) in May 1961, ISO Technical Committees 95 and 97 (TC 95 and 97) were formed as well as IEC TC 53 ISO TC 95 and 97 were modeled along the lines of X4 and X3 respectively. IEC TC 53 was active for a time, but recently abdicated its work to ISO/TC97/SC3.

The European Computer Manufacturers Association (ECMA) had just been founde and at the 1961 meeting of ISO, was invited to become a liaison member of TC 97. ANSI (at that time ASA) officially accepted the Secretariat for TC 95 and TC 97.

In 1968, X3 was reorganized so that the administrative and technical review responsibilities were absorbed by two standing committees. Further, the technical committees were realigned under the categories of hardware, software and systems.

<u>OBJECTIVES</u> - ANSC X3 operates under the general objective of the American Nationa Standards Institute and is responsible for fulfilling the objectives of ANSI for the domestic standards within its stated scope: standardization related to systems, computers, equipments, devices and media for information processing.

<u>MEMBERSHIP</u> - Membership is drawn approximately equally from three general groups: Procedures, consumers, and general interest.

ANSC X3 members represent their sponsoring organizations in all X3 matters. At the technical levels, members of a technical committee serve as individual professionals.

Membership is divided into three categories:

1. A Regular Member (principal and alternate) is one who seeks membership and demonstrates a valid and continuing interest in the work of ANSC X3. A regular member or his alternate has full voting privileges.

2. Ex-officio members are the chairmen of the technical committees, who may attend and have the right of full participation except voting privileges.

3. Liaison members and observers are classified as those individuals, organizations or other standards committees who have an interest in the work of X3. Liaison members and observers do not have a vote.

ORGANIZATION - The Business Equipment Manufacturers Association (BEMA) is designated by ANSI as the Secretariat for ANSC X3. As the X3 Secretariat, BEMA provides the essential administrative support.

The chairman and the vice chairman are appointed by the Secretariat for a period of three years. The chairman appoints a recording secretary and may appoint such other officers as are required for the conduct of ANSC X3 business.

STANDING COMMITTEES - Assisting X3 in discharging its responsibilities are three standing committees which advise X3 relative to the administration, evaluation, allocation and scheduling of standards projects. Two of the standing committees have staff responsibilities and one is a line executive committee (Figure D2).

The staff responsibilities are vested in the Standards Planning and Requirements Committee (SPARC) and the International Advisory Committee (IAC). The line responsibilities are vested in the Standards Steering Committee (SSC).

INTERNATIONAL ADVISORY COMMITTEE (IAC): The International Advisory Committee (IAC) is responsible for coordinating the work of ANSC X3 with respect to international affairs so as to assist ANSI in executing its responsibilities in its participation in the International Organization for Standardization (ISO), the International Electrotechnical Commission (IEC) and other national, regional, and international standards bodies. The IEC is responsible for policy statements on issues rather than technical positions.

STANDARDS PLANNING AND REQUIREMENTS COMMITTEE (SPARC): SPARC reviews the need for standards, makes recommendations to X3 on the desirability of standards and determines that the work of the technical committee is responsive to the original objectives of the project.

During the time a topic is undergoing scrutiny as a candidate for standardization, a group may be formed to gather facts to justify a standardization project. This group is considered an ad hoc group reporting to SPARC (Figure D3), but is not authorized to develop a proposed standard. Depending upon the recommendation at the conclusion of the study phase, the ad hoc committee is either dissolved or becomes a standards project under the Standards Steering Committee.

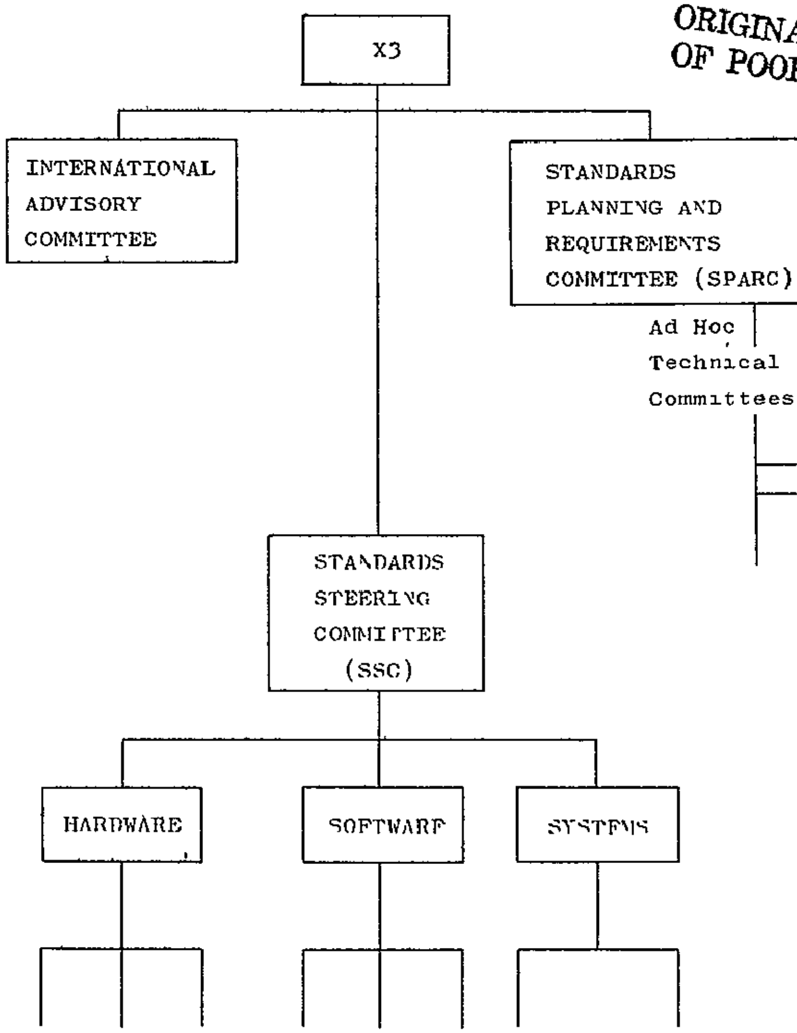


Figure D2. Organization of ANSC X3

SPARC consists of not more than 16 members, including the chairman. Both the chairman and the vice chairman must be from the consumer group and total membership is equally divided between consumers and producers. Included in the membership of SPARC are one representative from the Standards Steering Committee and one from the International Advisory Committee.

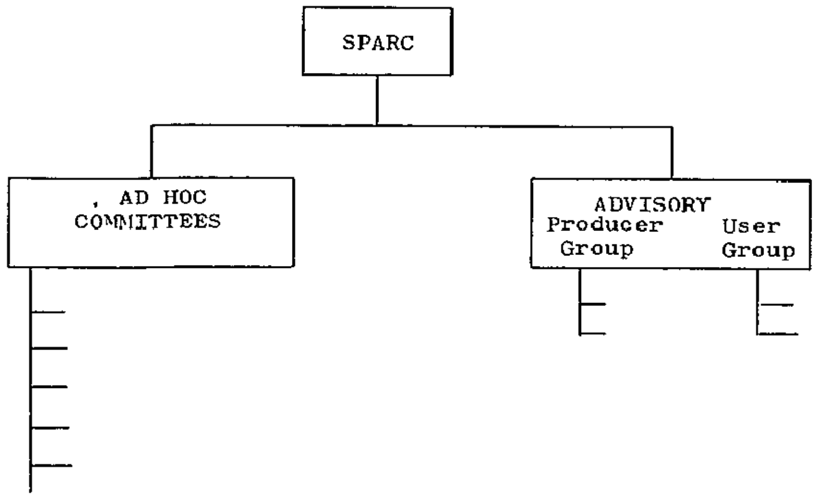


Figure D3. Basic Structure of the Standards Planning and Requirements Committee (SPARC).

Candidates for membership may be proposed by SPARC or any member of ANSC X3, all officers and members are appointed by the chairman of ANSC X3.

To aid in its deliberations, SPARC may convene a producer advisory group or a user advisory group. These groups are considered to be of short duration and are dissolved when the discussions on a single topic are exhausted or the study is complete.

STANDARDS STEERING COMMITTEE (SSC): The Standards Steering Committee is the administrative arm of the X3 organization and as such monitors and coordinates the standards projects which are being developed by the project groups. The SCC consists of a Chairman; Vice Chairman; Secretary; three Group Directors for Hardware, Software and Systems; and eight Section Managers for Technical Committees. All are appointed by the chairman of ANSC X3.

The Hardware Group (Figure D4) has two Section Managers: Recognition and Physical Media. The Recognition Section includes the projects on optical and magnetic ink character recognition. Physical Media projects cover magnetic tape, cards, disks, perforated tape, etc.

The Software Group (figure) has three Section Managers: Languages, Documentation and Data Representation. The Language Section includes development and maintenance projects for COBOL, FORTRAN, ALGOL, APT, and PL/1.

HARDWARE GROUP

- RECOGNITION
  - PHYSICAL MEDIA
    - Magnetic Tape
    - Perf. Paper Tape
    - Punched Cards
    - Magnetic Disks
  - OCR
  - MICR

SOFTWARE GROUP

- LANGUAGES
  - DOCUMENTATION
    - DATA REPRESENTATION
      - Codes
      - Labels
      - Data Representation
    - Documentation
    - Flowcharts
    - Network Oriented Project Management Systems
    - Vocabulary
  - PL/1
  - FORTRAN
  - COBOL
  - ALGOL
  - APT

SYSTEMS GROUP

- DATA COMMUNICATIONS
  - SYSTEM TECHNOLOGY
    - SYSTEM MEASUREMENT
    - I/O Interface
  - Data Communication

Figure D4. Standardization Projects Under the Standards Steering Committee

The Documentation Section handles general topics such as flowcharts, documentation, vocabulary, and network oriented computer systems. Under Data Representation are the projects on codes, labels and data elements.

The Systems Group (figure) consists of three Section Managers: Data Communications, Systems Technology and System Measurement.

RELATION TO OTHER ORGANIZATIONS - ANSC X3 relates to ISO through the American National Standards Institute on matters pertaining to ISO issues and the international aspects of developing standards for computers and information processing. All United States comments on international topics, administrative or technical, are transmitted to ANSI for transmittal to the proper secretariats

X3 reports to ANSI through the Information Systems Technical Advisory Board which acts as a management and advisory body. X3 technical committees maintain liaison with their European Computer Manufacturers Association (ECMA), International Federation for Information Processing (IFIP), International Telegraph and Telephone Consultative Committee (CCITT), and International Organization for Standardization (ISO) counterparts on both a formal and an informal basis with the approval of ANSI and the X3 secretariat.

FINANCE - X3 members pay no dues, but the producer category of member is drawn from the Business Equipment Manufacturers Association (BEMA) Data Processing Group member companies. The cost of supporting X3 is largely borne by the producer group through dues to BEMA.

All technical and administrative committee work is accomplished on a volunteer basis and as such is supported by the organization represented by the volunteer.

TECHNICAL WORK - The technical work of X3 is accomplished under both SPARC and SSC as described previously.

The technical committees under the Standards Steering Committee (SSC) are responsible for the technical development of draft standards on assigned topics identified as desired American National Standards.

TECHNICAL COMMITTEE ORGANIZATION: To provide an orderly method for developing standards, technical committees are organized along the same general lines as X3. The one major difference is that members of a technical committee function as individual experts, while X3 members represent a company or organization.

The chairman of a technical committee is appointed by the Secretariat for a period of two years and is eligible for reappointment.

The vice chairman is appointed by the Secretariat for a term of two years and is eligible for reappointment. His term is to be out of phase with the term of the chairman by one year

Each technical committee may have an international representative who is appointed by the Secretariat for a term of three years. It is his job to be cognizant of the work of corresponding international, regional and foreign standards organizations and report this work to the technical committee. The international representative is a member of the International Advisory Committee.

Membership on technical committees is open to any qualified individual who has an interest in a specific topic.

Each committee member maintains his membership by attendance at meetings and responding to ballots.

TECHNICAL COMMITTEES: The following technical committees have been authorized for the development and maintenance of American National Standards.

1. X3A1 OPTICAL CHARACTER RECOGNITION - Standardization of printed input and output used by optical scanning equipment. The work includes font design, hand-printed characters, print quality, etc.

2. X3A7 MAGNETIC INK CHARACTER RECOGNITION - Responsible for the development and maintenance of MICR standards. Also acts as consultant to the American Banking Association.

3. X3B1 MAGNETIC TAPE - Standardization of the physical and coding characteristics of magnetic tape. Typical projects include revisions to existing standards, development of standards for different recording techniques and standards for tape cassettes.

4. X3B2 PERFORATED TAPE - Develop standards for the physical and coding characteristics of paper tape. Typical projects include paper material standard, tape handling conventions, cores and reels, tapes of high density and edge punched cards.

5. X3B3 PUNCHED CARDS (PHYSICAL) - Standardization of the physical and coding characteristics of unpunched paper cards. Typical topics include: monitoring of existing standards on size and location of rectangular punched holes, special purpose cards, etc.

6. X3B7 INTERCHANGEABLE MAGNETIC DISK MEDIA - Standardization of the physical and magnetic characteristics and control formats of removable magnetic disk media. Specifications included are: magnetic surface characteristics, track width, recording mode, track quality characteristics, etc. Considerations for control formats include: record structure, gap structure, checking techniques, address structure, etc.

7. X3J1 COMPOSITE LANGUAGE DEVELOPMENT - Develop a standard for PL/1 in cooperation with ECMA TC 10 and with technical cognizance of IFIP TC 2.

8. X3J3 FORTRAN - Maintenance, updating, and clarification and interpretation of the American National Standard FORTRAN.

9. X3J4 COBOL - Maintenance, updating, and clarification and interpretation of the American National Standard COBOL.

10. X3J7 APT - Preparation of a draft proposed standard for APT programming language.

11. X3J8 ALGOL - Monitor and participate in the preparation of a draft International Standard on ALGOL.

12. X3K1 PROJECT DOCUMENTATION - Standards for documentation of project functions including introduction of project, definition, design, implementation, operation and evaluation.

13. X3K2 FLOWCHARTS - Develop standards for the techniques of flowcharting and the design and use of flowcharts.

14. X3K5 TERMINOLOGY AND GLOSSARY - Prepare and maintain a Master Working Vocabulary which is cognizant of the definitions required by all the subcommittees of ANSC X3.

15. X3K6 NETWORK ORIENTED PROJECT MANAGEMENT - Formulate and propose standard characteristics and properties of network management systems including fields of network applications such as PERT and CPM.

16. X3L2 CHARACTER CODES - Standardization of coded character sets including code representation, recording formats and format indicators. Projects include: code extension, code registration, control character sets and graphic character sets.

17. X3L5 LABELS - Standardization of labels and file formats for information interchange on magnetic tape, including tape cassettes

18. X3L8 REPRESENTATIONS OF DATA ELEMENTS - Develop standards for representing data elements of common interest such as time, location, organizations and materials.

19. X3S3 DATA COMMUNICATIONS - Define and develop standards for the operational characteristics governing the performance of digital data generating and receiving systems combined with communication systems. Areas of interest include data communication control procedures, formats, transmission speeds and system performance

20. X3T9 I/O INTERFACE - Develop standards for the logical, physical and electrical interface parameters which could be interconnected, such as central data processing equipment, control units and I/O devices. Areas of interest are vested in monitoring the International Organizations for Standardization (ISO) proposals.

AD HOC COMMITTEES: Ad hoc technical committees function as study groups under the Standards Planning and Requirements Committee (see SPARC). Typical of the study groups are: BASIC, operating system control language, display parameters and text processing.

D.7.2.2 Office Machines and Supplies (ANSC X4): When the subject of the need for standards for the data processing industry was raised, it was recognized that business machines would interact with this burgeoning industry. However, it was decided that office machines and supplies should be established as an entity independent of X3. As a result, ANSC X4 was established and became the United States counterpart of ISO/TC 95 (See historical discussion under Subsection D.7 2.1.)

<u>TECHNICAL COMMITTEES</u>-The following technical committees have been authorized under ANSC X4:

1. X4A5 DUPLICATING AND REPRODUCING MACHINES - Develop standards for a safety code for office machines, for the symbolism for machine controls, and for the physical properties of offset plates.
2. X4A6 DICTATING MACHINES - Develop and maintain standards for the functional elements of dictating machines including standardization of terminology and definitions of functions of dictating machines.
3. X4A7 BASIC PAPER LAYOUT - Develop standards for the physical characteristics of business forms, the mechanical characteristics for the placement of characters on a business form and relevant terminology.
4. X4A8 OFFICE MACHINES ELECTRICAL SAFETY - Develop electrical safety standards for office machines
5. X4A9 KEYBOARD ARRANGEMENTS - Develop standards for an arrangement or arrangements for keyboards used in general information interchange. Projects include: an ASCII Keyboard and General Purpose Typewriter Keyboard and the basic arrangement for the alphanumeric section of keyboards operated with two hands.
6. X4A10 MAILING AND ADDRESS MACHINES - Develop standards for mailing and addressing machines.
7. X4A11 CREDIT CARD STANDARDIZATION - Develop standards for credit cards and account numbering systems. Present interest is centered in the maintenance and international standardization of credit cards.
8. X4A15 ALPHANUMERIC MACHINES - Develop standards for key spacing and symbols, electrical requirements and interference levels, paper and ribbon spool dimensions.

D.7.2.3 Library Work, Documentation and Related Publishing Practices (ANSC Z39). The history of American National Standards Committee Z39 dates back to June 1939. At that time the American Standards Association, acting on a request from the American Association of Law Libraries, the Medical Library Association and the Special Libraries Association approved a Committee on Library Standards.

The American Library Association served as the original sponsor of Z39, in 1951, however, the sponsorship was assumed by the Council of National Library Associations.

The first meeting of Z39 was held in New York in March 1940 but progress was hampered by suspension of the work of the International Standards Organization during World War II. Several efforts were made to revitalize the committee but these efforts failed for lack of adequate financial support. Despite the difficulties, however, several standards were published.

It was not until 1961 when adequate financial support became available that Z39 was able to expand both nationally and internationally.

OBJECTIVES- The objectives of ANSC Z39 are best expressed in the scope of its work as:

> Standards for concepts, definitions, terminology, letters and signs, practices, and methods in the fields of library work, in the preparation and utilization of documents, and in those aspects of publishing that affect library methods and use.

MEMBERSHIP- Membership is available to any association or organization concerned with the scope of the project or with interest or activity in library work, documentation or related publishing practices. A total of 46 members includes libraries, professional. technical and educational institutes or associations, abstracting and indexing services, publishers, government agencies and commercial and industrial organizations.

ORGANIZATION- The officers consist of a chairman, vice chairman and secretary who serve for three years.

It is the responsibility of the chairman to give guidance for the formation of subcommittees and, with the advice of subcommittee chairmen, is responsible for inviting persons to serve on subcommittees. The chairman also assumes responsibility for funding activities of the Standards Committee and subcommittees. The vice chairman performs the usual duties of the office and in addition serves as the chairman of the program committee. Subcommittees are formed to fulfill the technical objectives of Z39 and are dissolved upon completion of their tasks.

RELATION TO OTHER ORGANIZATIONS- Z39 relates to the International Organization for Standardization (ISO) through the American National Standards Institute on matters pertaining to ISO issues and projects. Z39 acts as the United States organization responsible for the work of ISO/TC46 Documentation, and also serves as the Secretariat for ISO/TC46 Subcommittee 2, Conversion of Languages.

Z39 works through an ISO/TC46 Working Group to establish the UNESCO-France International Serials Center.

FINANCE -ANSC Z39 has been funded since 1961 by matching grants from the Council on Library Resources and the National Science Foundation.

TECHNICAL WORK - The technical work of Z39 is vested in its subcommittees:

SC/1	PROGRAM
SC/2	MACHINE INPUT RECORDS
SC/3	PERIODICAL TITLE ABBREVIATIONS
SC/4	BIBLIOGRAPHIC REFERENCES
SC/5	TRANSLITERATION

SC/6	ABSTRACTS
SC/8	PROOF CORRECTIONS
SC/9	TERMINOLOGY
SC/10	ARRANGEMENT OF PERIODICALS
SC/13	TRADE CATALOGS AND DIRECTORIES
SC/17	STANDARD BOOK NUMBERS
SC/19	BOOK PUBLISHERS ADVERTISING
SC/20	STANDARD SERIAL CODING
SC/21	TITLE LEAVES OF BOOK
SC/22	LIBRARY MATERIALS PRICE INDEXES
SC/24	REPORT LITERATURE FORMAT
SC/25	THESAURUS RULES AND CONVENTIONS
SC/26	PREPARATION OF SCIENTIFIC PAPERS
SC/27	IDENTIFICATION CODES FOR COUNTRIES
SC/29	PUBLICITY AND PROMOTION
SC/30	IDENTIFICATION CODE FOR LIBRARIES AND BOOK DEALERS
SC/31	MUSIC INDUSTRY CODE
SC/32	TECHNICAL REPORT NUMBERING
SC/33	BIBLIOGRAPHIC ENTRIES FOR MICROFICHE HEADERS AND ROLL MICROFILM CONTAINERS

D.8 Related National Standards Committees

The following committees are under the cognizance of ANSI but with other societies as the Secretariat.

D.8.1 Standards for Drawing and Drafting Practices (ANSI Y14). - Development of standards or recognized practices in engineering drafting and related documentation control systems (excluding architectural drawings and graphic symbols).

D.8.1.1 Secretariat: American Society of Mechanical Engineers (ASME), American Society for Engineering Education (SEE) and Society of Automotive Engineers (SAE).

D.8.2 Graphic Symbols and Designations (ANSI Y32). - Standardization of graphic symbols, reference designations and device-function designations.

D.8.2.1 Secretariat: American Society of Mechanical Engineers (ASME) and the Institute of Electronic and Electrical Engineers (IEEE).

☆ U S GOVERNMENT PRINTING OFFICE 737-503/64 REGION NO 3-11