CR-132393

Publicly Released
February 10, 1978

Part I-Final Report, Tasks 1 and 2

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

Volume III: Support of the Design Process

D6-60181-3

September 21, 1973

(NASA-CR-132393) FEASIBILITY STUDY OF AN INTEGRATED PROGRAM FOR AEROSPACE VEHICLE DESIGN (IPAD). VOLUME 3: SUPPORT OF THE DESIGN PROCESS Final Report (Boeing Commercial Airplane Co., Seattle) 136 p

N78-16013

HC A07/MF A01

Unclas

G3/02 02567

Prepared under Contract No. NAS1-11441 by
Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124

for

Langley Research Center
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

1 Report No	2 Government Accession No	3 Recipient's Catalog No

4 Title and Subtitle
FEASIBILITY STUDY OF AN INTEGRATED PROGRAM FOR AEROSPACE VEHICLE DESIGN (IPAD)
VOLUME III SUPPORT OF THE DESIGN PROCESS

5 Report Date
September 21, 1973

6 Performing Organization Code

7 Author(s)
W. B. Gillette R. E. Bateman
S. D. Hansen J. W. Southall

8 Performing Organization Report No
D6-60181-3

9 Performing Organization Name and Address
Boeing Commercial Airplane Company
P. O. Box 3707
Seattle, Washington 98124

10 Work Unit No

11 Contract or Grant No
NAS1-11441

12 Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D. C. 20546

13 Type of Report and Period Covered
Contractor Report

14 Sponsoring Agency Code

15 Supplementary Notes
Project Manager, Dr. R. E. Fulton, Structures and Dynamics Division,
NASA Langley Research Center, Hampton, Virginia 23365

16 Abstract

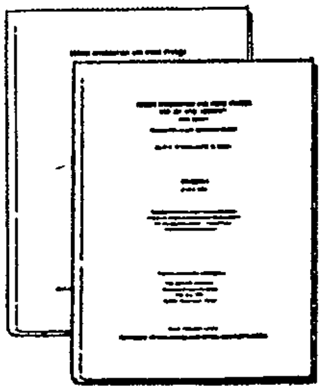
Volume III of the Boeing report on the IPAD feasibility study identifies the user requirements for computer support of the IPAD design process. The User-System interface, language, equipment, and computational requirements are considered.

17 Key Words (Suggested by Author(s))
Computer System
Management Information
Peripheral Equipment
Personal Terminal
User Language

18 Distribution Statement

19 Security Classif (of this report)	20 Security Classif (of this page)	21. No of Pages	22 Price*
Unclassified	Unclassified	136	[redacted]

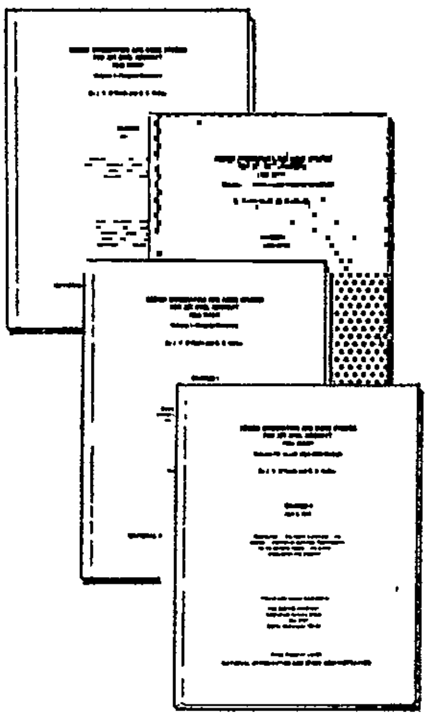
FEASIBILITY STUDY OF AN INTEGRATED PROGRAM FOR AEROSPACE VEHICLE DESIGN (IPAD)



Volume IA
Summary of IPAD Feasibility Study
D6-60181-1A

Volume IB
Concise Review of IPAD Feasibility Study
D6-60181-1B

Part I—Final Report, Tasks 1 and 2



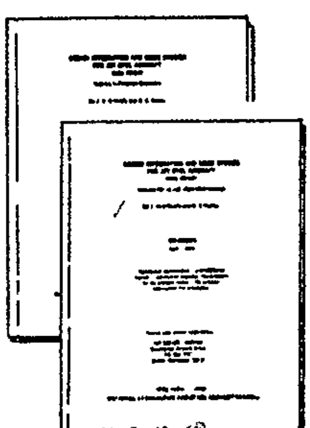
Volume II
The Design Process
D6-60181-2

Volume III
Support of the Design Process
D6-60181-3

Volume IV
IPAD System Design
D6-60181-4

Volume V
Catalog of IPAD Technical Program Elements
D6-60181-5

Part II—Final Report, Tasks 3 through 8



Volume VI
IPAD System Development and Operation
D6-60181-6

Volume VII
IPAD Benefits and Impact
D6-60181-7

SUMMARY

Volume III of the Boeing report on the IPAD feasibility study establishes the user's requirements for computer support of the design process. These requirements were determined during the study of the design process which is presented in Volume II. The user-System interface is described and includes maintaining the data and code bases, preparing to do a job, executing the job and examining the results. A user language is developed as a further means of describing this interface. The user-terminal environment is discussed and suggestions are presented for peripheral equipment and the manner in which the equipment should be arranged. The need for a Management Information System is presented. The required computational capabilities of the host hardware are developed from the design networks of Volume II and the Catalog of Technical Program Elements of Volume V. The answers for two Task 2 questions are included in this Volume.

It is concluded that a computing system which satisfies the user requirements presented in this Volume will provide improved control of complex design problems and improved technical communications. The initial implementation of the system should provide the data base management capability, the ability to execute jobs, continuity of day-to-day activities and information display capability. The long term development of this system should support project planning capability and migration to later generation host computers.

CONTENTS

CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS

TABLES

1.0 INTRODUCTION

This Volume reports the findings of Task 1 of the IPAD feasibility study relative to the user identified requirements for support of the IPAD design process. These support requirements and concepts have been placed in a separate document from the description of the design process to provide a convenient grouping of information.

1.1 OBJECTIVES

The following objectives were established to determine the user requirements for support of the IPAD design process: (1) to identify a user-System interface environment including a preferred language and rules for its use; (2) to identify a user-terminal interface environment; (3) to identify requirements for the peripheral equipment capability; (4) to identify an information base upon which the computational requirements for a host computer system would be determined; and (5) to identify a desired strategy for short term and long term IPAD implementation of the required computing system features.

1.2 BACKGROUND

The principal activity of the technical representatives assigned to this feasibility study was to characterize in detail the design processes for subsonic and supersonic commercial transports. These are reported in Volume II. Two items were measured from the design processes. They are the computational requirements of the host hardware presented in Section 6 of this volume and the status of the technical computer programs denoted Technical Program Elements (TPEs) presented in Volume 5. Other requirements which could not be measured include: user-System interface, Section 3; user-terminal environment, Section 4; and peripheral equipment capabilities, Section 5. These items are difficult to establish and were based on observation, experience and opinion. Section 2 contains the answers for two Task 2 questions which involve support of the design process.

For users of IPAD to function as an interdisciplinary team, it will be necessary to have consistent definition of terms and compatibility of input and output data at the boundaries of technical modules. To initiate the development of an IPAD community discipline, a user oriented command (or function) language has been developed. This language is presented in Section 3 and includes some initial throughts on rules which will be required for its effective use.

Finally, Appendix A contains examples of the IPAD User-System Interaction. These examples were developed to insure compatibility of the IPAD System design with the user needs for maintaining the IPAD data base and for doing work with IPAD.

2.0 ANSWERS TO TASK QUESTIONS

2.1 GENERAL

The answers to Task 2 questions which relate to the user requirements are presented in the following section. The remaining Task 2 questions which relate to the support of the design process are answered in Volume II and those which relate to the System design are answered in Volume IV.

2.2 TASK 2 QUESTIONS (10 and 11)

Task 2, Question 10--What degree of flexibility should be given to the system operator in arranging available OM's into different sequences according to the needs?

Answer--The user should have complete flexibility in arranging the available OM's. The IPAD System should support the user by keeping records of key words, input-output, and internal engineering descriptions to facilitate the selection of the proper OM's. This flexibility will reduce the total number of Technical Program Elements from which OM's will be constructed to do the tasks. Reducing the number should enhance the credibility and integrity of the result because a small number of Elements can be more thoroughly certified than can a large number.

Discussion--The IPAD System will provide sufficient support to allow the user to identify OM's already in the System that will do the proper portions of his job, and to allow the user to construct these OM's into the specific sequence of execution that he desires. If suitable OM's are not available, then the System will provide the user with the same degree of support in recognizing and constructing Technical Program Elements.

The System support will be based on information supplied at the time an Element is entered into the dictionary or at the time an OM is defined. In both cases, a key word list as well as a concise statement of the engineering activities will be provided, and the input-output list will be specified. The key word list will subsequently be used for a library search, and the engineering statement will relate the significant features of the internal calculations and results.

It is intended that users , assisted by the System, will examine the list of available OM's and Elements before entering their own code. The System should allow the user to select a path through a network of OM's and to simulate the solution.

This simulation should show the user which Elements will be executed as well as those data sets which will appear as input, output and those data sets which are internally connected. These System aids of searching key word lists and simulating network solutions will tend to reduce the amount of code in the community library. A smaller number of Elements and OM's is more likely to be thoroughly checked, and will enhance the credibility and integrity of the result.

Lastly, giving the user the ability to rapidly identify and modify OM sequences will tend to reduce problem set up effort and duplication of capability. If the system is very flexible, then OM's will be defined in smaller units with the very thought in mind to subsequent rearrangement.

Task 2, Question 11--What I/O devices will best serve IPAD?

Answer--The most important class of devices will be those required to support the user at his personal terminal. Large volume and special purpose offline devices will complement these personal terminals.

Discussion--The primary user device will be the personal terminal. The largest number will be teletype, either with paper printout or with a text scope. There will be a lesser number of teletypes with a scope capable of vector plotting. These will be interactive thru the keyboard. Finally, there will be a few display-interactive graphics scopes.

The user will be supported at his personal terminal by card readers, tape readers, digitizers and hardcopy printers and plotters.

Much of the input-output volume will be handled offline. In addition to the equipment typically provided as a part of the host installation, there will be a need for automated digitizers, document quality printers, high volume plotters and drafting quality plotters. This equipment is expensive, and the particular choices of equipment will depend on the tasks to be done.

In addition, the users of IPAD will need special devices to fully exercise the capabilities of the IPAD System. For example, a hybrid simulator will be required in Levels V and VI to couple the data contained in IPAD with the analog simulator for pilot evaluation of the airplanes response to control commands. Also the data base in IPAD would contain location information of documentation stored on microfilm, and a remotely accessed microfilm storage device would present reference material to the IPAD user.

3.0 USER-SYSTEM INTERFACE

3.1 OBJECTIVE

The major objective of the user-system interface is to develop an environment which will support the user in the maintenance of both data and code and in the preparation, execution and examination of jobs. The following information was provided as user identified requirements for the IPAD System design. In addition, a user-system interaction was postulated by representatives from the user group and the system design group. This process formed the primary method by which the user requirements were accounted for during the development of the IPAD system design presented in Volume IV. The user-system interaction that was developed as a result of this coordination work is presented in appendix A of this document.

3.1.1 Maintaining the Data Base

IPAD must support the development of a large data base that must provide short and long term storage of data. These data must be readily accessible to the user. Capability to extend the data base must be provided and the size of the data base should not degrade the IPAD system performance. Each IPAD installation will likely have its own data base. There will be corporate data of a historical nature that will be relegated to long term or archival storage. There will be data sets to be used to construct default inputs for general classes of problems. There will also be recently calculated or new information included as data sets in the data base.

It will be desirable for some governmental facility that has an IPAD System to maintain a data bank composed of standard data items, such as standard atmospheres, tables of ephemerides, physical constants, and so on. Other IPAD installations would then access the required standards as needed for current jobs. The central set of standards will help to promote uniformity throughout the total user community. It will also mean that each installation would not have to provide storage capacity for all the standard data items in use.

Each set of data must contain information that will preserve its integrity. The data set must be qualified in terms that relate the owner and the source of the data. The source may be either a reference for experimental and statistical data or a computing program that generated the data. In either case, the qualifier for the data set must contain sufficient information to support the resolution of any questions that may arise in regard to the contents of the data set.

Several types of security are required for the contents of the data base. The first type will be controlled by access codes. These access codes will establish read and write authorization. The next type of security will be imposed by corporate rules and will involve data considered proprietary to the company. Finally, government security classifications will be imposed on certain data.

A process must be provided to control purging data sets from the data base. Unconditionally purge of data sets cannot be allowed if other users have uncompleted activities that depend on that data set. This requirement will protect the integrity of the dependent user's input data. Therefore, the IPAD System should perform a search of tasks in work to find any that depend upon the data set to be purged. This dependence could be contained in the data set qualifiers. If such a dependence is found, the IPAD System should not accept the purge request.

There is clearly a requirement to provide managers or their designated lead personnel the capability to purge obsolete data sets. This capability could be controlled by permission codes, whereby a restricted set of users would be allowed to issue purge commands that the IPAD System would unconditionally obey. The permission codes will be discussed further in section 3.1.6.3.

3.1.2 Maintaining the Code Base

Each IPAD installation will contain a substantial collection of coded programs or program elements. These elements will be combined and executed as jobs.

Each element or set of technical code must contain information in the form of qualifiers that will preserve the integrity of that code. The IPAD System must not allow unrecorded modifications of code to be made. The System must provide an automatic revision of the qualifier so that users of the code will be aware that they are dealing with modified code. Also, the System should identify the person responsible for the revision of the code.

In addition to the qualifier, the System should provide the capability for brief documentation and a set of key words to be related to the code. This will facilitate searches for the proper code by prospective users.

Security provisions similar to that required for data sets will have to be provided for code (see section 3.1.6.3). Access codes will establish the users who have the authority to modify or duplicate a given set of code.

A capability to purge obsolete code is also required and should have the same safeguards provided for data sets.

3.1.3 Preparing to Do a Job

3.1.3.1 Selecting the Job

One of the first activities in the organized process of developing a product will be to establish comprehensive networks of the work to be done. There will be a plan to identify all tasks of the project and this plan will be expanded to include plans for all of the subtasks within each task. The next division of subtasks into jobs acccomplished by individual IPAD users. The plan of jobs within a subtask are the responsibility of the individual user. The plans so developed for the project, tasks and subtasks will be stored as a part of a Management Information System.

The preparation for a job will begin with collection of the proper code. The IPAD System should assist the user to locate a suitable job already in the code base. If a suitable existing job cannot be found, the user must construct the form of his job. In doing so, he will construct logical networks of code, and will insert instructions into the networks that will support his interactively monitoring and controlling the job during execution.

3.1.3.2 Preparing the Code

In most cases, the user will execute an existing collection of code. In these instances, the user will have only to collect the data needed as input. But there will also be cases where the user will have to modify or introduce new code to perform his work.

When the user does not know of an existing set of code to do his job, he should first search through the code available to him in the library. The search will begin on a key word basis. For the items indicated by the key word search, the user will then read the abstracts contained as a part of the information about that item. Further investigation into code that appears correct from the abstract will be done on a manual communication basis or by extracting complete documentation from microfilm storage.

The search for the correct code should be first made among the available jobs. If a job is found that will perform the user's intended activity, then the user would merely have to collect the input data. If no satisfactory job was found, then the search would be among the operational modules (OM) from which jobs are built. A set of OMs may be located that could be collected into a job. If the proper OMs are not available, then the searching must be done among the basic units of code, the coding modules. If the proper coding modules are not found, the user will have to supply new code in order to perform the job.

In a typical case, the user will find an existing job or will modify existing coding modules and OMs. However, in some cases the user will introduce new code. The process of modifying and building new forms of code for the job is best done on a personal terminal, with the user being able to edit the code until it is proper for his job. However, all of the System features of assembling code should be available in the batch job process. Then the coding modules will be developed as a network into large blocks or OMs, which, in turn, will be developed as a network into a job.

The user should be able to specify networks with logical branches. The decision points at these branches may depend on a calculated result. Or, at execution time, the user may supply certain parameters as a part of the execution instructions. Whereas calculated variables may be used to make decisions in the network during execution, there cannot be instructions that will do arithmetic replacements or any other alterations to the calculated values. This limitation is necessary to preserve the integrity of the results, by having them generated only by code that itself has integrity. It will be necessary to have all calculation and generation of values for variables to be done within the coding modules, which will be qualified.

The user will need the capability to insert instructions into the networks that will support his interactively monitoring and controlling the execution process. One class of instruction will be to monitor and display only. A higher level will be to monitor, display and wait for instructions. The user should have the ability to interrupt the execution and be able to preserve the results to that point. These commands may be inserted into the coding modules or into the actual networks.

As a final step in building a new network, or as a process of evaluating an existing one, the user should have a means of tracing through selected paths of the network. This will indicate to the user whether those parts of the networks are logically complete, if the intended CMs and OMs will be executed, and will determine the data sets required by that particular path.

3.1.3.3 Preparing the Data

There are two distinct parts to the process of preparing the data. The first part is to establish those sets of variables, termed unqualified data sets, that are required by the job. The second part is to provide values for the set of variables. In this condition, the data set is said to be qualified.

When the user selects an existing job, the IPAD System will provide a list of the unqualified data sets that must be provided for that job. The process is more difficult when a new or modified job is to be used. In this case, the IPAD System should assist the user in recognizing first the individual variables required. Then, the System will support searches in the library for unqualified data sets that contain these variables. Finally, the user will define new data sets as required.

Once the unqualified data sets have been determined, the user must qualify the data sets, or, in other words, tell the System which version of the data set contains the values desired during execution. In the cases where a given job is a part of a larger network, most of the inputs will actually be data sets produced by preceeding jobs. The user must determine the qualifier for the proper set in this instance. The user will then supply the remaining input data which was not produced by preceeding jobs.

Once the data is all collected into qualified data sets, the values should be examined for errors prior to actual execution. Gross errors may be found by the IPAD System, based on information in the library dictionary (see 3.1.6.1). An example would be that Mach number would be limited to $0 \leq M \leq 100$. Any value outside this limit could easily be detected by the System. Other examinations would be done by visual inspections of plots from on-line graphics and off-line plotters. Also, the System should be able to compare two data sets and inform the users of differences between the two. The user would be able to confirm that his intended change had been achieved, from this information.

3.1.4 Executing a Job

The schedule for a particular subtask will be identified by the project plan which will be recorded by the Management Information System. The user will execute the jobs that are required for the subtask. The user's effort will be divided between procedure development (preparing the code for the job) and data preparation (selecting the data to be used as input). These aspects were considered in section 3.1.3.

The execution of a job in IPAD should be a relatively simple matter. The execution process should be able to run unaided, or the user must be able to monitor, interrupt or interact during the execution of the network sequences. The capability must be provided to execute a job both by batch processing or by active monitoring at a personal terminal without sacrificing any of the data management characteristics of the IPAD System.

3.1.5 Examining the Results

The IPAD System should have a general display capability for examining results. This must support the display requirements of the typical user and the display requirements for a Management Information System. This capability should be based on a method of accepting specific display formats and the user need only specify the format and qualified data sets to be plotted. The displays should be graphical and textual, and will be done using on-line graphical devices, hardcopy printers and plotters, automatic type-setters, and so on. These display formats should be preserved as a part of the code base, and will become the basis for a standard documentation processes.

The Management Information System will use display formats to present data to management. Planning and control of resources as well as management of the product activities will be guided by information presented from the data base by the display formats. In addition to the usual output devices, control rooms may require projection of management information onto screens.

The user will employ selected display formats to initially determine if his results are essentially correct. Later, the display formats will be used for detail examination and interpretation of the results. Finally, the user will document his results for communication outside the IPAD environment. The documentation will use selected display formats to produce text as well as high quality line drawings and graphical presentations.

3.1.6 Other Considerations

3.1.6.1 User Community Discipline

Perhaps the most important result of the IPAD environment will be to bring a consistent discipline to the user community. This discipline means consistency both in the definition of variables and in the processes that produce values for the variables. Discipline will be achieved partly through the management structure of the user's organization, and partly through features of the IPAD System.

There should be a library maintained by the System, available to all the users. Within this library will be a dictionary of standard definitions, user definitions, variables and their gross ranges, and other terms as required to introduce consistency to the user community.

3.1.6.2 Continuity Over Task and Time

The IPAD System must provide for continuity in the day to day activities as well as the on-going long term activities of the using community. During day to day activities, the user must be able to interrupt a subtask at any reasonable point, inlluding a job in execution, place the subtask in a hold status, and return at an indeterminate later time to resume the subtask without penalty. This will provide the user convenience for completing subtasks.

In addition, the IPAD System must provide for continuity over an entire project network which may span several years. This may best be supplied in conjunction with the Management Information System, because continuity of results and timeliness of effort must be the product of the coordinated effort of all the users working to design the same product. Users, working on a subtask, may be located both physically and organizationally remote from some of their data sources. The IPAD System must provide communication regarding availability of data and must preserve that data.

3.1.6.3 Security and Control of the Data Base

There are three categories of control that must be considered for the IPAD System. The first will establish the extent to which the user will be allowed to use the features of the IPAD System. This control will be provided by permission codes. The second will establish the extent to which the user will be allowed to access the data base. This control will be provided by access codes. The third are the security codes which are identified by corporate or government requirements. For this type of security, use of the IPAD System and host computer must comply with the established regulations.

3.1.6.4 Management Information System

The titles of Management Information System and Management Information Display have been used to identify items of management information discussed in this document. The capability to provide control of the progress of the work accomplished relative to plans, schedules and status reports of work required for an identified project is an IPAD System design requirement. However, the capability to display information from the IPAD data base which is pertinent to management information items such as budget expenditures, technical reports, comparisons of evaluation parameters, etc., are OM functions and Technical Program Elements will be required to identify and format the desired information.

3.2 USER LANGUAGE

The language structure to be employed by the user as he communicates with the IPAD System is of great importance. That language must convey in a concise set of primary commands all of the functions and capabilities of the IPAD System. Further, that language and its rules must preserve the balance of human management authority within IPAD that the user is familiar with outside of IPAD. The command language postulated below has been derived by representatives of the future IPAD user community, and is felt to meet these needs.

Before presenting the 13 basic commands some terms and related rules will be defined. Then, general rules for this language will be stated.

3.2.1 Definition of Terms

The terms used to describe the user language are presented and defined.

3.2.1.1 Access Code

Access codes will be required to authorize the following functions.

1. Write - the user may write into the item and change its contents. Except for edit in place (see Section 3.2.3.5), the user gets a copy, and the result of his modifications will require a new qualifier.

2. Read - the user may read, but not write into the item. The user cannot get his own copy, in this case.

3. Extend - the user may write into the item only to extend the item. He cannot change the existing contents. The extended item will be requalified.

4. Execute - this code applies to operational modules and jobs. The user may have a copy of the item in executable binary form only.

3.2.1.2 Coding Module

Coding modules are the smallest division of user source code that can be qualified in the IPAD environment. There are two classes of information pertaining to the coding module. When unqualified, the only information will be a dictionary entry giving the generic name, key word list, generic abstract, and references. There will be only one of these entries for each generic group of coding modules. The generic abstract will describe the purpose and plans for that group of coding modules.

When qualified, the above information will be extended by the qualified name, access codes, security codes, status, specific abstract, the list linking internal variable names to engineering variables in the dictionary, and the source code itself. The status will indicate whether the code is not checked out, verified, or legally certified. The specific abstract will contain additional information describing this particular variant of the coding module.

3.2.1.3 Community Library

The community library is the library common to a community of users. There will be a general dictionary for each community library, and each user in the community will have available all the information in the community library, subject to limitations contained in the access, permission or security codes. There may be more than one community library in each IPAD.

3.2.1.4 Data Set

A data set is an identified, organized collection of alphanumeric information. It may be made entirely of text, of numerical information, or a combination of the two. Data sets will exist in two classes. First, there will be given to the dictionary the unqualified information of generic name, key words, generic abstract and references. In the qualified state, a data set will have associated with it the qualified name, access codes, security codes, and the specific alphanumeric contents. Those contents will be established by the particular stored data definition used in storing those contents.

3.2.1.5 Dictionary

There will be two dictionaries known to each user in the IPAD System. First, the community library dictionary will contain the definitions and information that are the standards for all of the dictionary types available to the users in that community. Second, each user will have a personal dictionary in his private subtask library. This dictionary will define all of the dictionary types required by the user to do his subtask which are not contained in the community library.

The discipline that will require the user community to agree on coherent standards and definitions will be largely motivated by the dictionary in the community library. It may be desirable in many community libraries to prevent, via permission codes, any users from establishing private dictionaries in their own subtask libraries. This would force all users to rely on the standard dictionary of the community, thus making all information known to the entire community. Access codes would still prevent unauthorized use and in certain cases, permission may be granted for a user to have his own dictionary.

There are several types of entries into these libraries, denoted as dictionary types. They are coding modules, data sets, display formats, engineering variables, jobs, operational modules, and stored data definitions. Each is defined separately in this section.

3.2.1.6 Display Format

A display format is a special class of job that is used for displaying data sets by graphical methods, whether on line or off line. It has been made a special class of job so that the action of display may be done with ease to support the Management Information System (see section 3.1.5).

The dictionary information for a display format will consist of name, abstract, reference and the display format source code. Display formats will not be qualified, as their unauthorized alteration or destruction is merely a question of nuisance rather than one of integrity of results.

3.2.1.7 Engineering Variable

An engineering variable has one standard definition in the dictionary. There is uniqueness by definition of this variable. There may be alternate names for this variable contained in the coding modules of the user community. But in engineering or scientific terms, there should be only one entry of this variable in the dictionary. The dictionary will contain the dictionary name of the variable, a key word list and an abstract.

3.2.1.8 IPAD

The Integrated Program for Aerospace-Vehicle Design (IPAD) is defined to include the technical data, technical program code and the IPAD System software. All computing systems outside of IPAD will be considered as remote systems. There may be more than one IPAD per host.

3.2.1.9 Job

A job is a network of operational modules whose code is in executable form. The job is the collection of code the user submits to the host computer for execution. The definition of a job in the dictionary will consist of two states. In the unqualified state, the information consists of generic name, key words, generic abstract, and references. In the qualified state, the information added describes the qualified name, access codes, security codes, specific abstract, and the higher-level language statements describing the network of the job.

3.2.1.10 Keywords

The key word list is an important item for each dictionary type. This list will allow users to search and find existing dictionary types that will likely fill their needs. This capability will help to limit the number of entries in the library which contain the same information or perform the same activity.

3.2.1.11 Operational Module

An operational module is a network of coding modules, whose code is in executable form. The definition of an operational module in the dictionary will have two states. In the unqualified state, the information consists of generic name, key words, generic abstract and references. In the qualified state, the information added describes the qualified name, access codes, security codes, specific abstract, and the higher-level language statements describing the network of the coding modules.

3.2.1.12 Permission Code

The permission codes will be equivalent to a management system within the command language. The typical user will operate with capabilities and limits established by permission codes. These are constraints on the user, in contrast to access codes, which are constraints on library information. There are several classes of control contained in the permission codes. A partial list of these classes is presented.

1. User entry into command language states--each user will have a profile of permission codes declaring which commands of the language he is allowed to use. For example, only designated users might be allowed to construct code. Use of the purge command to destroy library contents, even by the owner of the contents, may also be limited by the permission codes. In summary, this application of permission codes manages the user by controlling the parts of the command language he is allowed to use.

2. Management of the data base--only selected users, representing management, will have the authority granted by permission codes to control the data base. As examples, certain users will have permission to issue absolute purge commands. Other permission codes will allow selected users to alter the access codes normally established by the owner of the library item. In summary, this application of permission codes manages what the user has done after it has entered the library.

3. Management of security--selected users will have security monitor status, as afforded by permission codes. These monitors will have the power to establish and alter the security classification of a library item.

4. Monitoring the subtask activities--selected users will have the ability to examine subtask libraries granted by the permission codes. This allows management to monitor the process of the user activities while the user is still working within his subtask.

3.2.1.13 Qualifier

Qualifiers are the means of distinguishing between sets of library information with the same type and generic name but with different contents. Qualifiers also identify and help track the process by which the qualified information was created. The qualifier will contain a part given by the user, to be used as his notes on the item, and an automatic part generated by the IPAD System.

The qualifiers are somewhat different for the various dictionary types. Engineering variables, display formats and stored data definitions are not qualified. Identity of the owner will be sufficient information for these items. For coding modules, operational modules and jobs, there will be a user part of the qualifier, plus the system part that will describe the particular item in sufficient detail to establish its uniqueness. For data sets, the user will provide part of the qualifier, and the system part of the qualifier will contain sufficient information to allow the events leading to the generation of the contents of the data set to be repeated.

3.2.1.14 Security Code

Security codes are those established to meet corporate or governmental rules. The IPAD System and host computer must have provisions to comply with the established security regulations.

3.2.1.15 Stored Data Definition

The actual structure of the contents of a data set will be specified by the stored data definition. The stored data definition will identify the engineering variables and the order of those variables in the data set. Consequently, part of the qualifier of a data set will contain the identity of the stored data definition used to store that data set.

3.2.1.16 Subtask Library

The subtask library is where the individual user performs his subtask. The user may establish his own dictionary in this library, and has privacy from the other users in the community. Only users with permission codes to examine subtask library contents may violate this privacy.

3.2.2 General Rules

The following rules are of a general nature and apply to all of the command language. Other rules will be stated concurrent with the language command that they are associated with.

Rule 1--There will be provision for more than one community library in each IPAD. In this way, the community library becomes an additional qualifier on the contents of a particular IPAD. This will be helpful, for example, in performing key word searches, in that airplane and hydrofoil terms would not be interposed. Also, the provision for multiple community libraries may aid the management process. For example, company archives may be a separate community library.

Rule 2--There will be seven dictionary types, namely, coding module, data set, display format, engineering variable, operational module, stored data definition, and job. There will be uniqueness of unqualified names only within dictionary types and community library.

Rule 3--The owner of an entry in the community library or its dictionary will be the user who makes the entry.

Rule 4--Only the commands of ENTER, SEND and TRANSFER will allow the user to cross the boundary of the community library he is in, and only if allowed to do so by the permission codes.

Rule 5--A user in his own sub-task can access information only in the community library or his own subtask library. He must have information in another subtask library transferred to him by the owner of that subtask library, unless he has permission code authority to enter that subtask library.

Rule 6--Each command state will be interactive to the user in three degrees of conversation. For the beginning user, the dialog will be conversational, with the IPAD System requesting information, and the user responding. For the average user, the IPAD System will prompt by means of one word queries. For the most experienced users, the IPAD System will make no remarks on its own; the user will supply all information without aids. Of course, the user can change to any of these three modes of dialog at any time.

Rule 7--The IPAD System will maintain a message file for each user. When the user signs on into any subtask, the System will indicate to the user if any messages are in his message file.

3.2.3 Command Language

The following commands are all on the same level, and may be done in any order, providing that the prerequisites for the actions requested by the command have been done. This level of command is available as soon as the user has performed the log-on and initiation activity, including the naming of the community library and subtask library he wishes to be in. The command language is presented in a typical order of use for a totally new subtask. Table 3.1 summarizes the information discussed for each command and may be used for reference.

3.2.3.1 DEFINE

This is the command for entering or modifying definitions of unqualified dictionary types into the dictionary. The following information is required.

Command:

DEFINE/PLACE/TYPE/TITLE/DOCUMENT/RANGE

PLACE--The user specifies either the community library dictionary or his private subtask library dictionary.

TYPE--The user denotes which dictionary type is being defined.

TITLE--The unqualified name is given. For display formats, engineering variables and stored data definitions, this will be the only name.

DOCUMENT--A list of key words, then a generic abstract and references are given. This command does not apply to stored data definitions.

RANGE--For engineering variables only, a gross range that the value of the variable is never to exceed is given to the dictionary.

RULES--The following are special rules for this command.

1. The permission codes will prevent certain users from using this command.

2. A definition to the dictionary may be modified only by

   a. the owner,

   b. users with permission code authority to modify definitions other than their own.

3. The definition in the community library dictionary can not be changed or purged if there are qualified entries in the communicity library depend on this definition.

4. There must be an entry in the community library dictionary for each item stored in the community library.

3.2.3.2 ENTER

This is the command for entering information into the IPAD data base. The following information is required.

Command:

ENTER/PLACE/TYPE/TITLE/SECURITY/ACCESS/STATUS/DOCUMENT/LINK LIST/REMOTE

PLACE--The user specifies either the community library or his private subtask library as the destination.

TYPE--The user denotes which dictionary type is being entered. The engineering variable type is not valid, as all of the contents for this type are entered under DEFINE.

TITLE--The user gives first the generic name as known to the dictionary, then specifies his part of the qualifier. The exception is for display formats and stored data definitions; for these, only the name is given.

SECURITY--The legal security classification (codes) for the contents to be entered. This is required only for coding modules, data sets, operational modules and jobs.

ACCESS--The owner assigns the access codes for the contents to be entered. This is required only for the types listed under SECURITY.

STATUS--This is given only for coding modules, and declares whether the code is not checked out, verified, or legally certified.

DOCUMENT--Special documentation differing from that given to the dictionary is entered. Additions to the generic key word list are not allowed.

LINK LIST--This applies only to coding modules, and gives information relating internal variable names to engineering variables.

REMOTE--First the device, then its "address," and, for data sets, the stored data definition is given to be used in entering the information into the user's IPAD.

RULES--The following are special rules for this command.

1. The permission codes will prevent certain users from using this command.

2. The user can make entries only into his own community or subtask library.

3.2.3.3 TRANSFER

This command is the means of moving information, other than dictionary definitions, to different community or subtask libraries within the same IPAD. The following items are required.

Command:

TRANSFER/PLACE/TYPE/TITLE/DESTINATION

PLACE--The user specifies either the community library or his private subtask library as the source.

TYPE--The user specifies which dictionary type is being transferred. Engineering variables are the exception; they must be entered as an action under DEFINE.

TITLE--The user gives the generic name and enough of the qualifier to uniquely select the contents to be transferred.

The exception is for display formats and stored data definitions; for these, only the name is given.

DESTINATION--The user declares the community and subtask library that the information is to be copied into.

RULES--The following are special rules for this command.

1. The permission codes will prevent certain users from using this command.

2. TRANSFER is not a purge. The information is not transferred; only a copy is transferred.

3.2.3.4 SEND

This command is the means for moving information from the user's IPAD to another IPAD or remote location. The following items are required.

Command:

SEND/PLACE/TYPE/TITLE/REMOTE

PLACE--The user specifies either the community library or his private subtask library as the source.

TYPE--The user specifies which dictionary type is being sent. The dictionary information on engineering variables can be sent, as well as can the entire contents of any other type.

TITLE--The user gives the generic name and enough of the qualifier to uniquely select the contents to be sent. For display formats, engineering variables and stored data definitions, only the generic name is required.

REMOTE--First the device, then its "address", and, for data sets, the stored data definition is given to be used in sending the information from the user's IPAD.

RULES--The following are special rules for this command.

1. The permission codes will prevent certain users from using this command.

2. A record of a SEND and the user to contact for reference will be maintained.

3. An IPAD receiving an engineering variable via a SEND will not accept that information. An IPAD receiving any of the other dictionary types via a SEND will accept the contents, only if the proper dictionary information has been entered in the receiving IPAD with DEFINE commands. That is to say, dictionary definitions will be made only thru DEFINE.

4. SEND is not a purge. Only a copy of the information is sent.

3.2.3.5 EDIT

This command is a means of locating and changing information, other than that input via DEFINE, in the community and subtask libraries. The following items are required.

Command:

EDIT/PLACE/TYPE/TITLE/SECURITY/ACCESS/STATUS/TEXT/COMPILE

PLACE--The user specifies either the community library or his private subtask library as the source.

TYPE--The user specifies which dictionary type is having its contents edited. Engineering variables cannot be edited except thru DEFINE.

TITLE--The user gives the generic name and enough of the qualifier to uniquely select the contents to be edited. For display formats and stored data definitions, only the generic name is given.

SECURITY--For coding modules, operational modules, jobs and data sets, the user gives the modification to the present security code.

ACCESS--For the same four types as above, the user gives modifications to the access codes.

STATUS--For coding modules, the user can change the status of a set of code. The three choices are verified, legally certified or not checked out.

TEXT--This is a large set of higher level editing commands. This language will be drawn from existing editing languages.

COMPILE--In the construction of a coding module or display format, the user will require all of the host system's compiler functions.

RULES--The following are special rules for this command.

1. Any user may edit in his own subtask library. The permission codes may prevent a user from editing entries in the community library.

2. The ability to edit security codes requires permission code authority.

3. No item of information that is entered thru DEFINE can be accessed or edited by EDIT.

4. The ability to edit in place, that is edit contents without changing the qualifier, requires that no other entry in the community library depends on this entry for its generation.

5. The link list for a coding module is considered to be a part of the contents of that coding module, for editing purposes.

3.2.3.6 PURGE

This command is the means of erasing information, other than information entered through DEFINE, from community and subtask libraries. The following items are required.

Command:

PURGE/PLACE/TYPE/TITLE

PLACE--The user specifies either the community library or his subtask library as the location of the information to be purged.

TYPE--The user specifies which dictionary type is having its contents edited. Engineering variables cannot be purged except through DEFINE.

TITLE--The user gives the generic name and enough of the qualifier to uniquely select the contents to be purged. For display formats and stored data definitions, only the generic name is given.

RULES--The following are special rules for this command.

1. If the permission codes allow a user to use PURGE, then that purge request will have absolute authority only within the user's subtask library. Purge requests issued by the user to purge information from the community library will be subject to the following conditions.

    a. Purge requests on data sets will not be honored if a qualified data set in the community library depends on the set to be purged for its generation.

    b. Purge requests on coding modules, operational modules and jobs will not be honored if a qualified operational module or job in the community library depends on the item to be purged for its execution.

    c. A stored data definition cannot be purged if a qualified data set in the community library lists it as necessary to read the data set.

2. The permission codes will give absolute purge authority, unrestricted by Rule 1 above, to selected users.

After this absolute purge has occurred, a record will be kept of the purge action and the user to contact for reference.

3. Information entered through DEFINE cannot be erased thru PURGE.

3.2.3.7 COMPARE

This command is the means of making comparisons between similar information or comparisons of information with given values. The following items are required.

Command:

COMPARE/PLACE/TYPE/TITLE/FUNCTION

PLACE--The user specifies either the community library or his subtask library as the location of the information to be compared.

TYPE--The user specifies which dictionary type is having its contents compared. Engineering variables cannot be compared, and the options under FUNCTION may apply selectively to some of the types.

TITLE--The user gives the generic name and enough of the qualifier to uniquely select the contents to be compared. For display formats and stored data definitions, only the generic name is given.

FUNCTION--There are several types of COMPARE activities listed below. Other types may be added as needed.

1. RANGE - each engineering variable in the dictionary has a gross range associated with it. For example, a Mach number may be limited to be not less than zero nor more than 100. A gross range check would reveal any Mach numbers in a data set that were outside this range.

2. DIFFERENCE - for this option, the PLACE and TITLE information is repeated for a second item of the same TYPE. The two are then compared, and the system indicates where the contents of the two are different.

RULES--There are no special rules for COMPARE.

3.2.3.8 CONSTRUCT

This command is the means of developing new operational modules, display formats and jobs, and of modifying or interrogating existing operational modules, display formats and jobs. The following items are required.

Command:

CONSTRUCT/TYPE/TITLE/NETWORK/SIMULATE

TYPE--The user specifies whether an operational module, display format or job is being constructed.

TITLE--The user gives the generic name, then his part of the qualifier, if he is building a new item. If he is modifying, then he gives the generic name enough of the qualifier to uniquely select the item. Display formats have no qualifiers, therefore only the generic name is given.

NETWORK--The network for the type is developed, using a higher level language specifically for the purpose of network specification.

SIMULATE--This allows the user to select a given path thru the network. The IPAD System will trace this path, and indicate the data sets that appear as input and output and that are internally connected along this path.

RULES--The following are special rules for this command.

1. The permission codes may prevent a user from using CONSTRUCT.

2. The higher level network language will allow examination of computed values, but will not allow alteration of computed values for variables. All generation of data must be done inside coding modules.

3.2.3.9 EXECUTE

This command is the means of executing jobs. The following items are required.

Command:

EXECUTE/PLACE/TITLE/DATA/PARAMETERS

PLACE--The user specifies whether the job is in the community library or his private sub-task library.

TITLE--The user gives the generic name and enough of the qualifier to uniquely identify the job to be executed.

DATA--The user identifies by generic name, qualifier and place, the data sets to be used during the execution of this job. The user will also indicate disposition of the data sets once execution is complete. The IPAD System will assist the selection of data sets by providing the list of data sets required by the job.

PARAMETERS--The user specifies those job control parameters allowed by the job for internal control during execution.

RULES--There are no special rules for this command.

3.2.3.10 DISPLAY

This command is the means of bringing information from community or subtask libraries to a display device in a defined set of display formats. A display format is actually a special class of job that is executed via DISPLAY rather than EXECUTE. This is intended to allow information display to be done as easily as possible. Once the display format begins execution, the user will be in a dialog mode to specify the information concerning the data sets to be used in generating the displays. Of course, for batch jobs, the information will be given beforehand. Furthermore, display formats are not qualified, as a violation of their integrity does not affect the integrity of any of the results in the libraries. Consequently, only two items of information are required to get the display format into execution.

Command:

DISPLAY/PLACE/TITLE

PLACE--The user specifies whether the display format is located in the community library or in his subtask library.

TITLE--The user gives the generic name of the display format to be used.

RULES--There are no special rules for this command.

3.2.3.11 FIND

This command is the means of locating selected information in the libraries. The following items are required.

Command:

FIND/PLACE/TYPE/SEARCH

PLACE--The user tells whether to search in the community library or his own subtask library.

TYPE--The dictionary type to be searched is given. All types are allowed.

SEARCH--A special language for searching. There are different types of searches.

1. Information entered thru DEFINE can be located by means of key word searches.

2. Information in data sets can be found by key word searches on engineering variables known to be in that data set.

3. A user gives the name and qualifier of a data set to the IPAD System, and the System will tell the user all of the entries in both the community library and

subtask libraries which have used that data set. This information will enable a user to inform dependent users of an error in the data set.

RULES--There are no special rules for this command.

3.2.3.12 MESSAGE

This command is the means of sending a personal message to another IPAD user. The following items are required.

Command:

MESSAGE/ADDRESS/CONTENT

ADDRESS--The address information in sufficient detail to locate the user to whom the message is intended.

CONTENT--The content of the message.

RULES--If the receiving user is not signed on when the message is sent, the originator can have the message entered into the user's message file.

3.2.3.13 LEARN IPAD

This is a tutorial state whose purpose is primarily to teach the new user how to use IPAD. It may also be used in getting assistance in case of difficulty.

3.2.3.14 General Commands

These are a series of general commands available to the user at any time.

PAUSE--This is a short-term exit from a command state. The status is preserved for a subsequent RESUME command.

HOLD--This is a long term exit, such as log-off without completing the subtask, from a command state, with the ability for interruption of the subtask. The status is preserved for a subsequent RESUME command.

RESUME--This resumes the activity that was in progress when the PAUSE or HOLD command was issued.

STOP--This is an unconditional stop, without recovery ability.

RULES--There are no special rules for these commands.

Table 3.1

Application of Command and Dictionary Actions to Dictionary Types

ITEM		DICTIONARY TYPE: CODING MODULE	DATA SET	DISPLAY FORMAT	ENGINEERING VARIABLE	OPERATIONAL MODULE	STORED DATA DEFINITION	JOB
DICTIONARY ENTRY?		X	X	X	X	X	X	X
QUALIFIED?		X	X			X		X
COMMAND	SUB-COMMANDS							
DEFINE	PLACE, TYPE, TITLE	X	X	X	X	X	X	X
	DOCUMENT	X	X	X	X	X		X
	RANGE				X			
ENTER	PLACE, TYPE, TITLE	X	X	X		X	X	X
	SECURITY, ACCESS	X	X			X		X
	STATUS, LINK LIST	X						
	DOCUMENT, REMOTE	X	X	X	X	X		X
TRANSFER	PLACE, TYPE, TITLE, DESTINATION	X	X	X		X	X	X
SEND	PLACE, TYPE, TITLE, REMOTE	X	X	X	X	X	X	X
EDIT	PLACE, TYPE, TITLE	X	X	X		X	X	X
	SECURITY, ACCESS	X	X			X		
	STATUS, COMPILE	X						
	TEXT	X	X	X		X	X	X
PURGE	PLACE, TYPE, TITLE	X	X	X		X	X	X
COMPARE	PLACE, TYPE, TITLE	X	X	X		X	X	X
	FUNCTION: RANGE		X					
	FUNCTION: DIFFERENCE	X	X	X		X	X	X
CONSTRUCT	TYPE, TITLE, NETWORK, SIMULATE			X		X		X
EXECUTE	PLACE, TITLE, DATA, PARAMETERS							X
DISPLAY	PLACE, TITLE			X				
FIND	PLACE, TYPE, SEARCH	X	X	X	X	X	X	X
MESSAGE		N/A	N/A	N/A	N/A	N/A	N/A	N/A
LEARN IPAD		N/A	N/A	N/A	N/A	N/A	N/A	N/A

4.0 USER-TERMINAL ENVIRONMENT

4.1 PERSONAL TERMINAL ENVIRONMENT

The user will have a choice of doing a job in either the batch mode or from a personal terminal. The terminal mode of operation should satisfy certain human factors in order to achieve an optimum reduction in flowtime. The terminal mode should not frustrate the user. Dialog with the computer through the terminal should have response times compatible with the user's needs to think and communicate effectively. For instance, response times by the computer of less than one or two seconds are probably necessary if intensive creative dialog is to be done.

The language the user will employ to communicate with the computer must compromise somewhere between the limits of boredom and bewilderment. The user side of the dialog should accommodate the casual user as well as the dedicated user. The casual user will want to employ a highly descriptive language whereas the dedicated user will require a style approaching mnemonic language. The computer side of the dialog should provide effective prompting, dependent on the skill of the user.

4.2 TERMINAL EQUIPMENT ARRANGEMENTS

Careful arrangement of the user's facilities will be required to achieve the best use of personal terminals in the IPAD environment. There should be different types of work arrangements, dependent on the size of the using unit, their class of activity, and the particular equipment to be used.

For example, figure 4.1 shows an engineering organization of approximately 225 people. The central rooms will contain the on-line terminal equipment and off-line printers and plotters to support the using organization and its interface with IPAD. The equipment density represents projections for the time period of 1976 thru 1979.

Figure 4.2 shows an arrangement for a unit composed of eight users. These eight users would be a basic group which represents the major technical disciplines required to support each design network identified in section 4 of Volume II for the preliminary design Levels. Their IPAD terminals would consist of two silent teletype units and one interactive graphic console with vector plotting capability. With the equipment centered in the work area, access would be convenient, and terminal

ORIGINAL PAGE IS
OF POOR QUALITY

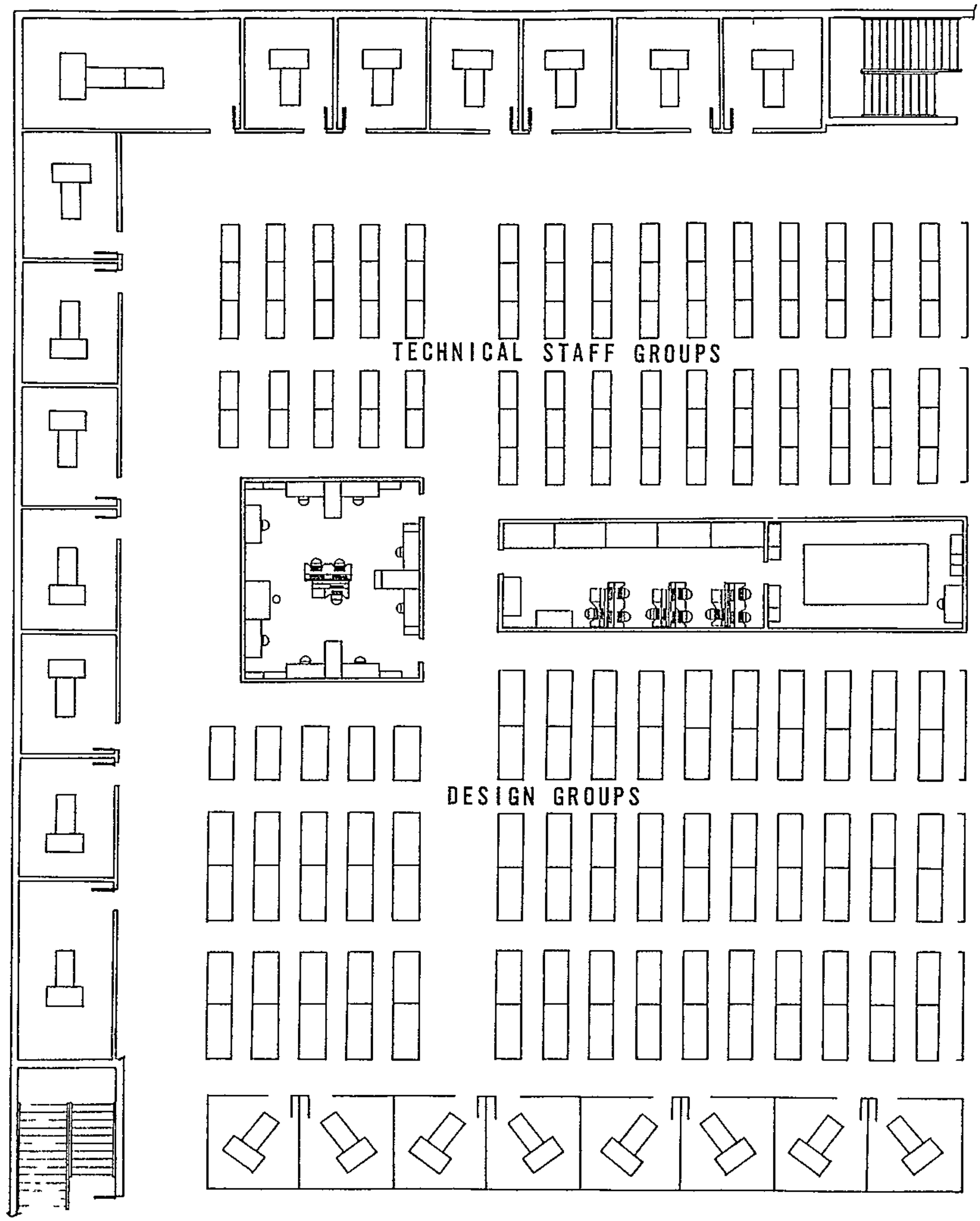


Figure 4.1 Example — IPAD User Organization Facilities — General Arrangement

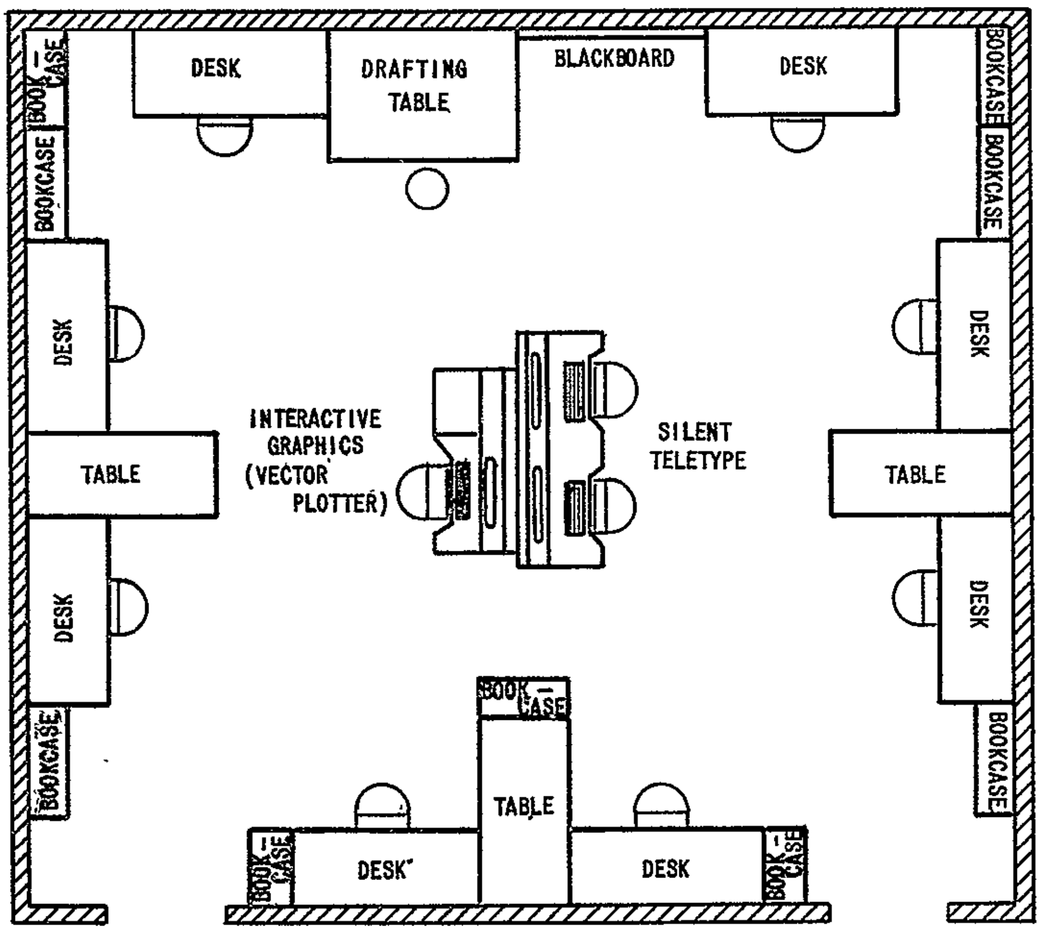


Figure 4.2 Example–IPAD User Peripheral Equipment– Basic Group Online Facilities Arrangement

availability could be determined without having to get up and go to another area. Furthermore, a job that would require sequential execution by different members of the unit, such as an examination of a configuration in Levels III and IV, could be done in a sequential manner at the terminal. Continuity of the solution would be present in the user's mind as well as in the IPAD System.

Figure 4.3 shows a central arrangement of on-line terminal equipment to support approximately 150 other engineers having less terminal support requirements. This room has six silent teletype units and three interactive graphics consoles with vector plotting capability. This room also has two manual digitizers and three work tables.

Figure 4.4 shows a central arrangement of off-line plotting equipment to support the entire organization. This room has one high volume low accuracy plotter, two medium volume high accuracy plotters and one very accurate drafting plotter. High volume printers and medium accuracy film plotters are assumed to be in a central facility remote to the using organization.

This arrangement is one example of many ways to provide the user access to IPAD. Effective utilization of IPAD will require facility planning, as well as the more obvious selection of the peripherial equipment itself.

4.3 MANAGEMENT INFORMATION DISPLAY

Display of management information will be accomplished by special jobs which provide pre-formatted displays of data. This will provide management the capability to review plans, schedules and costs. The plans will outline the sequence of design and analysis tasks and the technical depth of design and analysis required at each step. As the design progresses, the plans will be updated to show the completion of each step. This will provide management with a well-supported capability to track the costs, such as design engineering and development cost, and estimates of the production cost, product support cost and product operation cost.

Management information system displays will be accessed through personal terminals or project control rooms may have large-screen display devices which project the pre-formatted displays using the latest information from the data base. When information is in question, it may be possible for a manager to interrogate data in successively increasing detail until the source of the discrepancy is found.

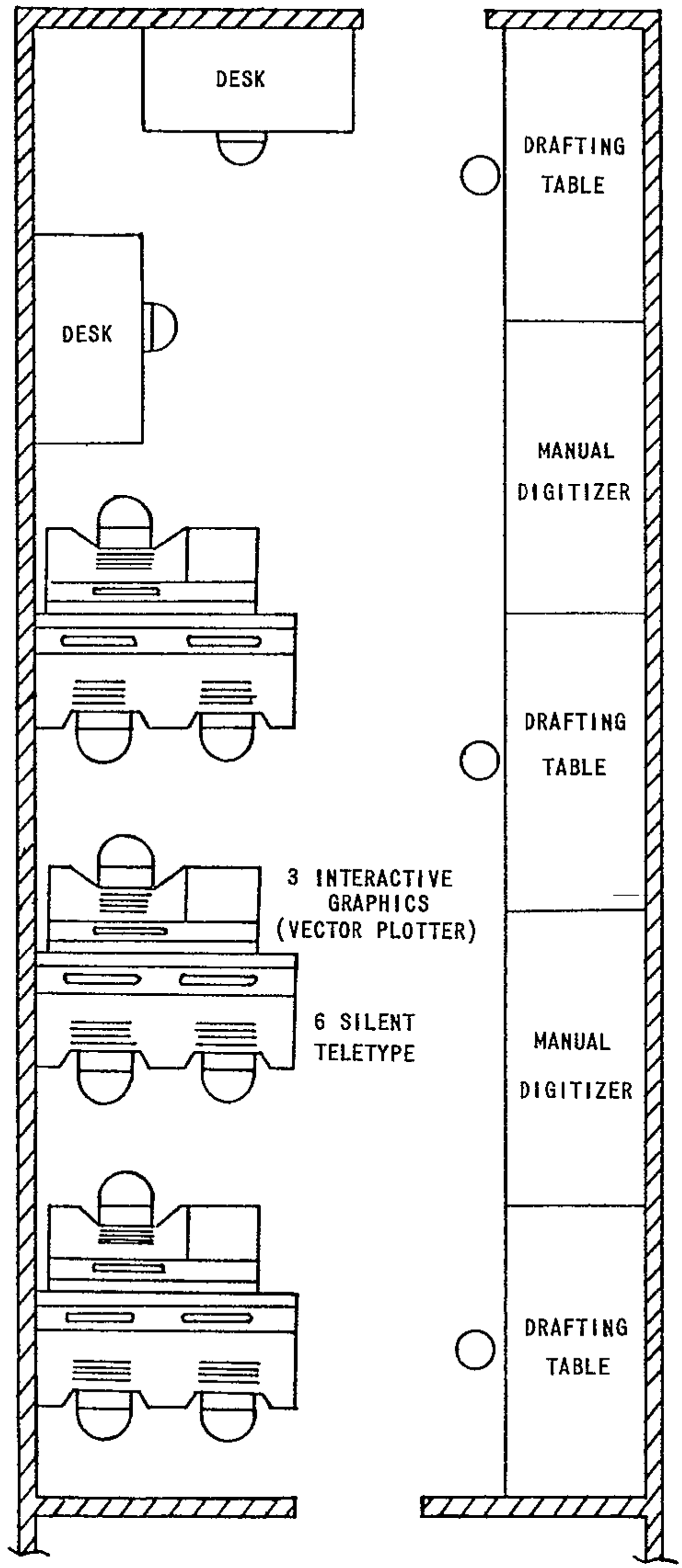


Figure 4.3 Example — IPAD User Peripheral Equipment—Community Online Facilities Arrangement

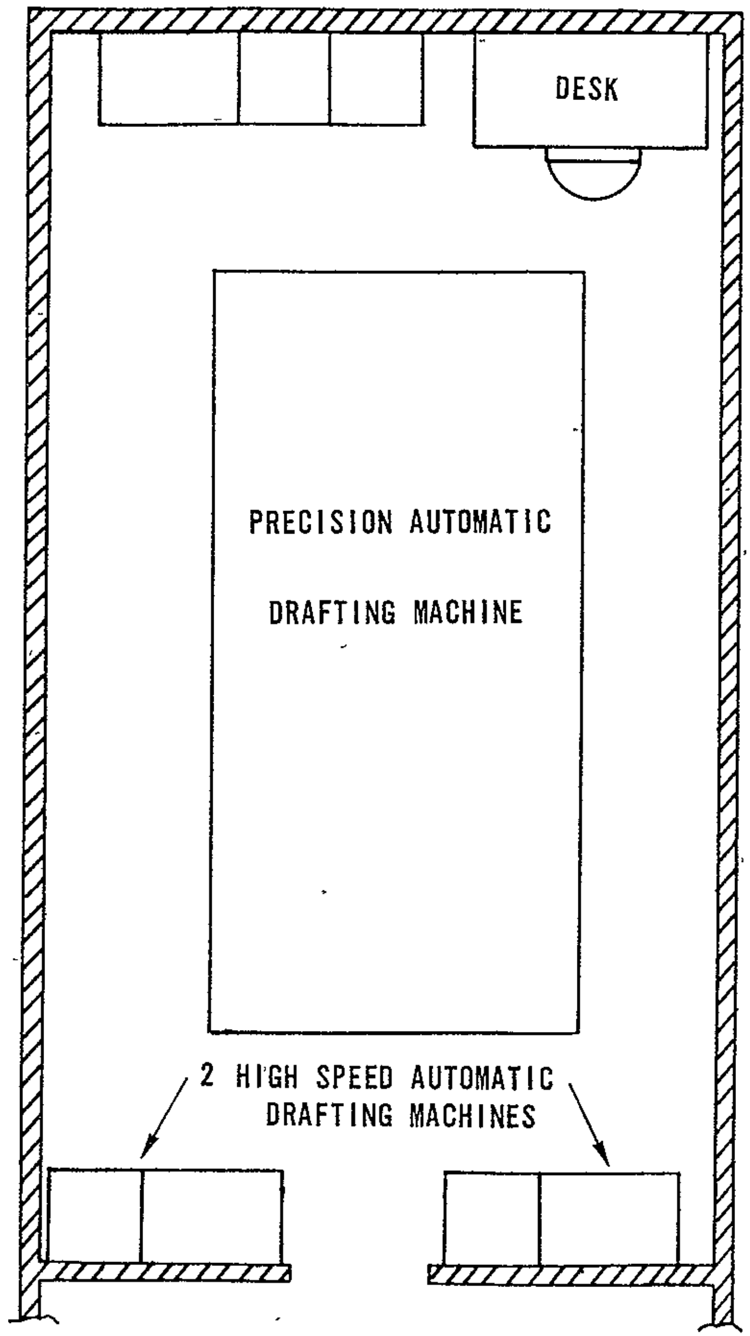


Figure 4.4 Example — IPAD User Peripheral Equipment—Community Offline Facilities Arrangement

5.0 PERIPHERAL EQUIPMENT CAPABILITIES

Efficient use of the capabilities provided by the IPAD system will require a variety of peripheral equipment. There are three distinct categories for this equipment. Personal terminals will provide on-line communications with the computer. Off-line equipment will provide services where time is not the prime factor. Certain special devices will be needed to support the activities involved in the design of the product.

5.1 PERSONAL TERMINAL EQUIPMENT

The largest and most used category is expected to be silent personal terminals. Careful consideration will be required at each IPAD installation to provide the most effective types and equipment density. The following list gives a definition of the types.

1. Keyboard and printer - the traditional teletype, for use primarily in data and code manipulation.

2. Interactive text scope - a text scope, with keyboard. This device only displays text. Interaction will be through the keyboard.

3. Keyboard-interactive graphics scope - a scope with vector plotting capability, as well as text. The interaction will be through the keyboard.

4. Display-interactive graphics scope - a scope with vector plotting capability, as well as text. The interaction will be through devices such as light pens or "control sticks," as well as through the keyboard.

5. Card reader - a low volume card reader for inputting low card volume data.

6. Tape reader - a small on-line tape reader, probably a casette type. Certain data sets and programs where portability is desired may be efficiently stored and loaded in this method.

7. Digitzers - a manually operated device to read curves and introduce them into the system.

8. On-line printer - required to produce immediate hardcopy of text. Document quality is not required.

9. On-line x-y plotter - required to produce immediate hardcopy of graphical displays. Document quality may not be required.

5.2 OFF-LINE EQUIPMENT

This category considers that equipment which is not usually a part of a personal terminal. The common host system equipment will be required, in addition to some devices that are not part of the typical installation. The following types are necessary to support the design process.

1. High volume card reader - the typical input device in present host installations.

2. Automatic digitizer - a large capacity digitizer with automatic line following capability, for inputting large columes of data from drawings.

3. High volume printer - the typical "chain-print" device for producing large volumes of printed output.

4. Document quality printer - required to allow utilization of the IPAD capability for having display formats that prepare reports directly from the data base. This may be a type-setting device or a high-quality typewriter.

5. High volume plotter - a device to produce large numbers of plots for internal company communication. Accuracy is not a high requirement.

6. Drafting quality plotter - to produce highly accurate drawings for design and manufacturing purposes.

7. Tape drive - the typical tape read/write device for entering information and producing copies onto magnetic tape.

5.3 SPECIAL DEVICES

A category of special devices has been identified from requirements established both by the design networks and by the user. Following are two examples of this category. Other devices will be required by additional applications of IPAD.

1. Hybrid simulator - this represents the occasions when the IPAD system will drive any simulation process using an analog computer.

2. Microfilm storage - a system where documented results would be microfilmed. Location information would be maintained as a part of the IPAD data base, and users would employ the IPAD System to retrieve a particular microfilm record.

6.0 COMPUTATIONAL REQUIREMENTS

6.1 GENERAL DISCUSSION

The purpose of this section is to provide an estimate of the computational requirements for the IPAD System host hardware. These requirements were calculated for design projects 1 and 2 enumerated in section 4 of Volume II and are based on the catalog of Technical Program Elements contained in Volume V.

The computational requirements will be expressed in two parts. The first part will express the significant computational information relating to each design level of Project 1 and Project 2. The second part will attempt to identify a total usage requirement for the IPAD System's host hardware in terms of hardware capacity and performance.

The technical tasks for the IPAD Feasibility Study are expressed in terms of design projects and consists of work flow networks and Technical Program Elements which are formulated to solve the postulated design problem. Each design project is broken down into product levels as shown in figure 6.1.

From the technical tasks which are required to solve the postulated design problems, certain information regarding IPAD computational requirements can be formulated for each design level of Project 1 and Project 2. This information is as follows:

A. The purpose of each level;

B. The user environment expressed in terms of the following.

   1. Identification of the primary users of the analysis.

   2. An estimate of the desired computational flow time for executing the analysis in order to be responsive to the needs of the users.

   3. An estimate of the number of configurations which would be executed through the analysis during a typical design program.

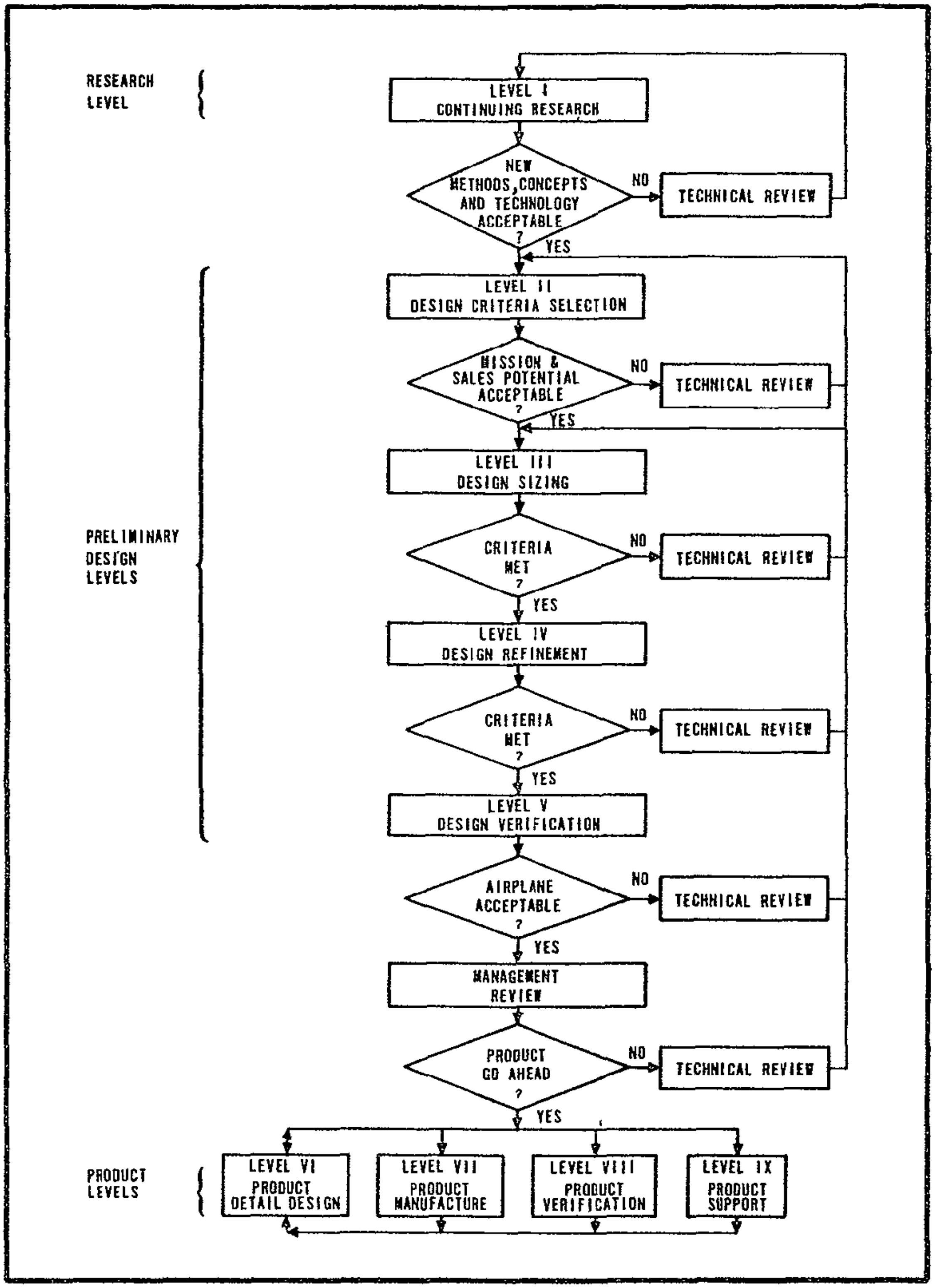


Figure 6.1 IPAD Product Levels

4. An estimate of the number of executions of the level analysis which the user groups would desire to store on-line.

5. A description of the types of runs that will be executed through the analysis. Will they be full or partial executions of the entire analysis? Will they be single or multiple executions (trade studies) of the analysis? Will the analysis be executed in a serial or a parallel manner? Also how will optimization be used in the analysis of each level?

C. The computational size of each level. The analysis of each level is comprised of activities described in Section 4 of Volume II and will be performed by Technical Program Elements which are described in Volume V. Since the Technical Program Elements are interdependent, in order to complete the analysis with a converged design, most of the elements will have to be executed several times for each configuration. A description of how the Technical Program Elements might be ordered to do the analysis is shown in tables for each level. The text describes what the particular order is designed to achieve - e.g., a converged design or a single pass through the analysis. For the later levels (Level V and beyond), since the analyses will be executed in parallel by each technical discipline, the computational activity is estimated in total for each technical discipline.

The computational size of the analysis of each level is represented by the sum of the computational size of each Technical Program Element times the number of times that the Element is expected to be executed. The computational size of each Technical Program Element is represented by Central Processor (CP) seconds of execution (on the CDC 6600), amount of input and output (60 bit) data words, average data transfer rate, and boxes of program source code cards. Each box of source code contains 2000 cards. A typical input/output profile is shown in figure 6.2. It is important to note that the estimates of computational size of each Technical Program Element are based on executing that Element separately and in a batch mode. Consequently, the computational size of each Element will probably increase if it were operated in the IPAD System and in an interactive mode. Also, the computational sizes are based on

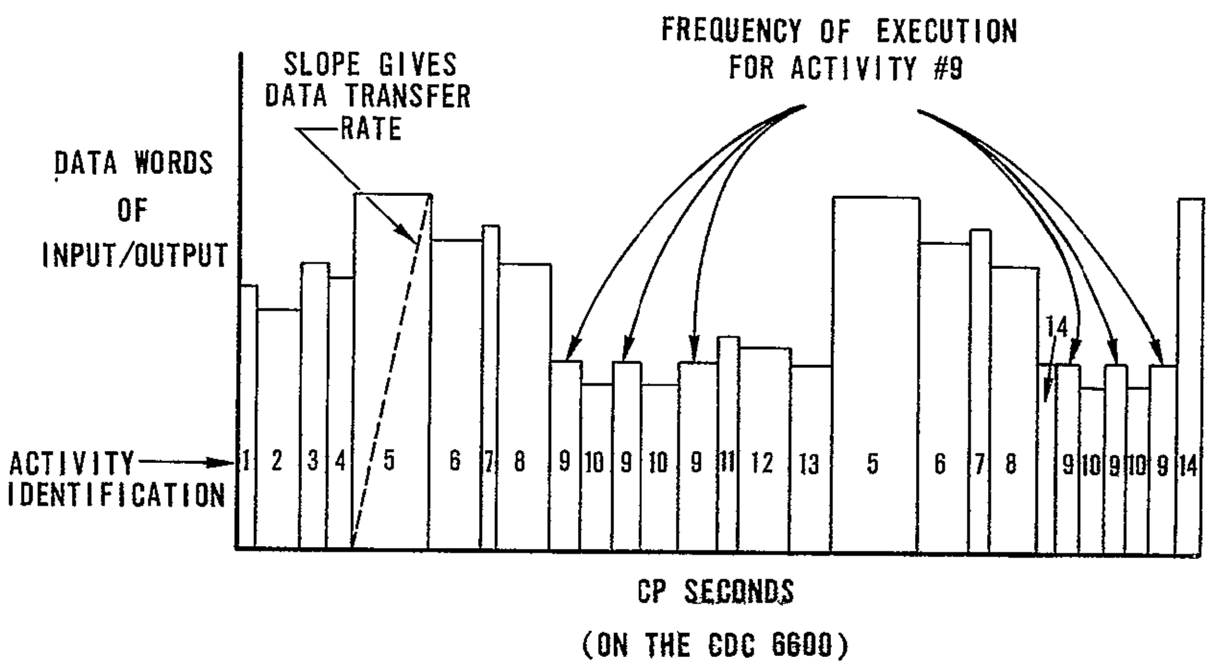


Figure 6.2 Design Computational Requirements for Each Design Level

Elements which are candidates for first implementation of IPAD. Beyond first implementation, it is expected that the capability afforded by an IPAD System will allow the development of more sophisticated Technical Program Elements having increased capability.

D. Execution Frequency and Computational Load. In order to demonstrate what the demand of the analysis of each level might be on the IPAD System host hardware, the information from the user environment is used to develop a continual, averaged daily execution rate. The execution rate is then multiplied by the CP time per execution to develop an averaged daily computational load on the host hardware.

E. Computational Flow Time. The computational flow time represents an estimate of the time it would take to:

1. Develop procedures and prepare data for a job;

2. Execute the job;

3. Analyze and evaluate the output;

4. Document and exchange the output information.

F. Man/Machine Interfaces. The devices listed in this section represent the capability which could be used in conjunction with the Technical Program Elements which are candidates for first implementation of IPAD. Beyond first implementation, it is expected that the demand for more sophisticated devices will increase as more sophisticated Technical Program Elements are developed.

The format of computational requirements (section 6.2 and 6.3) relates information identified as A thru F (above) for each Level of Project 1 and Project 2. Section 6.4 presents the demand on an IPAD System in terms of hardware performance and capacity in order to accomplish the computational tasks of one Project 1 and one Project 2 design effort. In addition section 6.4 contains an estimated company mix based on these projects and is intended to represent all areas of a major aerospace company.

6.2 PROJECT 1 - SUBSONIC COMMERCIAL TRANSPORT

Design Project 1 was chosen to be the design of a subsonic commercial transport. The design Project 1 was divided into design levels which represent distinct activities that support the design process as shown in figure 6.1. The computational requirements described below are based on the identified computational activities in the Preliminary Design Levels (Levels II, III, IV, and V), together with an estimate of the computational activities in the Product Levels (Levels VI, VII, VIII, and IX).

6.2.1 Project 1 - Level II Computational Requirements

A. Purpose

The purpose of Level II is to select the design mission requirements that the candidate airplanes defined and analyzed in the succeeding design levels will be required to satisfy.

B. User Environment

The user environment for Level II is expected to have the following characteristics.

1. The primary users of Level II will be the Marketing and Preliminary Design groups.

2. In order to be responsive to the needs of the users, the desired minimum computational flow time required to execute the Level II analysis is one half hour.

3. The user groups would use the Level II analysis to execute approximately 30 design mission selections during two thirds of the Preliminary Design Phase.

4. The user groups should have 10 runs accessible on-line.

5. Most of the runs in Level II will be single executions; although within the Level II analysis, approximately 50 airplanes will be generated to satisfy each design mission.

C. Computational Size

Level II is comprised of the activities listed in table 6.1. These activities are described in section 4.2.3.2 of Volume II, and the estimates of the computational size of each activity are based on the Technical Program Element descriptions in Volume V. With allowances for recycling within the Level II analysis, to select the design mission will require 1700 CP seconds (on the CDC 6600). The Technical Program Elements which are required to perform the Level II analysis consist of approximately 35 boxes of source cards.

The input to Level II is on the order of 1000 data words together with marketing and economic data stored in the data base. The output from Level II is on the order of 10,000 data words. The largest activity within Level II has approximately 20,000 words of input and output. The average data transfer rate is on the order of 100,000 words in 1700 CP seconds.

D. Execution Frequency and Computational Load

Projecting the Level II activity to usage on a continual basis would result in the following computational load on the IPAD system by each user group:

$$\left(\frac{30 \text{ EXECUTIONS}}{\text{PD PHASE}}\right)\left(\frac{1 \text{ PD PHASE}}{\text{YEAR}}\right)\left(\frac{1 \text{ YEAR}}{280 \text{ DAYS}}\right)\left(\frac{8 \text{ MONTHS LEVEL TIME}}{12 \text{ MONTHS PROJECT TIME}}\right)$$
$$\text{X (1 USER GROUP)} = .07 \frac{\text{GROUP EXECUTIONS}}{\text{DAY}}$$

$$\left(\frac{.07 \text{ GROUP EXECUTIONS}}{\text{DAY}}\right)\left(\frac{.47 \text{ CP HOURS}}{\text{EXECUTION}}\right)(1.3 \text{ RERUN FACTOR})$$
$$= .043 \text{ CP } \frac{\text{HOURS}}{\text{DAY}} \text{ (ON THE CDC 6600)}$$

E. Computational Flowtime

Based on the Technical Program Elements which are candidates for first implementation of IPAD, an estimate of the minimum computational flowtime required to execute a run through the Level II

Table 6.1 Computational Activity Flow — Project I — Design Level II

Activity ID No.	Activity CP Sec*	Activity Input/Output (10^x)	Iterations	Source Cards**
II-2	216	4		2
II-3	50	3		1
II-3	50	3		1
II-5	58	4		28
II-6	150	3		2
II-7	432	4		1
II-3	50	3		
II-3	50	3		
II-5	58	4		
II-6	150	3		
II-7	432	4		
	1696 (.47 Hours)	10^4		35

* CDC 6600
** Boxes

analysis is approximately two days. The Level II analysis contains a "thumbprint" analysis which identifies approximately 50 candidate airplanes which will satisfy each design mission. At present, it requires a man decision to select which airplane configurations will be pursued in the analysis.

F. Man/Machine Interfaces

The types of man/machine interfaces desired for Level II would be:

1. Interactive text scopes to display and enter data.

2. On-line x-y plotters to plot most graphical output.

3. Drafting plotters to plot selected graphical output;

4. Printers to print document quality output.

5. Card readers to input data.

6.2.2 Project 1 - Level III Computational Requirements

A. Purpose

The purpose of Level III is to size an airplane which will perform the required design mission.

B. User Environment

The user environment for Level III is expected to have the following characteristics.

1. The primary users of Level III will be the Preliminary Design groups.

2. In order to be responsive to the needs of the users, the desired minimum computational flowtime required to execute the Level III analysis is one half day.

3. The user groups would use the Level III analysis to execute approximately 100 configuration sizing

runs during two thirds of the Preliminary Design phase.

4. The user groups should have 10 runs accessible on-line.

5. Most of the runs in Level III will be single executions, although occasionally there will be multiple configurations which are executed in series (trade studies). It is expected that configuration optimization type of runs will be limited to the first half of Level III "Geometry Sizing" and that suboptimization within particular activities will be accomplished during the second half of Level III "Structure Sizing."

C. Computational Size

Level III is comprised of the activities listed in Table 6.2. These activities are described in section 4.2.3.3 of Volume II, and the estimates of the computational size of each activity are based on the Technical Program Element descriptions in Volume V. With allowances for recycling within the Level III analysis, to size the configuration's geometry and structure will require 4300 CP seconds (on the CDC 6600). The Technical Program Elements which are required to perform the Level III analysis consist of approximately 170 boxes of source cards.

The input to Level III is on the order of 10^4 data words, and the output from Level III is on the order of 10^5 data words. The largest activity within Level III has on the order of 10^6 data words of input and output. The average data transfer rate is on the order of 10^7 data words in 4300 CP seconds.

D. Execution Frequency and Computational Load

Projecting the Level III activity to usage on a continual basis would result in the following computational load on the IPAD system by each user group:

Table 6.2 Computational Activity Flow
Project I—Design Level III

Activity ID No.	Activity CP Sec*	Activity Input/Output (10^x)	Iterations	Source Cards**
III-1				
III-2	10	3		11
III-3	29	4	3	15
III-5	3	3		8
III-6	20	3		2
III-9	29	4		11
III-2	10	3		
III-3	29	4	2	
III-5	3	3		
III-6	20	3		
III-9	29	4		
III-2	10	3		
III-3	29	4	2	
III-5	3	3		
III-6	20	3		
III-9	29	4		
III-11	50	3		5
III-12	8	3		3
III-13	2	3		3
III-14	350 (total)	6	4	10
III-15	350 (total)	6		10
III-17	352	6		78
III-14	125 (total)	6	2	
III-15	125 (total)	6		
III-17	352	6		
III-2	10	3		
III-3	29	4	2	
III-7	3	3		
III-6	20	3		
III-9	29	4		
III-2	10	3		
III-3	29	4		
III-7	3	3		
III-6	20	3		
III-9	29	4		
III-2	10	3		
III-3	29	4		
III-7	3	3		
III-6	20	3		
III-9	29	4		
III-11	50	3		
III-12	8	3		
III-13	2	3		
III-14	350 (total)	6	4	
III-15	350 (total)	6		
III-17	352	6		
III-14	125 (total)	6	2	
III-15	125 (total)	6		
III-21	200	5		10
III-22	45	4		2
III-23	3	2		4
III-24	100	3		
	4280	10^6		169
	(1.19 Hours)			

* CDC 6600

** Boxes

$$\left(\frac{100 \text{ EXECUTIONS}}{\text{PD PHASE}}\right)\left(\frac{1 \text{ PD PHASE}}{\text{YEAR}}\right)\left(\frac{1 \text{ YEAR}}{280 \text{ DAYS}}\right)\left(\frac{8 \text{ MONTHS LEVEL TIME}}{12 \text{ MONTHS PROJECT TIME}}\right)$$

$$\times\ (1 \text{ USER GROUP}) = .24 \ \frac{\text{GROUP EXECUTIONS}}{\text{DAY}}$$

$$\left(\frac{.24 \text{ GROUP EXECUTIONS}}{\text{DAY}}\right)\left(\frac{1.19 \text{ CP HOURS}}{\text{EXECUTION}}\right)\left(1.3 \text{ RERUN FACTOR}\right)$$

$$= .37 \text{ CP } \frac{\text{HOURS}}{\text{DAY}} \quad (\text{ON THE CDC 6600})$$

E. Computational Flow Time

Based on the Technical Program Elements which are candidates for first implementation of IPAD, an estimate of the minimum computational flowtime required to execute a run through the Level III analysis is approximately two and one half weeks.

F. Man/Machine Interfaces

The types of man/machine interfaces desired for Level III would be:

1. Graphics scopes to display and enter data;
2. Interactive text scopes to display and enter data;
3. On-line x-y plotters to plot most graphical output;
4. Drafting plotters to plot selected graphical output;
5. Printers to print document quality output;
6. Card readers to input data.

6.2.3 Project 1 - Level IV Computational Requirements

A. Purpose

The purpose of Level IV is to refine the configuration determined in Level III.

B. User Environment

The user environment for Level IV is expected to have the following characteristics.

1. The primary users of Level IV will be the Technology Staff groups and to some extent the Preliminary Design groups.

2. In order to be responsive to the needs of the users, the desired minimum computational flowtime required to execute a configuration once through the Level IV analysis is one week.

3. The user groups would use the Level IV analysis to execute approximately 30 configuration refining runs during two thirds of the Preliminary Design phase.

4. The user groups should have 4 runs accessible on-line.

5. Since the Level IV analysis is quite long and complex, most of the runs using the analysis will be limited to having each technical discipline execute their own analyses using the latest information in the data base, and update the information in the data base with the results of their analyses. Therefore, even though the Level IV Detailed Design Network (Section 4.2.2.1 of Volume II) indicates that Level IV is a serial analysis, most of the time, the activities will be executed in parallel. The exceptions to this will be that Level IV analysis may be executed in total once in the beginning of the Level IV phase, and once in total after all the technical disciplines are satisified with their individual analyses of the Level IV configurations. Optimization type of runs will be limited to suboptimization within particular activities.

C. Computational Size

Level IV is comprised of the activities listed in Table 6.3. The activities are described in section 4.2.3.4, and the estimates of the computational size of each activity are based on the Technical Program Element Descriptions in Appendix A. To make one complete execution through all the options within the

Table 6.3 Computational Activity Flow
Project I — Design Level IV

Activity ID No	Activity CP Sec*	Activity Input/Output (10^x)	Iterations	Source Cards**
IV-1				
IV-2	614	3		7
IV-4	1000	3		14
IV-5	60	5		156 (total)
IV-4	200	3		
IV-10	1000	6		30
IV-9	600	4		
IV-12	253	4		26
IV-15A	13	3		16
IV-17	1040	4		30
IV-18	244	5		
IV-20	350 (total)	6	3	8
VI-21	5400	6		7
IV-23	100	5		
IV-24	8	3		
IV-20	350 (total)	6	3	
IV-21	5400	6		
IV-23	100	5		
IV-25	100	5		
IV-20	350 (total)	6	3	
IV-21	5400	6		
IV-23	100	5		
IV-25	100	5		
EM-1	10	5		
EM-2	500	6		78
EM-3	1230	3		18
EM-4	1000	4		7
EM-5	147	5		1
EM-7	18	4		2
EM-8	200	5		10
EM-9	75	4		3
EM-10 }	600	6		5
EM-11 }				
IV-28	200	6		
IV-31	40	5	10	
IV-29	250	4		24
IV-30	20	5		1
IV-28	200	6		
IV-31	40	5		
IV-35	5	2		
IV-38	200	5		
IV-41	200	6		
IV-42	480	5		9
IV-43	1800	6		
IV-45	350	5		
IV-46	5400	6		
IV-48	200	5		
IV-42	480	5		
IV-43	1800	6		
IV-45	350	5		
IV-46	5400	6		
IV-48	200	5		
IV-28	200	6		
IV-31	40	5		
IV-29	250	4		
IV-30	20	5		
IV-28	200	6		
IV-31	40	5		
IV-35	5	2		
IV-38	200	5		
IV-52	200	6		
IV-53	255	4		
IV-42	480	5		
IV-43	1800	6		
IV-45	350	5		
IV-46	5400	6		
IV-48	200	5		
IV-42	480	5		
IV-43	1800	6		
IV-48	200	5		
IV-52	200	6		
IV-53	255	4		
IV-4	1000	3		
IV-10	1000	6		
IV-9	600	4		
IV-15A	13	3		
IV-17	1040	4		
IV-18	244	5		
IV-56	200	3		
	88439	10^6		453
	(24 56 Hours)			

* CDC 6600

** Boxes

Level IV analysis will require 88400 CP seconds (on the CDC 6600) in a batch mode. Under normal circumstances, the computational time will be considerably longer because of the necessity of having to go back and re-execute previous levels, and because there will be a considerable amount of on-line, interactive activity required to support the Level IV analyses. The Technical Program Elements which are required to perform the Level IV analysis consist of approximately 450 boxes of source cards.

The input to Level IV is on the order of 10^5 data words, and the output from Level IV is on the order of 10^6 data words. The largest activity within Level IV has approximately 3.2 million data words of input and output. The average data transfer rate is on the order of 10^8 data words in 88400 CP seconds.

D. Execution Frequency and Computational Load

Projecting the Level IV activity to usage on a continual basis would result in the following computational load on the IPAD system by each user group:

$$\left(\frac{30\ \text{EXECUTIONS}}{\text{PD PHASE}}\right)\left(\frac{1\ \text{PD PHASE}}{\text{YEAR}}\right)\left(\frac{1\ \text{YEAR}}{280\ \text{DAYS}}\right)\left(\frac{8\ \text{MONTHS LEVEL TIME}}{12\ \text{MONTHS PROJECT TIME}}\right)$$

$$\times\ (1\ \text{USER GROUP}) = .06\ \frac{\text{GROUP EXECUTIONS}}{\text{DAY}}$$

$$\left(\frac{.06\ \text{GROUP EXECUTIONS}}{\text{DAY}}\right)\left(\frac{24.56\ \text{CP HOURS}}{\text{EXECUTIONS}}\right)(1.3\ \text{RERUN FACTOR})$$

$$= 2.01\ \text{CP}\ \frac{\text{HOURS}}{\text{DAY}}\ (\text{ON THE CDC 6600})$$

E. Computational Flow Time

Based on the Technical Program Elements which are candidates for first implementation of IPAD, an estimate of the minimum computational flowtime required to execute a run through the Level IV analysis is approximately one month for one design cycle and two months to complete all the iterations required for a converged design.

F. Man/Machine Interfaces

The types of man/machine interfaces desired for Level IV would be:

1. Graphics scopes to display and enter data;

2. Interactive text scopes to display and enter data;

3. On-line x-y plotters to plot most graphical output;

4. Drafting plotters to plot selected graphical output;

5. Printers to print document quality output;

6. Card readers to input data.

6.2.4 Project 1 - Level V Computational Requirements

A. Purpose

The purpose of Level V is to verify the configuration established in the previous levels by conducting wind tunnel tests and more refined analyses on the configuration's components.

B. User Environment

The user environment for Level V is expected to have the following characteristics.

1. The primary users of the Level V will be the Technology Staff groups and the Preliminary Design groups.

2. In order to be responsive to the needs of the users, the desired minimum computational flowtime required to execute a configuration through the Level V analysis is one month.

3. The user groups would use the Level V analysis to execute approximately 8 configuration verification exercises during two thirds of the Preliminary Design phase.

4. The user groups should have 2 runs accessible on-line.

5. Since the Level V is very long and complex, it will be used in a parallel manner in which each technology discipline will execute its own analyses using the latest information in the data base and update the information in the data base with the results of their analyses. Therefore Level V will never be executed in its entirety at any one time, but instead, will consist of parallel, partial executions. Optimization type of runs will be limited to suboptimization within particular activities.

C. Computational Size

Since most of the activities in the Level V analysis will be performed in a parallel sequence--rather than in a serial sequence which existed in the previous levels, it is not feasible to develop the computational activity flow in the same manner as was done for the previous levels. Instead, each technical discipline was asked to estimate the computational size of their Level V analyses as a whole, rather than for each individual activity. Consequently, the Level V computational activity sizes shown in Table 6.4 represent each technology discipline's estimate of the effort required to support the Level V analysis for eight configurations over an eight month period. The total estimated CP time for Level V is 294 hours (on the CDC 6600). The Technical Program Elements which are required to perform the Level V analysis consist of approximately 560 boxes of source cards.

The input to Level V is on the order of 10^6 data words, and the output from Level V is on the order of 10^8 data words. The largest technical discipline's effort consists of 130 million data words of input and output. The average data transfer rate is 2.9×10^8 data words in 1×10^6 seconds.

D. Executation Frequency and Computational Load

Projecting the Level V activity to usage on a continual basis would result in the following computational load on the IPAD system by each user group:

Table 6.4 Computational Activity Sizes by Technical Discipline
Project I—Design Level V

Technology	CP Sec* $(x10^3)$	Source Cards (Boxes)	Input/Output (Words-10^x)
Aerodynamics	147	72	5
Flight Controls	57	15	8
Propulsion	1	5	4
Loads	33	50	6
Stress	560	72	8
Flutter	28	8	7
Weights	44	250	7
Design	93	40	7
Other	96	51	7
	1059	563	10^8

$$\text{CP Time/Configuration} = \frac{1059000}{(3600)(8)} = 36.77 \text{ Hours}$$

* CDC 6600

$$\left(\frac{8\ \text{EXECUTIONS}}{\text{PD PHASE}}\right)\left(\frac{1\ \text{PD PHASE}}{\text{YEAR}}\right)\left(\frac{1\ \text{YEAR}}{280\ \text{DAYS}}\right)\left(\frac{8\ \text{MONTHS LEVEL TIME}}{12\ \text{MONTHS PROJECT TIME}}\right)$$

$$\times\ (1\ \text{USER GROUP}) = .019\ \frac{\text{GROUP EXECUTIONS}}{\text{DAY}}$$

$$\left(\frac{.019\ \text{GROUP EXECUTIONS}}{\text{DAY}}\right)\left(\frac{36.77\ \text{CP HOURS}}{\text{EXECUTION}}\right)(1.3\ \text{RERUN FACTOR})$$

$$= .89\ \text{CP}\ \frac{\text{HOURS}}{\text{DAY}}\quad (\text{ON THE CDC 6600})$$

E. Computational Flowtime

Because of the complex nature of the Level V analysis, most of the technical discipline's activities will be done in parallel. This parallel operation may have serious implications for handling the updating of information in the data base and the concept of a "job" within the context of the IPAD Paper Design. The implications are that each technology discipline will want to execute their activities (jobs) with the "most current, official" information in the data base and will want to keep re-executing their jobs as long as information pertinent to their analysis continues to change. Therefore, no group is finished executing their jobs until every group is finished, because most of the activities are interdependent. The minimum computational flowtime (excluding wind tunnel activities) to cycle one configuration through the Level V analysis is one and one half months for one design cycle and three months to complete all the iterations required for a converged design.

F. Man/Machine Interfaces

The types of man/machine interfaces desired for Level V would be:

1. Hybrid simulators to evaluate airplane handling qualities;

2. Interactive graphics scopes to display and edit data;

3. Graphics scopes to display and enter data;

4. Interactive text scopes to display and enter data;

5. On-line x-y plotters to plot most graphical output;

6. Drafting plotters to plot selected graphical output;

7. Printers to print document quality output;

8. Card readers to input data.

6.2.5 Project 1 - Level VI-IX Computational Requirements

A. Purpose

Levels VI through IX are the product levels in which the functions of Product Detail Design (Level VI), Product Manufacture (Level VII), Product Verification (Level VIII), and Product Support (Level IX) are performed.

B. User Environment

The user environment for Levels VI through IX is expected to have the following characteristics.

1. The primary users of Levels VI through IX will be the Product Design groups with support from the Technology Staff groups primarily in Level VI.

2. The user groups will be using IPAD throughout the duration of the product levels which is expected to be an active design period of two years after which a sustaining effort will last through the life of the product.

3. The activities within Level VI are expected to be similar to those within Level V in which there will be parallel executions of detail design and analysis activities by each technical discipline using the latest information in the data base and updating the data base with the results of their work. Therefore, Level VI will never be executed in its entirety at any one point in time. Optimization type of runs will be limited to suboptimization within particular activities.

Beyond Level VI, the primary usage of the IPAD system would be to receive, store, and disseminate information using the IPAD data management system and the data base.

C. Computational Size

Since the design and analysis activities in Levels VI through IX are executed in a similar manner to those in Level V, in a parallel, independent sequence, a representative of each technical discipline was asked to estimate the computational size of his analyses within Levels IV through IX as a whole. The computational activity sizes shown in table 6.5 represent each technical discipline's estimate of the effort required to support the product levels over a two year period. The total estimated CP time for Levels IV through IX is 1076 hours (on the CDC 6600). The Technical Program Elements which are required to perform the analyses in Levels VI through IX consist of approximately 720 boxes of source cards.

The input to Level IV is on the order of 10^8 data words, and the output from Levels IV through IX is on the order of 10^9 data words.

D. Execution Frequency and Computational Load

Projecting the computational activities within Levels VI through IX to usage on a continual basis would result in the following computational load on the IPAD system by each user group:

$$\left(\frac{\text{1 EXECUTION}}{\text{PRODUCT PHASE}}\right)\left(\frac{\text{.5 PRODUCT PHASE}}{\text{YEAR}}\right)\left(\frac{\text{1 YEAR}}{\text{280 DAYS}}\right)\left(\frac{\text{24 MONTHS LEVEL TIME}}{\text{24 MONTHS PROJECT TIME}}\right)$$

$$\times\ (\text{1 USER GROUP}) = .0018\ \frac{\text{GROUP EXECUTIONS}}{\text{DAY}}$$

$$\left(\frac{\text{.0018 GROUP EXECUTIONS}}{\text{DAY}}\right)\left(\frac{\text{1076 CP HOURS}}{\text{EXECUTION}}\right)(\text{1.3 RERUN FACTOR})$$

$$= 2.50\ \text{CP}\ \frac{\text{HOURS}}{\text{DAY}}\quad (\text{ON THE CDC 6600})$$

Table 6.5 Computational Activity Sizes by Technical Discipline
Project I – Design Level VI – IX

Technology	CP Sec* ($x10^3$)	Source Cards (Boxes)	Input/Output (Words-10^x)
Aerodynamics	400	100	5
Flight Controls	114	30	8
Propulsion	4	10	4
Loads	100	50	7
Stress	840	100	9
Flutter	28	10	7
Weights	176	250	8
Design	1860	100	9
Other	352	65	8
	3874	715	10^9

$$\text{CP Time/Configuration} = \frac{3874000}{3600} = 1076.11 \text{ Hours}$$

* CDC 6600

E. Computational Flowtime

Because of the complex nature of the design and analysis activities within the product levels, most of the technical discipline's activities will be done in parallel over a long period of time. Beyond doing the required amount of detailed design and analysis to obtain an airplane which can be manufactured, certified and put into service, additonal effort will be spent on solving problems, design improvements, customer variations, and keeping information current. This means that the computational flowtime for product levels will last from the product go-ahead to the end of the program.

F. Man/Machine Interfaces

The types of man/machine interfaces desired for Levels VI through IX would be:

1. Hybrid simulators to evaluate airplane handling qualities;

2. Interactive graphics scopes to display and edit data;

3. Graphics scopes to display and enter data;

4. Interactive text scopes to display and enter data;

5. On-line x-y plotters to plot most graphical output;

6. Drafting plotters to plot selected graphical output;

7. Printers to print document quality output;

8. Card readers to input data.

6.3 PROJECT 2 - SUPERSONIC COMMERCIAL TRANSPORT

Design Project 2 was chosen to be the design of a supersonic commercial transport. The Design Project 2 was divided into design levels which represent distinct activities that support the design process as shown in figure 6.1. The computational requirements described below are based on the identified computational activities in the Preliminary Design Levels (Levels II, III, IV, and V), together with an estimate of the computational activities in the Product Levels (Levels VI, VII, VIII, and IX).

The information presented below for Project 2 is very similar to the information presented for Project 1 except that some of the values of the computational requirements parameters are different.

6.3.1 Project 2 - Level II Computational Requirements

A. Purpose

The purpose of Level II is to select the design mission requirements that the candidate airplanes defined and analyzed in the succeeding design levels will be required to satisfy.

B. User Environment

The user environment for Level II is expected to have the following characteristics.

1. The primary users of Level II will be the Marketing and Preliminary Design groups.
2. In order to be responsive to the needs of the users, the desired minimum computational flow time required to execute the Level II analysis is one half hour.
3. The user groups would use the Level II analysis to execute approximately 45 design mission selections during almost one half of the Preliminary Design Phase.
4. The user groups should have 10 runs accessible on-line.
5. Most of the runs in Level II will be single executions; although within the Level II analysis, approximately 50 airplanes will be generated to satisfy each design mission.

C. Computational Size

Level II is comprised of the activities listed in table 6.6. These activities are described in section 4.3.3.2 of Volume II, and the estimates of the computational size of each activity are based on the Technical Program Element descriptions in Volume V. With allowances for recycling within the Level II analysis, to select the design mission will require 1700 CP seconds (on the CDC 6600). The Technical Program Elements which are required to perform the Level II analysis consist of approximately 35 boxes of source cards.

The input to Level II is on the order of 1000 data words together with marketing and economic data stored in the data base. The output from Level II is on the order of 10,000 data words. The largest activity within Level II has approximately 20,000 words of input and output. The average data transfer rate is on the order of 100,000 words in 1700 CP seconds.

D. Execution Frequency and Computational Load

Projecting the Level II activity to' usage on a continual basis would result in the following computational load on the IPAD System by each user group:

$$\left(\frac{45 \text{ EXECUTIONS}}{\text{PD PHASE}}\right)\left(\frac{.5 \text{ PD PHASE}}{\text{YEAR}}\right)\left(\frac{1 \text{ YEAR}}{280 \text{ DAYS}}\right)\left(\frac{10 \text{ MONTHS LEVEL TIME}}{24 \text{ MONTHS PROJECT TIME}}\right)$$

$$\times \text{ (1 USER GROUP)} = .033 \ \frac{\text{GROUP EXECUTIONS}}{\text{DAY}}$$

$$\left(\frac{.07 \text{ GROUP EXECUTIONS}}{\text{DAY}}\right)\left(\frac{.47 \text{ CP HOURS}}{\text{EXECUTION}}\right)\left(1.3 \text{ RERUN FACTOR}\right)$$

$$= .020 \text{ CP } \frac{\text{HOURS}}{\text{DAY}} \quad \text{(ON THE CDC 6600)}$$

E. Computational Flowtime

Based on the Technical Program Elements which are candidates for first implementation of IPAD, an estimate of the minimum computational flowtime required to execute a run through the Level II

Table 6.6 Computational Activity Flow — Project 2 — Design Level II

Activity ID No.	Activity CP Sec *	Activity Input/Output (10^x)	Iterations	Source Cards**
II-2	216	4		2
II-3	50	3		1
II-3	50	3		1
II-5	58	4		28
II-6	150	3		2
II-7	432	4		1
II-3	50	3		
II-3	50	3		
II-5	58	4		
II-6	150	3		
II-7	432	4		
	1696] (.47 Hours)	10^4		35

* CDC 6600

** Boxes

analysis is approximately two days. The Level II analysis contains a "thumbprint" analysis which identifies approximately 50 candidate airplanes which will satisfy each design mission. At present, it requires a man decision to select which airplane configurations will be pursued in the analysis.

F. Man/Machine Interfaces

The types of man/machine interfaces desired for Level II would be:

1. Interactive text scopes to display and enter data;

2. On-line x-y plotters to plot most graphical output;

3. Drafting plotters to plot selected graphical output;

4. Printers to print document quality output;

5. Card readers to input data.

6.3.2 Project 2 - Level III Computational Requirements

A. Purpose

The purpose of Level III is to size an airplane which will perform the required design mission.

B. User Environment

The user environment for Level III is expected to have the following characteristics.

1. The primary users of Level III will be the Preliminary Design groups.

2. In order to be responsive to the needs of the users, the desired minimum computational flowtime required to execute the Level III analysis is two days.

3. The user groups would use the Level III analysis to execute approximately 115 configuration sizing

runs during almost one half of the Preliminary Design Phase.

4. The user groups should have 10 runs accessible on-line.

5. Most of the runs in Level III will be single executions; although occasionally there will be multiple configurations which are executed in series (trade studies). It is expected that configuration optimization type of runs will be limited to the first half of Level III "Geometry Sizing" and that suboptimization within particular activities will be accomplished during the second half of Level III "Structure Sizing."

C. Computational Size

Level III is comprised of the activities listed in table 6.7. These activities are described in section 4.3.3.3 of Volume II, and the estimates of the computational size of each activity are based on the Technical Program Element descriptions in Volume V. With allowances for recycling within the Level III analysis, to size the configuration's geometry and structure will require 75000 CP seconds (on the CDC 6600). The Technical Program Elements which are required to perform the Level III analysis consist of approximately 335 boxes of source cards.

The input to Level III is on the order of 10^4 data words, and the output from Level III is on the order of 10^6 data words. The largest activity within Level III has on the order of 10^6 data words of input and output. The average data transfer rate is on the order of 10^7 data words in 75000 CP seconds.

D. Execution Frequency and Computational Load

Projecting the Level III activity to usage on a continual basis would result in the following computational load on the IPAD system by each user group:

Table 6.7 Computational Activity Flow — Project 2 — Design Level III

Activity ID No.	CP Sec *	Input/Output (10^x)	Iter-ations	Source Cards**
III-2	10	3		11
III-3	420	4	3	47
III-5	3	3		8
III-6	66	3		5
III-9	29	4		11
III-2	10	3		
III-3	420	4	2	
III-5	3	3		
III-6	66	3		
III-9	29	4		
III-2	10	3		
III-3	420	4		
III-5	3	3	2	
III-6	66	3		
III-9	29	4		
III-11	50	3		
III-12	8	3		5
III-13	8	4		6
III-14	500	5		10
III-15	100	5		10
III-16	2800	6		80 ▷2
III-14	500	5		
III-15	60	5	2	
III-16	2000	6		
III-18	66	3		
III-19	446	6		85
III-14	500	5		
III-15	60	5	2	
III-16	2000	6		
III-18	66	3		
III-19	446	6		
III-2	10	3		
III-3	420	4	2	
III-7	3	3		
III-6	66	3		
III-9	29	4		
III-2	10	3		
III-3	420	4		
III-7	3	3		
III-6	66	3		
III-9	29	4		
III-2	10	3		
III-3	420	4		
III-7	3	3		
III-6	66	3		
III-9	29	4		

Activity ID No.	CP Sec *	Input/Output (10^x)	Iter-ations	Source Cards**
III-11	50	3		
III-12	8	3		
III-13	8	4		
III-14	500	5		
III-15	60	5	3	
III-16	2000	6		
III-18	66	3		
III-19	446	6		
III-14	500	5		
III-15	60	5	2	
III-16	2000	6		
III-18	66	3		
III-19	446	6		
III-23	2257/225 ▷1	6		35
III-24	45	4	10	2
III-26	30	5		2
III-2	10	3		
III-3	420	4		
III-7	3	3	2	
III-6	66	3		
III-9	29	4		
III-2	10	3		
III-3	420	4	2	
III-7	3	3		
III-6	66	3		
III-9	29	4		
III-11	50	3		
III-12	8	3		
III-13	8	4		
III-14	500	5		
III-15	60	5	3	
III-16	2000	6		
III-18	66	3		
III-19	446	6		
III-14	500	5		
III-15	60	5	2	
III-16	2000	6		
III-18	66	3		
III-19	446	6		
III-23	1027/225 ▷1	6		
III-24	45	4	5	
III-26	30	5		
III-12	8	3		
III-13	8	4		
III-14	500	5		
III-15	60	5	3	
III-16	2000	6		
III-18	66	3		
III-19	446	6		
III-14	500	5		
III-15	60	5	2	
III-16	2000	6		
III-18	66	3		
III-19	446	6		
III-23	1027/225 ▷1	6		
III-24	45	4	5	
III-26	30	5		
III-31	200	6		18
III-32	600	3		
III-33	100	3		
	74632 (20.73 Hours)	10^6		335

▷1 First Iteration / All Other Iterations

▷2 1st Stress Cycle

* CDC 6600

** Boxes

$$\left(\frac{115 \text{ EXECUTIONS}}{\text{PD PHASE}}\right)\left(\frac{.5 \text{ PD PHASE}}{\text{YEAR}}\right)\left(\frac{1 \text{ YEAR}}{280 \text{ DAYS}}\right)\left(\frac{10 \text{ MONTHS LEVEL TIME}}{24 \text{ MONTHS PROJECT TIME}}\right)$$

$$\times \text{ (1 USER GROUP)} = .086 \ \frac{\text{GROUP EXECUTIONS}}{\text{DAY}}$$

$$\left(\frac{.086 \text{ GROUP EXECUTIONS}}{\text{DAY}}\right)\left(\frac{20.73 \text{ CP HOURS}}{\text{EXECUTION}}\right)\text{(1.3 RERUN FACTOR)}$$

$$= 2.32 \text{ CP } \frac{\text{HOURS}}{\text{DAY}} \quad \text{(ON THE CDC 6600)}$$

E. Computational Flow Time

Based on the Technical Program Elements which are candidates for first implementation of IPAD, an estimate of the minimum computational flowtime required to execute a run through the Level III analysis is approximately one month. The reason that the computational flowtime is greater than Project 1 Level III is the requirement for finite element structural analysis and the attempt to solve configuration flutter problems, instead of assessing flutter penalties as will be done in Project 1 Level III.

F. Man/Machine Interfaces

The types of man/machine interfaces desired for Level III would be:

1. Graphics scopes to display and enter data;

2. Interactive text scopes to display and enter data;

3. On-line x-y plotters to plot most graphical output;

4. Drafting plotters to plot selected graphical output;

5. Printers to print document quality output;

6. Card readers to input data.

6.3.3 Project 2 - Level IV Computational Requirements

A. Purpose

The purpose of Level IV is to refine the configuration determined in Level III.

B. User Environment

The user environment for Level IV is expected to have the following characteristics.

1. The primary users of Level IV will be the Technology Staff groups and to some extent the Preliminary Design groups.

2. In order to be responsive to the needs of the users, the desired minimum computational flowtime required to execute a configuration once through the Level IV analysis is two weeks.

3. The user groups would use the Level IV analysis to execute approximately 40 configuration refining runs during almost one half of the Preliminary Design Phase.

4. The user groups should have 4 runs accessible on-line.

5. Since the Level IV analysis is quite long and complex, most of the runs using the analysis will be limited to having each technology discipline execute their own analyses using the latest information in the data base, and update the information in the data base with the results of their analyses. Therefore, even though the Level IV Detailed Design Network (Section 4.3.2.1 of Volume II) indicates that Level IV is a serial analysis, most of the time, the activities will be executed in parallel. The exceptions to this will be that Level IV analysis may be executed in total once in the beginning of the Level IV phase, and once in total after all the technology disciplines are satisified with their individual analyses of the Level IV configurations. Optimization type of runs will be limited to sub-optimization within particular activities.

C. Computational Size

Level IV is comprised of the activities listed in table 6.8. The activities are described in section 4.3.3.4 of Volume II, and the estimates of the computational size of each activity are based on the Technical Program Element Descriptions in Volume V. To make one complete execution through all the options within the Level IV analysis will require 60 CP hours (on the CDC 6600) in a batch mode. Under normal circumstances, the computational time will be considerably longer because of the necessity of having to go back and re-execute previous levels, and because there will be a considerable amount of on-line, interactive activity required to support the Level IV analyses. The Technical Program Elements which are required to perform the Level IV analysis consist of approximately 650 boxes of source cards.

The input to Level IV is on the order of 10^5 data words, and the output from Level IV is on the order of 10^6 data words. The average data transfer rate is on the order of 10^8 data words in 2×10^5 CP seconds.

D. Execution Frequency and Computational Load

Projecting the Level IV activity to usage on a continual basis would result in the following computational load on the IPAD system by each user group:

$$\left(\frac{40\ \text{EXECUTIONS}}{\text{PD PHASE}}\right)\left(\frac{.5\ \text{PD PHASE}}{\text{YEAR}}\right)\left(\frac{1\ \text{YEAR}}{280\ \text{DAYS}}\right)\left(\frac{10\ \text{MONTHS LEVEL TIME}}{24\ \text{MONTHS PROJECT TIME}}\right)$$

$$\times\ (1\ \text{USER GROUP}) = .03\ \frac{\text{GROUP EXECUTIONS}}{\text{DAY}}$$

$$\left(\frac{.03\ \text{GROUP EXECUTIONS}}{\text{DAY}}\right)\left(\frac{60.41\ \text{CP HOURS}}{\text{EXECUTIONS}}\right)(1.3\ \text{RERUN FACTOR})$$

$$= 2.36\ \text{CP}\ \frac{\text{HOURS}}{\text{DAY}}\quad (\text{ON THE CDC 6600})$$

E. Computational Flow Time

Based on the Technical Program Elements which are candidates for first implementation of IPAD, an

Table 6.8 Computational Activity Flow — Project 2 — Design Level IV

Activity				
ID No	CP Sec*	Input/Output (10^x)	Iterations	Source Cards**
IV-4	250	4		6
IV-14	1300	4		60
IV-5	1885	4		22
IV-6	50	2		5
IV-7	310	3		2
IV-8	60	5		200 ▷1
IV-5	1885	4		
IV-6	50	2		
IV-7	310	3		
IV-12	300	3		2
IV-5	1885	4		
IV-6	50	2		
IV-7	310	3		
IV-12	300	3		
IV-17	13	3		16
IV-19	1175	5		40
IV-22	200	4		37
IV-23	500	5		
IV-27X	200	5		5 ▷2
IV-27	20000	5		10 ▷3
IV-26	500	5		10
IV-28	11200	6		80 ▷4
IV-26	500	5 }		
IV-27	200	5 }	2	
IV-28	8000	6 }		
IV-30	422	6		
IV-31A	140	5		
IV-26	500	5 }		
IV-27	200	5 }	3	
IV-28	8000	6 }		
IV-30	422	6		
IV-32	190	6		
IV-26	500	5 }		
IV-27	200	5 }	2	
IV-28	8000	6 }		
IV-30	422	6		
IV-32	190	6		
EM-1	10	6		78
EM-2	2000	6		18
EM-3	1230	3		7
EM-4	1000	4		1
EM-5	147	5		1
EM-7	150	4		20
EM-8	400	5		3
EM-9	75	4		5
EM-10 } EM-11 }	600	6		

Activity				
ID No.	CP Sec*	Input/Output (10^x)	Iterations	Source Cards**
IV-35	500	6 }		
IV-38	45	4 }	10	
IV-36	690	3 }		
IV-37	60	4 }		
IV-45	550	5		
IV-48	500	6		
IV-49	844	5		
IV-50	200	5		
IV-51	2700	6		
IV-53	500	5		
IV-54	200	5		
IV-55	8000	6		
IV-57	550	5		
IV-49	844	5		
IV-50	200	5		
IV-51	2700	6		
IV-53	500	5		
IV-54	200	5		
IV-55	8000	6		
IV-57	550	5		
IV-35	500	6 }		
IV-38	45	4 }	5	
IV-36	690	3 }		
IV-37	60	4 }		
IV-45	550	5		
IV-62	500	6		
IV-63	1885	4		
IV-49	884	5		
IV-50	200	5		
IV-51	2700	5		
IV-53	500	5 }		
IV-54	200	5 }	3	
IV-55	8000	6 }		
IV-57	550	5		
IV-62	500	6		
IV-63	1885	4		
IV-4	250	4		
IV-5	1885	4		
IV-6	50	2		
IV-7	310	3		
IV-12	300	3		
IV-17	13	3		
IV-19	1175	5		
IV-22	2920	4		
IV-23	200	5		
IV-48	500	6		
IV-49	844	5		
IV-50	200	5		
IV-51	2700	6		
IV-57	550	5		
IV-62	500	6		
IV-63	1885	4		
IV-6	50	2		
IV-7	310	3		
IV-12	300	3		
IV-17	13	3		
IV-19	1175	5		
IV-22	2920	4		
IV-23	200	5		
IV-66	2600	4		
	217478 (60.41 Hours)	10^6		657

▷1 Total

▷2 Aerothermal Environment

▷3 Thermal Analyzer

▷4 1st Stress Cycle

* CDC 6600

** Boxes

estimate of the minimum computational flowtime required to execute a run through the Level IV analysis is approximately two months for one design cycle and four months to complete all the iterations required for a converged design.

F. Man/Machine Interfaces

The types of man/machine interfaces desired for Level IV would be:

1. Graphics scopes to display and enter data;

2. Interactive text scopes to display and enter data;

3. On-line x-y plotters to plot most graphical output;

4. Drafting plotters to plot selected graphical output;

5. Printers to print document quality output;

6. Card readers to input data.

6.3.4 Project 2 - Level V Computational Requirements

A. Purpose

The purpose of Level V is to verify the configuration established in the previous levels by conducting wind tunnel tests and more refined analyses on the configuration's components.

B. User Environment

The user environment for Level V is expected to have the following characteristics.

1. The primary users of the Level V will be the Technology Staff groups and the Preliminary Design groups.

2. In order to be responsive to the needs of the users, the desired minimum computational flowtime required to execute a configuration through the Level V analysis is two months.

3. The user groups would use the Level V analysis to execute approximately 8 configuration verification exercises during two thirds of the Preliminary Design phase.

4. The user groups should have 2 runs accessible on-line.

5. Since the Level V is very long and complex, it will be used in a parallel manner in which each technology discipline will execute their own analyses using the latest information in the data base and update the information in the data base with the results of their analyses. Therefore Level V will never be executed in its entirety at any one time, but instead, will consist of parallel, partial executions. Optimization type of runs will be limited to suboptimization within particular activities.

C. Computational Size

Since most of the activities in the Level V analysis will be perfomed in a parallel sequence--rather than in a serial sequence which existed in the previous levels, it is not feasible to develop the computational activity flow in the same manner as was done for the previous levels. Instead, each technology discipline was asked to estimate the computational size of their Level V analyses as a whole, rather than for each individual activity. Consequently, the Level V computational activity sizes shown in table 6.9 represent each technical discipline's estimate of the effort required to support the Level V analysis for eight configurations over an eight month period. The total estimated CP time for Level V is 1360 hours (on the CDC 6600). The Technical Program Elements which are required to perform the Level V analysis consist of approximately 890 boxes of source cards.

The input to Level V is on the order of 10^6 data words, and the output from Level V is on the order of 10^8 data words. The average data transfer rate is 5 x 10^8 data words in 5 x 10^6 seconds.

Table 6.9 Computational Activity Sizes by Technical Discipline
Project 2 — Design Level V

Technology	CP Sec* ($x10^3$)	Source Cards (Boxes)	Input/Output ($Words-10^x$)
Aerodynamics	300	100	6
Flight Controls	114	30	8
Propulsion	2	10	4
Loads	330	150	7
Stress	3400	200	8
Flutter	36	8	7
Weights	88	250	7
Design	186	60	7
Other	446	81	7
	4902	889	10^8

$$\text{CP Time/Configuration} = \frac{4902000}{(3600)(8)} = 170.21 \text{ Hours}$$

* CDC 6600

D. Execution Frequency and Computational Load

Projecting the Level V activity to usage on a continual basis would result in the following computational load on the IPAD system by each user group:

$$\left(\frac{8 \text{ EXECUTIONS}}{\text{PD PHASE}}\right)\left(\frac{.5 \text{ PD PHASE}}{\text{YEAR}}\right)\left(\frac{1 \text{ YEAR}}{280 \text{ DAYS}}\right)\left(\frac{16 \text{ MONTHS LEVEL TIME}}{24 \text{ MONTHS PROJECT TIME}}\right)$$

$$\times \text{ (1 USER GROUP)} = .0095 \ \frac{\text{GROUP EXECUTIONS}}{\text{DAY}}$$

$$\left(\frac{.0095 \text{ GROUP EXECUTIONS}}{\text{DAY}}\right)\left(\frac{170.21 \text{ CP HOURS}}{\text{EXECUTION}}\right) \text{ (1.3 RERUN FACTOR)}$$

$$= 2.10 \text{ CP } \frac{\text{HOURS}}{\text{DAY}} \quad \text{(ON THE CDC 6600)}$$

E. Computational Flowtime

Because of the complex nature of the Level V analysis, most of the technical disciplines' activities will be done in parallel. This parallel operation may have serious implications for handling the updating of information in the data base and the concept of a "job" within the context of the IPAD Paper Design. The implications are that each technical discipline will want to execute their activities (jobs) with the "most current, official" information in the data base and will want to keep re-executing their jobs as long as information pertinent to their analysis continues to change. Therefore, no group is finished executing their jobs until every group is finished, because most of the activities are interdependent. The minimum computational flowtime (excluding wind tunnel activities) to cycle one configuration through the Level V analysis is three months for one design cycle and six months to complete all the iterations required for a converged design.

F. Man/Machine Interfaces

The types of man/machine interfaces desired for Level V would be:

1. Hybrid simulators to evaluate airplane handling qualities;

2. Interactive graphics scopes to display and edit data;

3. Graphics scopes to display and enter data;

4. Interactive text scopes to display and enter data;

5. On-line x-y plotters to plot most graphical output;

6. Drafting plotters to plot selected graphical output;

7. Printers to print document quality output;

8. Card readers to input data.

6.3.5 Project 2 - Level VI-IX Computational Requirements

A. Purpose

Levels VI through IX are the product levels in which the functions of Product Detail Design (Level VI), Product Manufacture (Level VII), Product Verification (Level VIII), and Product Support (Level IX) are performed.

B. User Environment

The user environment for Levels VI through IX is expected to have the following characteristics.

1. The primary users of Levels VI through IX will be the Product Design groups with support from the Technology Staff groups primarily in Level VI.

2. The user groups will be using IPAD throughout the duration of the product levels which is expected to be a period of three years.

3. The activities within Level VI are expected to be similar to those within Level V in which there will be parallel executions of detail design and analysis activities by each technical discipline using the latest information in the data base and updating the data base with the results of their work. Therefore, Level VI will never be executed in its entirety at any one point in time.

Optimization type of runs will be limited to suboptimization within particular activities. Beyond Level VI, the primary usage of the IPAD system would be to receive, store, and disseminate information using the IPAD data management system and the data base.

C. Computational Size

Since the design and analysis activities in Levels VI through IX are executed in a similar manner to those in Level V, in a parallel, independent sequence, a representative of each technical discipline was asked to estimate the computational size of their analyses within Levels IV through IX as a whole. The computational activity sizes shown in table 6.10 represent each technical discipline's estimate of the effort required to support the product levels over a three year period. The total estimated CP time for Levels IV through IX is 3557 hours (on the CDC 6600). The Technical Program Elements which are required to perform the analyses in Levels VI through IX consist of approximately 1070 boxes of source cards. The input to Level IV is on the order of 10^8 data words, and the output from Levels IV through IX is on the order of 10^9 data words.

D. Execution Frequency and Computational Load

Projecting the computational activities within Levels VI through IX to usage on a continual basis would result in the following computational load on the IPAD system by each user group:

$$\left(\frac{\text{1 EXECUTION}}{\text{PRODUCT PHASE}}\right)\left(\frac{\text{.33 PRODUCT PHASE}}{\text{YEAR}}\right)\left(\frac{\text{1 YEAR}}{\text{280 DAYS}}\right)\left(\frac{\text{36 MONTHS LEVEL TIME}}{\text{36 MONTHS PROJECT TIME}}\right)$$

$$\times \text{ (1 USER GROUP)} = .00119 \ \frac{\text{GROUP EXECUTIONS}}{\text{DAY}}$$

$$\left(\frac{\text{.00119 GROUP EXECUTIONS}}{\text{DAY}}\right)\left(\frac{\text{3557 CP HOURS}}{\text{EXECUTION}}\right)\text{(1.3 RERUN FACTOR)}$$

$$= 5.51 \text{ CP } \frac{\text{HOURS}}{\text{DAY}} \text{ (ON THE CDC 6600)}$$

Table 6.10 Computational Activity Sizes by Technical Discipline
Project 2—Design Level VI – IX

Technology	CP Sec* ($x10^3$)	Source Cards (Boxes)	Input/Output (Words-10^x)
Aerodynamics	1200	150	7
Flight Controls	228	60	8
Propulsion	8	10	5
Loads	1000	150	7
Stress	5100	200	9
Flutter	36	10	7
Weights	350	250	8
Design	3720	150	9
Other	1164	98	8
	12806	1078	10^9
CP Time/Configuration $= \frac{12806000}{3600} = 3557.22$ Hours			

* CDC 6600

E. Computational Flowtime

Because of the complex nature of the design and analysis activities within the product levels, most of the technology discipline's activities will be done in parallel over a long period of time. Beyond doing the required amount of detailed design and analysis to obtain an airplane which can be manufactured, certified and put into service, additonal effort will be spent on solving problems, design improvements, customer variations, and keeping information current. This means that the computational flowtime for product levels will last from the product go-ahead to the end of the program.

F. Man/Machine Interfaces

The types of man/machine interfaces desired for Levels VI through IX would be:

1. Hybrid simulators to evaluate airplane handling qualities;
2. Interactive graphics scopes to display and edit data;
3. Graphics scopes to display and enter data;
4. Interactive text scopes to display and enter data;
5. On-line x-y plotters to plot most graphical output;
6. Drafting plotters to plot selected graphical output;
7. Printers to print document quality output;
8. Card readers to input data.

6.4 PREDICTED TOTAL IPAD ENVIRONMENT

Using the information presented in the computational requirements section of each level of each project, it is possible to formulate what the demand on an IPAD system might be in terms of hardware capacity and performance in order to accomplish the computational tasks of a Project 1 and a Project 2 design effort.

6.4.1 Computational Load

Based on the computational requirements presented in the previous sections for each level of Project 1 and Project 2, estimates of the total computational load on an IPAD System can be formulated. These estimates are based on the computational loads generated by continual, averaged executions of each design project--rather than using the computational loads generated by executing a single design project. The reason for this is that a single execution of a design project will generate peak computational loads as shown in figure 6.3, because of the requirement that a certain number of configurations have to be executed within a certain specified time in order to complete the design project within the specified time period. The continual, averaged computational loads are based on the average executions per day for each level and the CPU time required to execute each level as summarized in figures 6.4 and 6.5 for the preliminary design levels.

Table 6.11 shows the total execution time (CDC 6600 central processor) estimated for each project. This time is based on a scenario which represents an estimate of the number of configurations to be investigated in each level during the preliminary design phase of an airplane development cycle. In addition, the table presents the computational flow time required for one design cycle and to reach a converged design solution. The computational flow time is based on the design networks in Volume II and is estimated from the—computational activities shown in Tables 6.1 through 6.9. Where experience permits, a comparative time estimate is included to show the equivalent time required for execution of similar programs using standalone batch processing.

It will be noted in Table 6.11 that the standalone time and the integrated time to develop the converged design of a subsonic airplane is shown to be unchanged by the integrated environment. This illustrates that the critical path for the development of a subsonic aircraft is not established by the calculated data required to define the aircraft. The items that establish the critical path for a well defined problem such as subsonic commercial transport are expected to remain unchanged. These include items such as customer input and coordination, development testing, management assessment of technical risk, flight testing, etc. However, for poorly defined problems such as a supersonic commercial transport, the critical path may be the development of technical data. Development cycles of 4 1/2 years for a subsonic transport and 8 years for a supersonic transport were assumed as the basis for the estimate of the

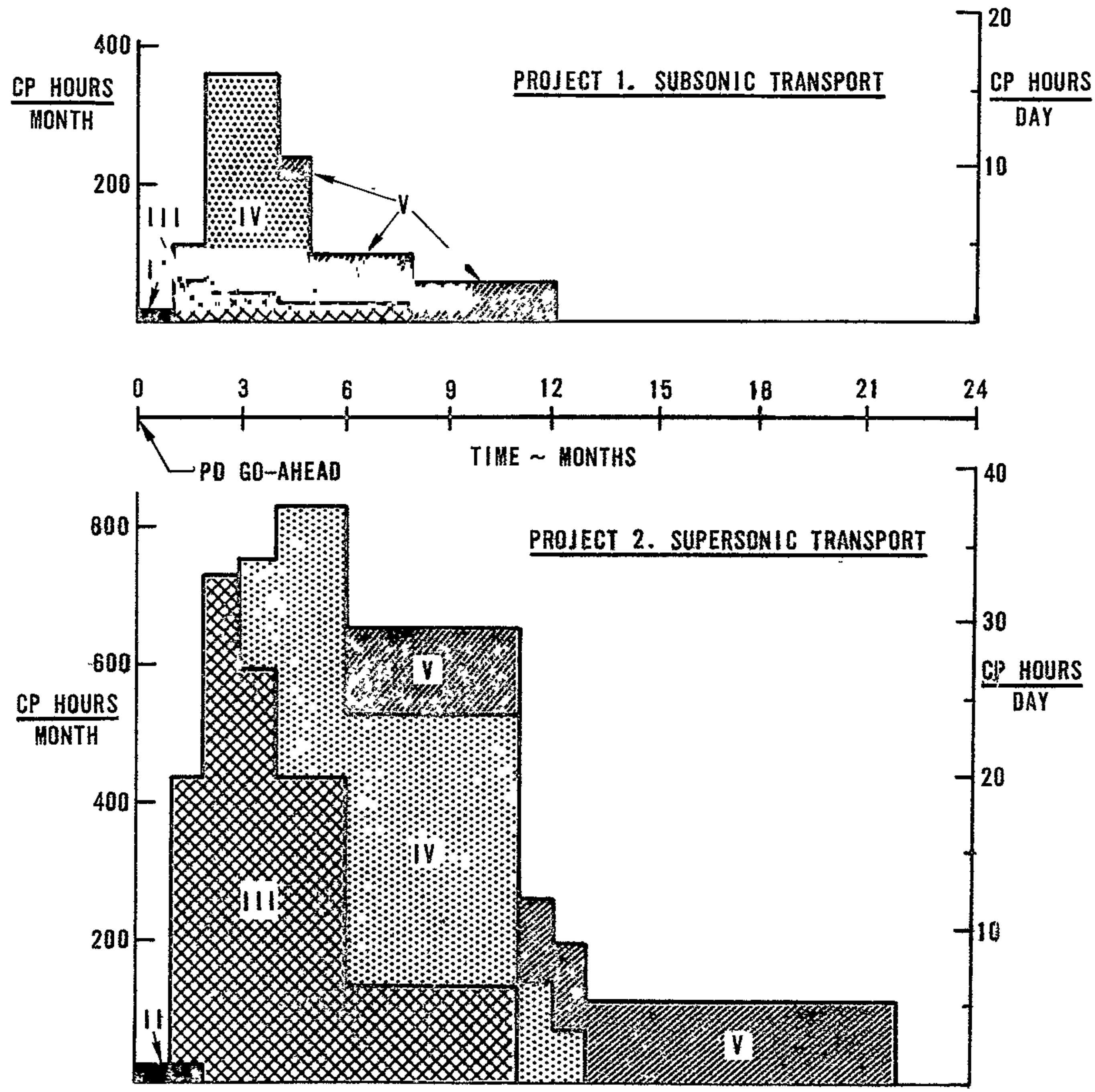


Figure 6.3 Project I and Project 2 Computational Loads for Preliminary Design Levels

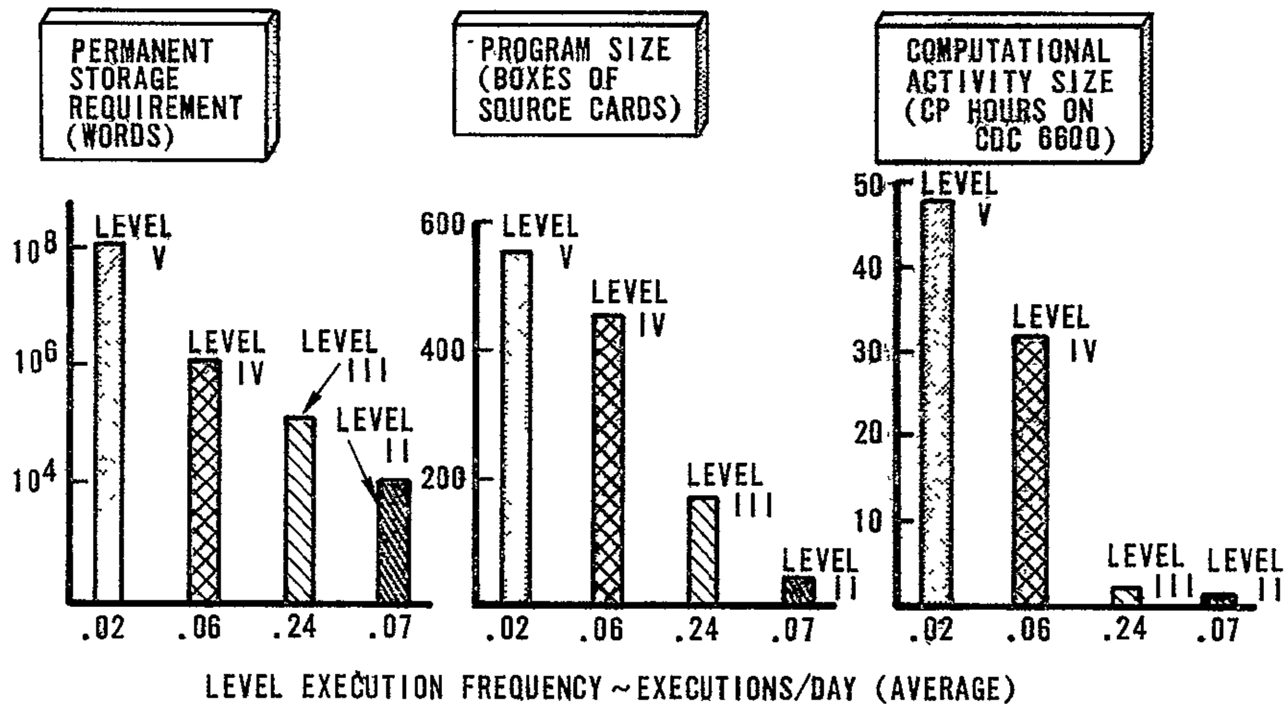


Figure 6.4 Project I — Subsonic Commercial Transport — Computational Requirements Preliminary Design Levels

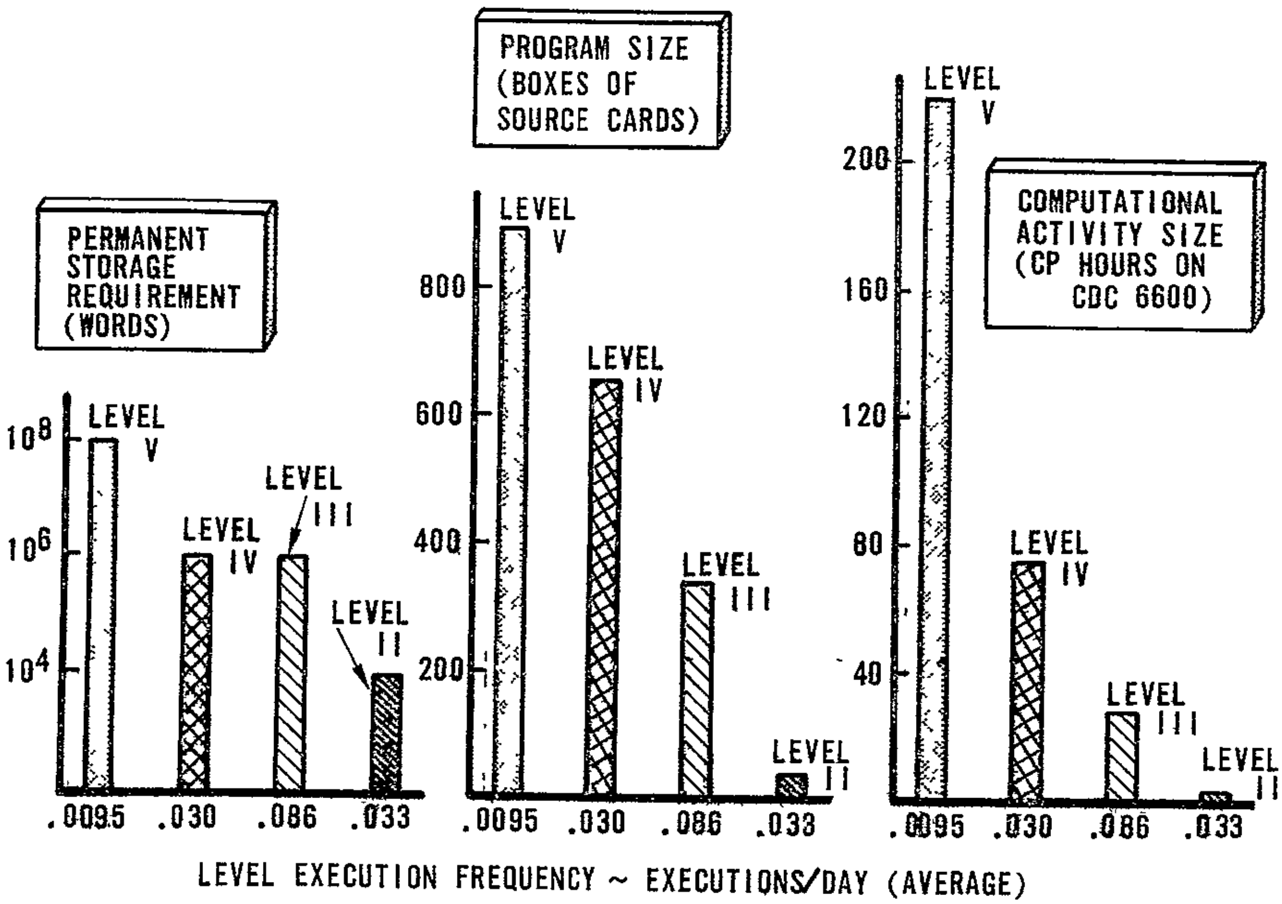


Figure 6.5 Project 2—Supersonic Commercial Transport—Computational Requirements Preliminary Design Levels

Table 6.II Summary Execution Time and Computational Flow Time Project I and Project 2

	DESIGN LEVEL	ESTIMATED NUMBER OF CONFIGURATION INVESTIGATIONS PER LEVEL IPAD INTEGRATED ENVIRONMENT	ESTIMATED EXECUTION TIME CP HOURS (CDC 6600) IPAD INTEGRATED ENVIRONMENT	ESTIMATED COMPUTATIONAL FLOW TIME (CDC 6600) ONE DESIGN CYCLE STANDALONE	ESTIMATED COMPUTATIONAL FLOW TIME (CDC 6600) ONE DESIGN CYCLE INTEGRATED ENVIRONMENT	ESTIMATED COMPUTATIONAL FLOW TIME (CDC 6600) CONVERGED DESIGN CYCLE STANDALONE	ESTIMATED COMPUTATIONAL FLOW TIME (CDC 6600) CONVERGED DESIGN CYCLE INTEGRATED ENVIRONMENT
PROJECT I SUBSONIC TRANSPORT	II	30	15 (½*)	2 Weeks	2 Days	**	**
	III	100	125 (1¼*)	8 Weeks	2½ Weeks	**	**
	IV	30	750 (25*)	***	I Month	***	2 Months
	V	8	300 (37*)	***	1½ Months	***	3 Months
	II Thru V	168	1190	—	—	12 to 15 Months	12 to 15 Months
	VI Thru IX	1	1080	—	—	42 Months	42 Months
	II Thru IX	169	2270****	—	—	54 Months	54 Months
PROJECT 2 SUPERSONIC TRANSPORT	II	45	25 (½*)	2 Weeks	2 Days	**	**
	III	115	2415 (21*)	***	1 Month	**	**
	IV	40	2400 (60*)	***	2 Months	***	4 Months
	V	8	1300 (170*)	***	3 Months	***	6 Months
	II Thru V	208	6140	—	—	***	22 to 30 Months
	VI Thru IX	1	3560	—	—	***	74 Months
	II Thru IX	209	9700****	—	—	***	96 Months

* CP HRS FOR EACH CONFIGURATION EVALUATED

** I DESIGN CYCLE PROVIDES A CONVERGED DESIGN AT LEVELS II AND III (SEE VOLUME II SEC 4.I)

*** NO EFFORT WAS MADE TO CORRELATE STANDALONE COMPUTATIONAL FLOWTIME TO THESE IPAD LEVELS

**** DOES NOT INCLUDE ALLOWANCES FOR RERUN OR OTHER CATEGORY INCLUDED IN FIGURES 6.6,6.7,AND 6.8

computational requirements that the IPAD System and host hardware must support.

6.4.2 Host Hardware Capacity

Based on the computational loads generated by the execution of each level of Project 1 and Project 2, together with other information regarding the Technical Program Elements which are candidates for first implementation of IPAD, information can be developed which will aid in specifying the host hardware capacity requirements for the IPAD System. These requirements are shown in figure 6.6 for Project 1 and in figure 6.7 for Project 2 which contain the following information for each level:

A. CPU Hours (on the CDC 6600)

   This item represents the computational load which would be required to execute a project on a continual, averaged basis. The project would consist of one continual preliminary design effort (Levels II through V) and a fraction of one product design effort (Levels VI through IX). In other words, it is unlikely that there will be continual product design effort because of limitations in resources and markets. Therefore, the user groups per year for the Project 1 product levels is reduced to 1/2, and the user groups per year for the Project 2 product levels is reduced to 1/8.

B. I/O Rate (Words/6600 CP Second)

   This item represents the input/output rate for typical Technical Program Elements in each level which are candidates for first implemenation of IPAD. This information was gathered from the existing Technical Program Elements (day files) in the form of (sectors/CP Second) X (64 words/sector). For example: Technical Program Element STR-6 (Program ATLAS) used in Level V of Project 1 and 2 has an I/O rate of 630 sectors/second.

C. Data Storage (Millions of 60 Bit Words)

   The data storage estimates are based on the amounts of sectors of data generated by typical Technical Program Elements in each level multiplied by the estimated number of runs that the user groups would desire to store on-line for each level.

LEVEL	CPU HOURS (CDC 6600)	I /O RATE (WORDS/6600 CP SEC)	DATA STORAGE (≈60 BIT WORDS)	SIMULTANEOUS INTERACTIVE PORTS
I	~ .1	~ 40,000	~ 1 M	~ 1
II	.1	30,000	1 M	1
III	.4	30,000	10 M	2
IV	2.0	30,000	75 M	5
V	.9	40,000	250 M	10
VI – IX	1.2	60,000	125 M	10
OTHERS	~ 2.0	~ 100,000	~ 5 M	~ 3
TOTAL	6.7	57,000 (WEIGHTED MEAN)	470 M	32

- BASED ON ONE AVERAGE 24 HOUR PERIOD

Figure 6.6 Project I – Subsonic Commercial Transport – Host Hardware Capacity Requirements

LEVEL	CPU HOURS (CDC 6600)	I/O RATE (WORDS/6600 CP SEC)	DATA STORAGE (≈60 BIT WORDS)	SIMULTANEOUS INTERACTIVE PORTS
I	~ .1	~ 40,000	~ 1 M	~ 1
II	.1	30,000	1 M	1
III	2.3	40,000	40 M	2
IV	2.4	40,000	300 M	5
V	2.1	60,000	1000 M	10
VI – IX	.7	80,000	1000 M	10
OTHERS	~ 2.0	~ 100,000	~ 5 M	~ 3
TOTAL	9.7	59,000 (WEIGHTED MEAN)	2350 M	32

- BASED ON ONE AVERAGE 24 HOUR PERIOD

Figure 6.7 Project 2—Supersonic Commercial Transport — Host Hardware Capacity Requirements

D. Simultaneous Interactive Ports

The number of simultaneous interactive ports for each level are based on the estimates of the number of interactive ports required to support each design effort with Technical Program Elements which are candidates for first implementation, together with an estimate of the number of simultaneous users. The line represented by Level I is devoted to continuing research which is described in Section 4.1 of Volume II.

The line represented by "others" includes Technical Program Element development, checkout, and maintenance. Beyond first implementation, it is expected that all items affecting hardware capacity are expected to increase.

In order to demonstrate what the demand on an IPAD system might be if a large aerospace company, such as The Boeing Company, were to commit to use the IPAD System, the components of the hardware capacity were scaled from the two projects to represent the following company mix of design efforts:

- 7 Continual Project 1 Preliminary Design Efforts
- 1 Project 1 Product Detail Design Effort committed to production status at two year intervals.
- 1 Continual Project 2 Preliminary Design Effort
- 1 Project 2 Product Detail Design Effort committed to production status at eight year intervals.

This company mix of design efforts would generate the hardware capacity requirements shown in figure 6.8.

6.4.3 Hardware Performance Requirements

The hardware capacity requirements were used to derive the hardware performance requirements shown in figure 6.9 for the following cases:

A. One continuous Project 1 effort

B. One continuous Project 2 effort

C. A continuous company mix effort.

C-2

LEVEL	CPU HOURS (CDC 6600)	I/O RATE (WORDS/6600 CP SEC)	DATA STORAGE (≈ 60 BIT WORDS)	SIMULTANEOUS INTERACTIVE PORTS
I	~ 1.0	~ 40,000	~ 1 M	~ 3
II	.3	30,000	1 M	1
III	5.1	30,000	110 M	3
IV	16.4	30,000	600 M	20
V	8.3	45,000	2750 M	20
VI – IX	2.0	70,000	1000 M	50
OTHERS	~ 9.0	~ 100,000	~ 40 M	~ 3
TOTAL	42.1	50,000 (WEIGHTED MEAN)	4510 M	100

- BASED ON:
  - EIGHT CONCURRENT DESIGN EFFORTS (7 Project 1 and 1 Project 2)
  - ONE AVERAGE 24 HOUR PERIOD

Figure 6.8 Company Mix— Host Hardware Capacity Requirements

PERFORMANCE CHARACTERISTIC		PROJECT 1	PROJECT 2	COMPANY MIX
• CPU POWER	• 3^{rd} GENERATION SCIENTIFIC MACHINE	.6 CDC 6600	.9 CDC 6600	3.8 CDC 6600
	• INSTRUCTION RATE (10^6 INSTRUCTIONS/SEC)	3.5	3.5	3.5
	• 4^{th} GENERATION VECTOR/ARRAY PROCESSOR	.04	.08	.32
	• INSTRUCTION RATE (10^6 INSTRUCTIONS/SEC)	40	40	40
• DATA STORAGE (BILLION BITS)		28	141	270
• SIMULTANEOUS INTERACTIVE PORTS		32	32	100
• GRAPHIC CONSOLES		4	4	12

Figure 6.9 Project 1, Project 2, and Company Mix Hardware Performance Requirements

The CPU power required is expessed both in terms of present third generation hardware (CDC 6600) and estimated fourth generation vector/array processors.

The requirements presented are based on extensive estimates from existing OM's which may be candidates for first implementation. Beyond first implementation, the hardware performance requirements would be expected to increase as more sophisticated OM's are included and the capability of interactive graphics is expanded.

The information shown for the company mix is considered realistic for a large aerospace company. It compares to the Boeing Company requirements for the 1966 through 1969 time period when two CDC 6600 computers were used to support the engineering development of commercial airplanes and at that time, one CDC 6600 was dedicated to the SST development effort. Other design organizations within the company were using additional computing facilities.

7.0 CONCLUSIONS

It is concluded that development of a computing system which provides the capabilities identified in sections 3 thru 6 will provide improved control of complex design problems and improved communications of technical information within the using community and with other organizations. These improvements during design should raise the quality of the product technical definition, aid in the identification and tracking of program costs, and improve the capability of the final product. Improved communications between engineering and manufacturing organizations and improved technical definitions of the product should decrease the number of engineering changes required during the manufacturing process, which should reduce both recurring and nonrecurring product costs.

8.0 USER RECOMMENDATIONS FOR IMPLEMENTATION

8.1 PURPOSE

The purpose of user recommendations for implementation is to identify a desired strategy for the initial development plan for the IPAD System. Subsequent extension during the long term development is also considered. The recommendations given here are in the context of user requirements for the design of the IPAD System.

8.2 INITIAL IMPLEMENTATION

The initial implementation of the IPAD System should provide all of the capabilities required to support the user-terminal-System interface requirements specified in sections 3, 4 and 5.

Specifically the IPAD System must provide for entering technical programs and data into the data base, modifying programs and data, displaying data, constructing and executing jobs, and continuity of the day-to-day activities.

It is recommended that NASA Langley Research Center identify the application standards to be included in the initial IPAD data base. Items such as standard atmosphere, dimensional units and standard conversions, numerical constants, physical constants, physical variables, etc., must be selected and specified prior to or during initial implementation.

Credibility of the IPAD technical capability recommended in Volume II will be highly affected by the training available to the initial users of IPAD. Relatively few design engineers have been exposed to a computer oriented environment. Therefore, a transition from design methods which use only limited computational capability will be required. It is recommended that a training program be developed to include familiarization in computer capabilities and IPAD applications. It is further recommended that IPAD System personnel who are highly trained in the use of the IPAD System and facilities be available to assist the using engineers.

8.3 LONG TERM DEVELOPMENT

The long term development of the IPAD System should support project planning capability and migration to later generation host computers. Additional capabilities and new facilities should be added when they are required to support the development and use of new technical code. In addition, improvements should be incorporated as problems are identified. The availability of next-generation host computers will also provide new potential for the IPAD System as well as increased computing power. Development work done in support of other major systems should be monitored for ideas which may be useful for the IPAD System. These events cannot be described or scheduled, but the IPAD System must be extended when it is prudent in order to provide the user with increased computing power and broader support from the IPAD System.

APPENDIX A

IPAD USER-SYSTEM INTERACTION

Volume III presented the user requirements for the user-system interface, and a language for the user to communicate with the IPAD System. Appendix B of Volume IV presents a generalized work flow model. Examples in this Appendix will demonstrate how typical users will apply the IPAD System to the general work flow model to perform assignments within the IPAD environment.

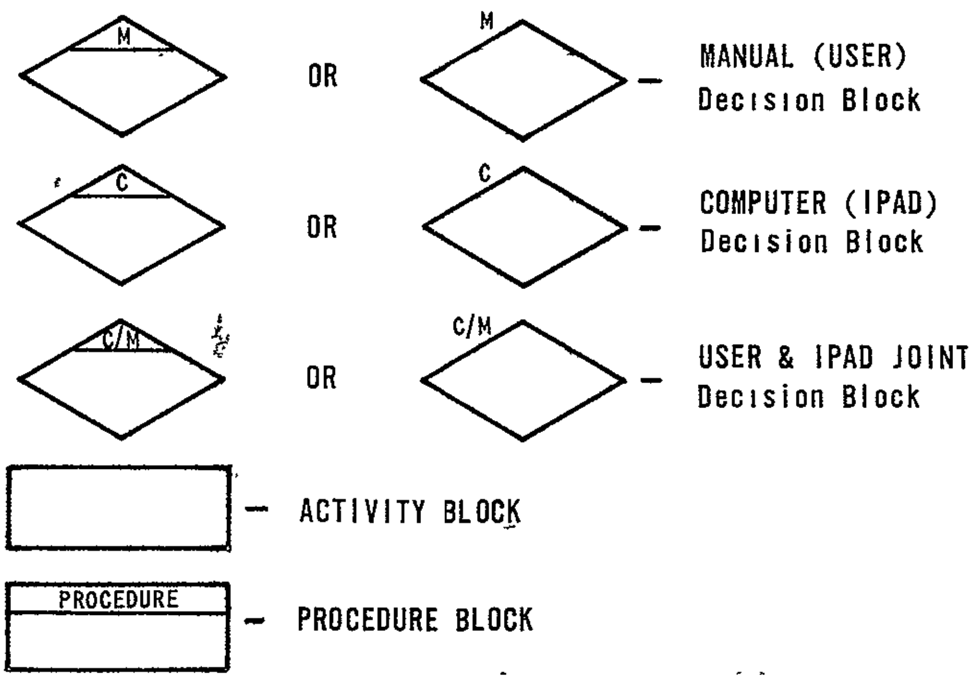
Before presenting this picture of the user-System interaction, several terms must be described. The total work in the design of a product is divided into four stages: project, task, subtask, and job. The project is the highest division, above which there is no meaningful division, and at which, all reporting would become complete. Each project will be divided into tasks, which will in turn be divided into subtasks. The individual user is identified at the subtask level, and his subtask is done by executing a sequence of jobs.

The size of a project and the related task, subtask and job sizes are not necessarily large. For example, if the project were to design an SST, then one task would be to design a delta-winged SST, one subtask would be to configure candidates in design Level III (see section 4.3.2.1 of Volume II). This would obviously be a large project. On the other hand, if the project were small, for example, to find uses for stability augmentation systems (SAS) on an SST, then one task would be to use SAS to improve ride quality, one subtask would be to improve the ride quality by changing the SAS electronics and not the airframe structure, and one job would be to synthesize the ride quality gains and filters. These examples do not imply that all activity must be partitioned into project, task, subtask and job. A user may employ IPAD to execute a subtask that is not related to a task or a job.

The generalized work plan presented in Volume IV has five activities: plan, prepare, modify, work, report. In the user-System interaction presented below, these five activities are identified. Plan-modify and prepare-modify are combined and shown together.

The user-System interaction network shown on figure A1 may be seen from two points of view. The network as a whole presents the major steps of the complete process for a project. But at the subtask level, where the individual user is active, the perspective changes and the network shows in considerable detail the user-System interaction. It is not necessary for a knowledgeable user to perform all the steps of this part of the network to do his subtask. For example, consider the user who knows which job he wants to execute, and has identified the data sets to use. In this case, his activity would start at Block 33 to set up and execute the job. The execution will be done in Block 38, and examination of the results will be done in Blocks 39 to 48.

The user-System interaction network is presented, followed by five procedures used in the network. The networks use the following blocks.



Each of these blocks will be numbered, for ease of relating the discussion of the block to the network drawing representing the activity.

A.1 USER-SYSTEM INTERACTION NETWORK.

This network presents the general activity done by the user in the IPAD environment, from the planning stage at the project

FOLDOUT FRAME 1

FOLDOUT FRAME 2

Figure A1. IPAD User/System Interaction

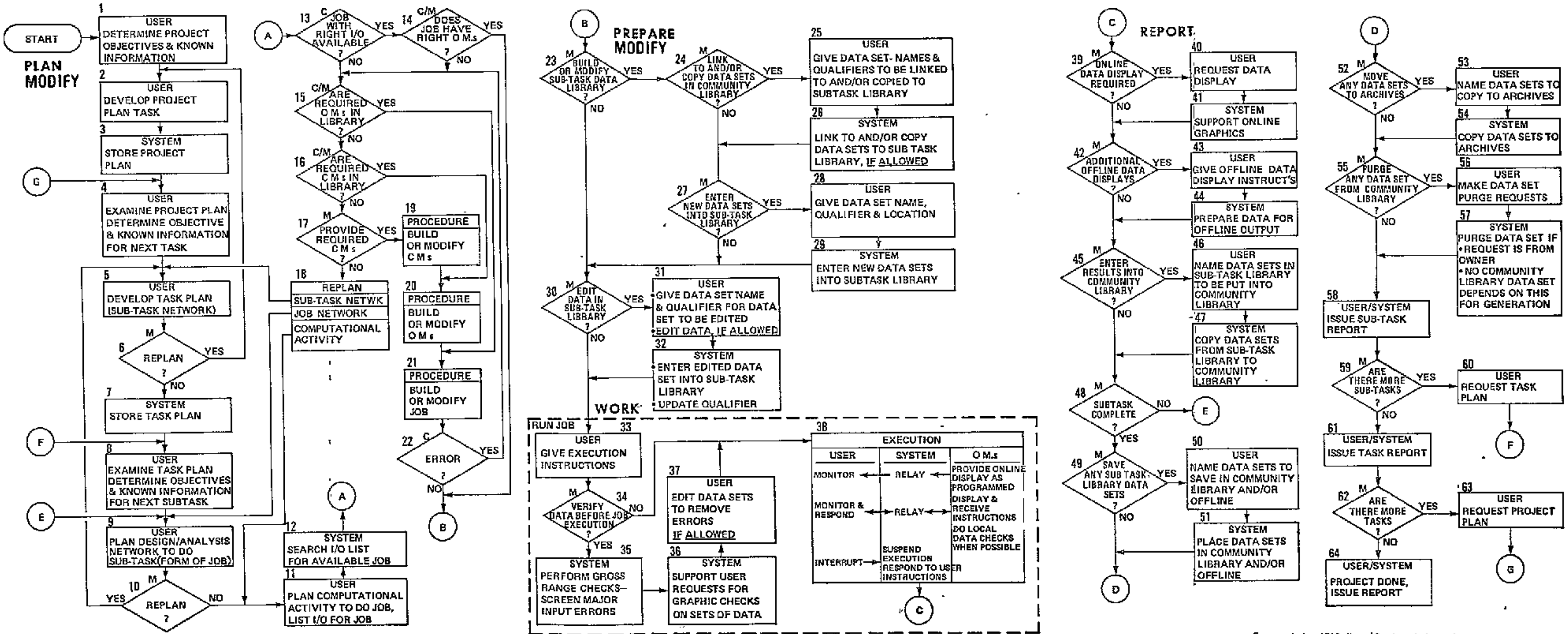


Figure A.1 IPAD User/System Interaction

level, to the execution of a single job, to the reporting of the work as it is completed. The network is shown on figure A1. The first blocks are in the plan-modify activity of the work flow model.

Block 1. Determine Project Objectives and Known Information--This is a management function, to collect background information for the planning of the project. This is supported by IPAD in gathering information related to previous projects from the data base.

Block 2. Develop Project Plan--The managers will develop comprehensive plans and schedules for the project, using the information gathered in Block 1.

Block 3. Store Project Plan--The project plan is stored in IPAD, and becomes accessible by the Management Information System. This project plan is in the form of a network of tasks.

Block 4. Examine Project Plan--This activity marks the development of the plan for a particular task. Using the Management Information System and the stored project plan, the next task to be done is located. The objectives of this task and the information available at that time are also obtained from the data base.

Block 5. Develop Task Plan--A plan of the subtasks to do the task is developed, using the information from Block 4. This plan of the subtasks is stored in the data base, and thus is available for monitoring by the Management Information System.

Block 6. Replan?--The project plan (task network) must be modified, and the activity returns to Block 2 if a task plan cannot be developed, either because sufficient information will not be available at that point in the project plan, or because the objectives cannot be met.

Block 7. Store Task Plan--The subtask network describing the task plan is stored in the data base, where it can be accessed by the Management Information System.

Block 8. Examine Task Plan--This activity marks the point where the individual user is first seen in the IPAD environment. The user examines the task plan and determines what information is available and what the objectives are for the subtask that he is to do. The Management Information System will also inform the user of the proper schedule for his subtask.

Block 9. Plan Design-Analysis Network--The user will develop the network of design and analysis jobs that he must do

ORIGINAL PAGE IS
OF POOR QUALITY

to complete the subtask. This planning uses the information developed in Block 8.

Block 10. Replan?--The subtask plan (job network) must be modified, and the activity returns to Block 5 if a subtask plan cannot be developed, either because sufficient information will not be available at that point in the project plan, or because the objectives cannot be met.

Block 11. Plan Computational Activity To Do Job--The user develops a computational plan for the job, including the modular activities and their sequence. A list of input data and output data are prepared.

Block 12. Search Input-Output List for Available Job--It is possible that a satisfactory job is already available in the IPAD environment. The user commands the System to search the input-output lists of the jobs known to the System, and to indicate any jobs that satisfy the input-output list that he prepared in Block 11.

Block 13. Job With Right Input-Output Available?--If the System finds one or more jobs with the correct input-output, the activity proceeds to Block 14. If not, the activity goes to Block 15.

Block 14. Does Job Have Right OM's--The user must determine if any of the available jobs with the correct input-output also have suitable OM's. The user makes this determination by having the System supply him with the abstracts of the OM's in the jobs. If a job is found that uses OM's accepted by the user, then the activity moves to Block 23 to prepare the input data. If not, then a new job must be built, and the activity goes to Block 15.

Block 15. Are Required OM's in Library?--The user searches both his subtask library and the community library to see if satisfactory OM's are available, but not yet collected into a job. This search is done by giving the System a list of OM key words, then reading the abstract of any candidate OM's found by the System. If satisfactory OM's are found, the activity goes to Block 21, to build the OM's into a job. If the user cannot find suitable OM's to build his job, or can find none at all, then the activity goes to Block 16.

Block 16. Are required CM's in Library?--The OM's to be built or modified will be done with CM's. The necessary CM's may already exist in the user's subtask library or in the community library. This is determined by searching the library looking for CM's with key words matching those supplied by the user. If all the required CM's are available, then the activity proceeds to Block 20 to build the desired OM's. If suitable CM's are not available to build the OM's, or not available at all, the activity goes to Block 17.

Block 17. Provide Required CM's?--If new CM's are required, the user will decide either to supply the new code or not to. If the choice is to supply new code, then the activity moves to Block 19, where the CM's are modified. If the user chooses not to provide the missing code, then the activity goes to Block 18.

Block 18. Replan--At this point, the user has decided that the needed code is not available, and that he is not going to supply that code. Three options are available. The activity can go to Block 5 to develop an entirely new task plan, with subtasks different from the current ones. Or, the activity can return to Block 9, where only the form of the job will be replanned. Lastly, the activity can return to Block 11, where the plan of the computational activity for the job is revised.

Block 19. Procedure: Build or Modify CM's--This procedure is presented in section A.4 and is the means of entering new code and collecting it into CM's.

Block 20. Procedure: Build or Modify OM's--This procedure is presented in section A.5 and is the method whereby the required OM's are produced.

Block 21. Procedure: Build or Modify Job --This procedure is also presented in section A.6 and is the activity that results in the job to be executed.

Block 22. Error?--It is possible that the attempt in Block 21 to produce a job is not successful. If not, then the activity returns to Block 15, to try again to provide the needed CM's and OM's, guided this time by the unsuccessful attempt to build a job. If there is no error, the user has a job ready to execute, and is now ready to prepare the data.

Block 23. Build or Modify Subtask Data Library?--If the data required to execute the job is already available, the activity proceeds to Block 30. If not, then Block 24 is performed next.

Block 24. Link to and/or Copy Data Sets in Community Library--If there are data required for job execution located in the community library, the activity proceeds to Block 25. If not, then Block 27 is done next.

Block 25. Give Data Set Names to be Linked to and/or Copied to Subtask Library--The user gives the names and qualifiers of the data sets he wants to be available to his subtask library. He needs to give only enough of the qualifier to make the identification unique.

Block 26. Link to and/or Copy Data Sets to Subtask Library --The System will either provide a copy or link to the data set. Both actions are subject to security, access and permission code approval.

Block 27. Enter New Data Sets Into Subtask Library?--The user specifies whether he needs to enter new data into the IPAD environment. If not, the activity skips to Block 30.

Block 28. Give Data Set Name, Qualifier and Location--The user tells the System the defining information for the data set, names his part of the qualifier, and gives the location where the data will enter the System.

Block 29. Enter New Data Sets Into Subtask Library --The System brings in the data and completes the qualifier.

Block 30. Edit Data in Subtask Library?--At this point, the user has available a complete collection of the qualifier data sets required to execute his job. However, some of the data sets may require some editing of their contents. If not, the activity skips to the Work stage, which begins with Block 33.

Block 31. Give Data Set Name and Qualifier, Edit Data--The user gives the name and qualifier of each data set that he wants to edit. He will be allowed to edit if the permission and access codes allow him to do so.

Block 32. Enter Edited Data, Update Qualifier--The System responds to the user's editing instructions, updates the qualifier, and enters the edited data into the subtask library. If the user has edit-in-place permission, and desires to do so, the editing is done to a copy, instead of the original.

The next group of activity blocks are in the Work category. This is the activity that executes the job, using the identified data sets.

Block 33. Give Execution Instructions--The user completes the instructions required by the job, the IPAD System, and the host operating system in order to execute the job.

Block 34. Verify Data Before Job Execution?--The user may choose to verify the data before it is executed. If so, two types of verification are performed in Blocks 35 and 36. If not, the activity goes to Block 38.

Block 35. Perform Gross Range Checks--The System can use the gross range information contained in the library for each engineering variable to check the data sets. For example, if the gross range for Mach number if $0 \leq M \leq 100$, then a gross range check would indicate a Mach number = -12 as being in error. This will help to identify major errors resulting from keypunch errors, etc. The gross range check is done entirely by the System, and the System signals the user of all discrepancies noted.

Block 36. Support User Requests for Graphic Checks on Data--The user employs the graphic capabilities supported by the IPAD System to check the data sets for errors in the data that are small in magnitude. Tables of variables may be displayed against each other, and geometries will be drawn and viewed for smoothness.

Block 37. Edit Data Sets to Remove Errors--The user will want to remove all the errors discovered during Blocks 35 and 36. This editing will be supported by the System, and will be subject to the access and permission codes.

Block 38. Execution--This represents the actual execution of the job by the host computer. The job may be operated "hands off", or may be executed with varying degrees of user and IPAD System intervention. Block 38 presents a table that shows the actions of the user, the IPAD System and the OM in execution. Three different user functions are shown.

1. The OM is programmed to provide online display. The IPAD and host systems relay to the user, who monitors the display.

2. The OM is programmed to display and receive instructions. The IPAD and host systems relay the display to the user, who monitors and responds. The IPAD and host systems relay the response back to the OM.

3. The user may issue an interrupt command at any time. The IPAD and host systems suspend execution and respond to the user instructions.

These functions cover the general categories of user-system interaction during execution. Adequate support of the user's needs during execution will be one of the more important aspects of the IPAD environment.

The actual execution of the job in Block 38 marks the end of the Work activity. The final phase of the generalized work model is the Report on the results.

Block 39. Online Data Display Required?--The user may desire to examine the results online. If not, the activity skips to Block 42.

Block 40. Request Data Display--The user employs the DISPLAY command or equivalent features of the IPAD System to selectively view the results of the job execution. The user may request online hardcopy of the displays.

Block 41. Support Online Graphics--The System provides the support for the user's graphic requests.

Block 42. Additional Offline Data Displays? --The user may select offline plots, drawings or text to be generated. If not, the activity skips to Block 45.

Block 43. Give Offline Data Display Instructions--The user gives the instructions for the offline displays, using the DISPLAY command or its equivalent. The job itself may have prepared some offline data displays, as well.

Block 44. Prepare Data for Offline Output--The System prepares the data as instructed by the user.

Block 45. Enter Results into Community Library?--Certain results of the job may be of importance to the user community. These must be transferred to the community library. If not, the sequence skips to Block 48.

Block 46. Name Data Sets to be Put Into Community Library--The user names the data sets in the subtask library to be transferred to the community library.

Block 47. Copy Data Sets to Community Library--The System places copies of the named data sets into the community library, transferring as well the access codes the user has placed on the data sets.

Block 48. Subtask Complete?--If the job just executed has completed the subtask, the activity proceeds to Block 49. If there is more to be done to complete the subtask, the activity returns to Block 9 to determine the next job in the subtask plan.

Block 49. Save Any Subtask Library Data Sets?--When the subtask is declared complete by the user, the subtask library will not be saved. Consequently, the user must deliberately save those data sets in the subtask library that are to be saved. This will be done in Blocks 50 and 51. If not, the activity skips to Block 52.

Block 50. Name Data Sets to Save in Community Library and/or Offline--The user names the data sets to be saved and gives their destination, whether to the community library, punched cards, magnetic tape, etc.

Block 51. Place Data Sets in Community Library and/or Offline--The System responds to the user's instructions.

Block 52. Move Any Data Sets to Archives?--Some of the data sets generated by the job may be moved to the archival storage. If none are to be moved, the activity skips to Block 55.

Block 53. Name Data Sets to Copy to Archives--The user names those data sets to be sent to archives.

Block 54. Copy Data Sets to Archives--The System performs the action requested by the user.

Block 55. Purge Any Data Set From Community Library--The completion of the job may be the last usage for a data set in the community library. In such cases, the data set may then be purged, by the activity of Blocks 56 and 57. If not, the activity skips to Block 58.

Block 56. Make Data Set Purge Requests--The user identifies by name and qualifier the data sets to be purged.

Block 57. Purge Data Set--The System will purge the data set if several conditions are met. If the owner makes the purge request, and if no other data sets in the community library depend on that data set for their generation, then the data set will be purged. Or, if the user has absolute purge authority granted by the permission codes, then the data set will be purged.

Block 58. Issue Subtask Report--This marks the end of the subtask that was begun in Block 8. The report alerts the Management Information System that this subtask is completed.

Block 59. Are There More Subtasks?--The managers will use the Management Information System to tell if there are more subtasks in this task. If so, the activity proceeds to Block 60. If not, the activity skips to Block 61.

Block 60. Request Task Plan --Again using the Management Information System, the task plan is recalled, the next subtask is determined, and the activity returns to Block 8 to plan the next subtask.

Block 61. Issue Task Report--When all the subtasks within a task are complete, a report on the task is issued. This alerts the Management Information System that this task is complete.

Block 62. Are There More Tasks?--The managers will use the Management Information System to tell if there are more tasks to be done before the project is complete. If not, the activity skips to Block 64.

Block 63. Request Project Plan--Using the Management Information System, the project plan is recalled, the next task is determined, and the activity returns to Block 4 to plan the next task.

Block 64. Project Done, Issue Report--The project is completed when the last task has been done. A report is issued and the Management Information System can then alert the user community. This does not terminate the community library, however. The information generated during the project may be of use for some time.

A.2 PROCEDURE: DEFINE ENGINEERING VARIABLE TO DICTIONARY

Several groups of activities in the user-System interaction will be repeated many times in the same manner. These are presented separately as procedures to emphasize that they are within themselves complete actions. Five of these procedures are discussed as support for the user-System interaction network of figure A1. Each procedure is presented separately, beginning with this section.

This procedure presents the process of defining a new engineering variable to the dictionary. An engineering variable is the uniquely defined entry in the engineering or scientific sense. The variable may have many names in the code, but it has only one engineering definition. The network for this procedure is presented on figure A2.

Block EV-1. Dictionary Search--The user selects either the subtask dictionary or the community dictionary, and gies a key word list that identifies the variable he wants to define. The System searches the engineering variable key word list in the selected dictionary and indicates variables that agree with the user's key word list.

Block EV-2. Variable Already in Dictionary?--The user reads the abstracts for the variables identified by the key word search. If the variable he wanted to define is already in the dictionary then he is finished with this procedure. If not, the activity moves to Block EV-3.

Block EV-3. Make Entry Into Dictionary--The user must make his own entry for the variable. He gives the name, an abstract and the gross range check information. The System makes this a new entry, and adds the key word list given in Block EV-1. The activity then returns to the point where the procedure was called.

A.3 PROCEDURE: DEFINE OR MODIFY DATA SET INTO DICTIONARY

This procedure gives the process of defining or modifying the part of a data set that is contained in the dictionary. The network for this procedure is presented on figure A3.

Block DS-1. Define or Modify?--The user specifies whether he wants to define a new data set or modify an existing one. If he wants to define a new one, the action skips to Block DS-6.

Block DS-2. Give Location and Name--The user gives the location and name of the data set dictionary entry that is to be modified. The System searches the library for any qualified data sets, CM's or OM's that depend on or have used this data set.

Block DS-3. Dependent Modules Not Owned by User?--If there are dependent modules or data sets not owned by the user, then the user may not modify the data set definition, and the activity skips to Block DS-6.

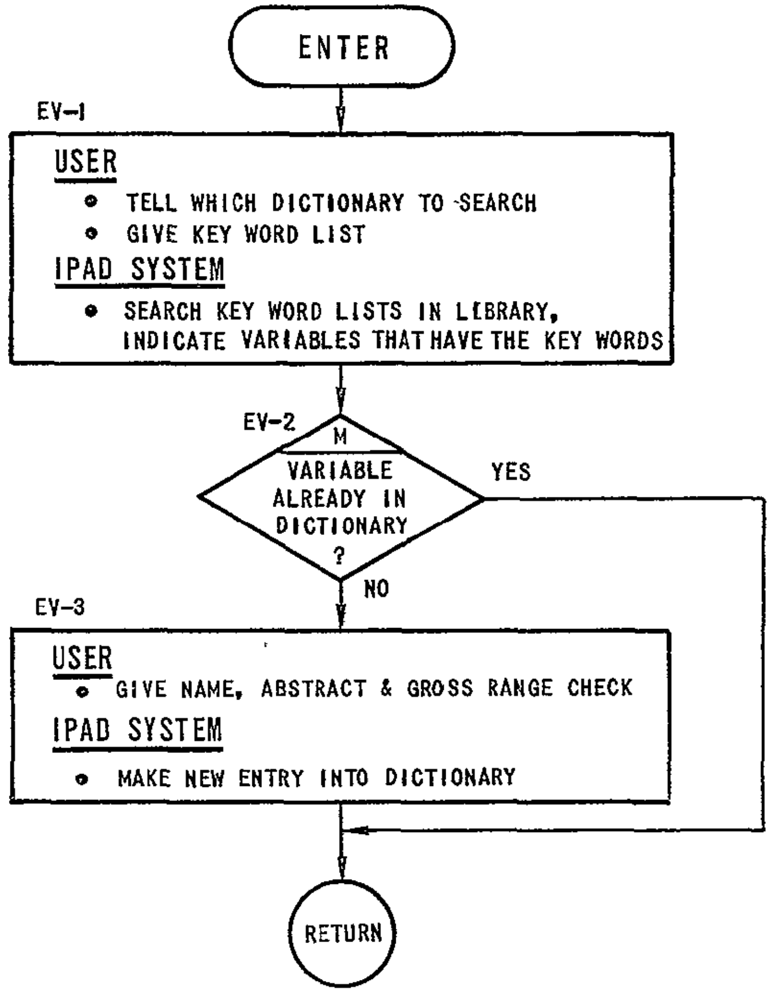


Figure A.2 Procedure: Define Engineering Variable to Dictionary

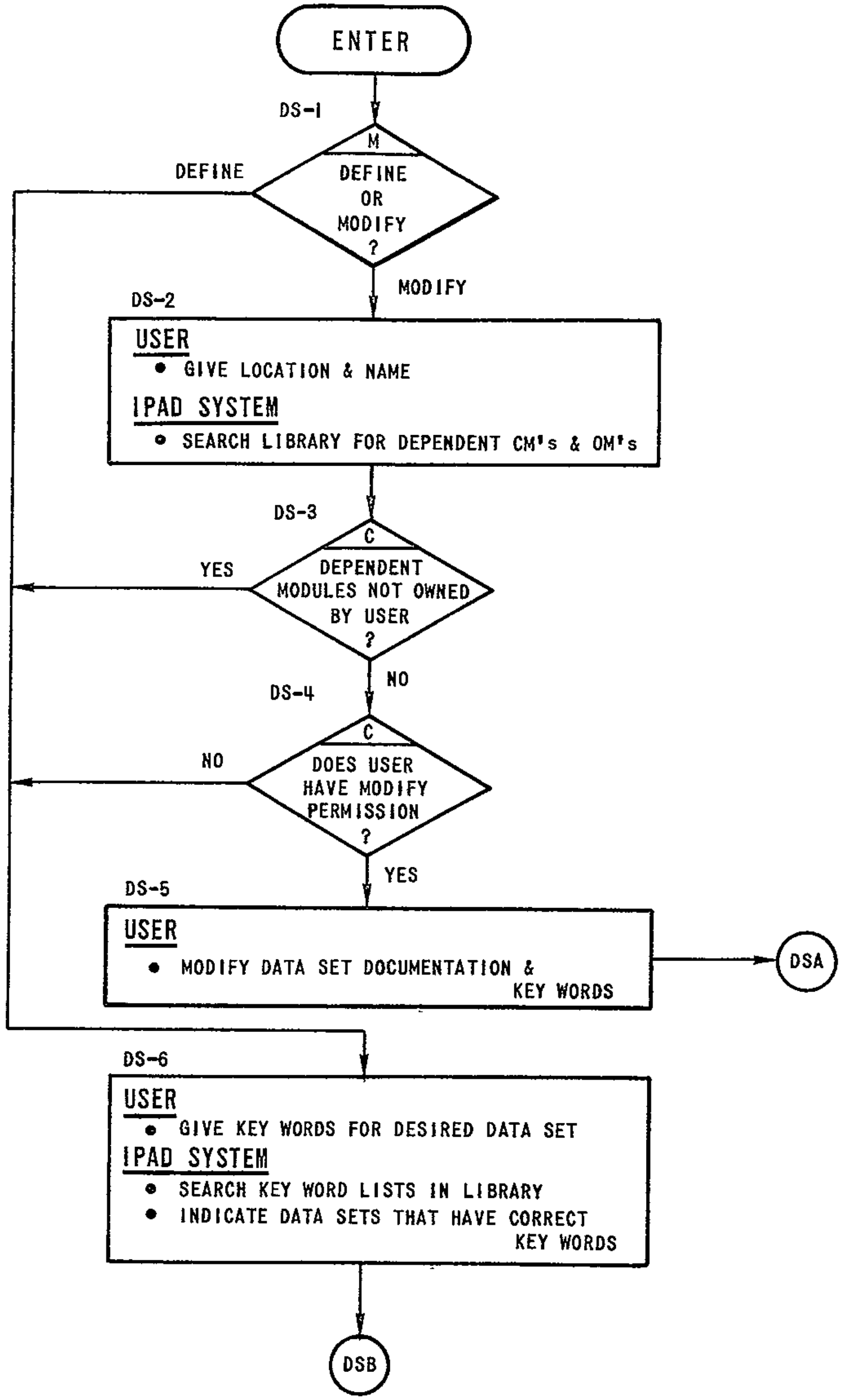


Figure A.3 Procedure: Define or Modify Data Set into Dictionary

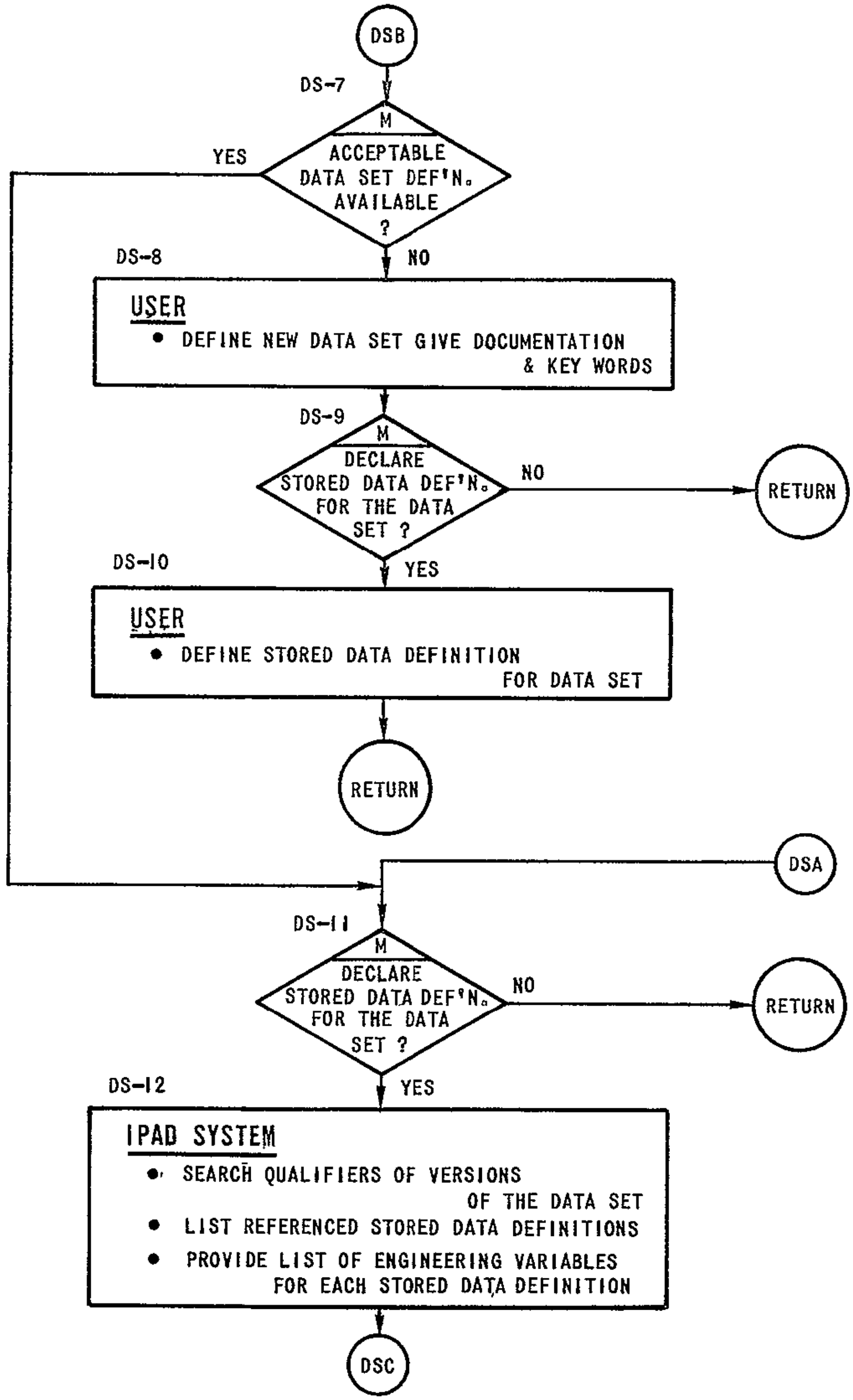


Figure A.3 (Cont'd)

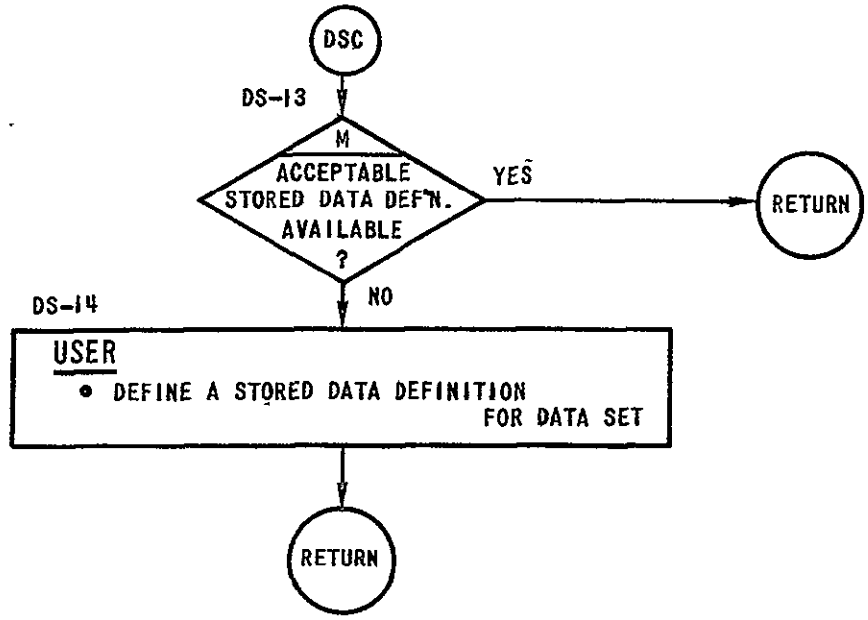


Figure A.3 (Cont'd)

Block DS-4. Does User Have Modify Permission?--The user must have access code and permission code authority to modify a definition. If not, the activity skips to Block DS-6.

Block DS-5. Modify Data Set Documentation and Key Words--The user supplies the desired changes thru the DEFINE command to alter the abstract, references or key words. The System makes the changes, and the activity proceeds to Block DS-11.

Block DS-6. Define New Data Set--The user has either chosen to define a new data set, or has not been allowed to modify an existing one, and must define a new one. He gives the key words for the new data set. The System searches the key word lists for data sets in the dictionary, and indicates existing data sets that have the correct key words.

Block DS-7. Acceptable Data Set Definition Available?--The user examines the abstracts of the data sets indicated by the System, and if a suitable data set is already defined, the activity goes to Block DS-11.

Block DS-8. Define New Data Set--The user gives the name of the new data set and an abstract, and the System combines these with the key words given in Block DS-6 to make a new entry into the dictionary.

Block DS-9. Declare Stored Data Definition?--The user may care to define a stored data definition at this time for the data set. If not, the procedure is complete, and the activity returns to the point where the procedure was called.

Block DS-10. Define Stored Data Definition--The user defines a stored data definition for the data set. The activity exits the procedure at this point.

Block DS-11. Declare Stored Data Definition for the Data Set?--This Block is reached after an existing data set has been modified or chosen, and may or may not have stored data definitions related to it. If the user wants to check on this, the activity proceeds to Block DS-12. If not, the activity exits the procedure at this point.

Block DS-12. Search Qualifiers of Data Set, List Referenced Stored Data Definitions--The System searches the qualifiers of all the versions of the data set, lists the stored data definitions referenced in the qualifiers, and lists the engineering variables in each stored data definition.

Block DS-13. Acceptable Stored Data Definitions Available--The user examines the information provided by Block DS-12, and if an acceptable stored data definition exists, then the activity exits the procedure at this point.

Block DS-14. Define a Stored Data Definition--The user has not found a suitable stored data definition, and so he provides that information which the System enters into the library. This concludes the procedure for defining or modifying data sets.

A.4 PROCEDURE: BUILD OR MODIFY COMPUTATIONAL MODULE

A coding module is the smallest division of user source code that is qualified in the IPAD environment. The coding module has two distinct parts. The dictionary contains the generic information for the coding module, and the library contains the specific contents of a qualified version of the coding module. The network for this procedure is shown on figure A4.

Block CM-1. Enter New Code?--The user chooses whether new code will be entered (Block CM-5) or existing code will be modified (Block CM-2).

Block CM-2. Are CM Names Known?--The user has chosen not to enter new code. If he knows the CM names to work with, the activity jumps to Block CM-6. If not, the System will assist him in the next block in determining the CM's to use.

Block CM-3. Provide Key Word List and Search for CM's--The user provides a key word list for a CM search, which the System does. If any CM's are found, the user examines the abstracts of the CM's for suitability.

Block CM-4. Suitable Code Available?--If the user has found CM's in the library suitable for building with or for modifying, the activity skips to Block EM-6.

Block CM-5. Provide Code, Name, Key Words--The user is entering new code into the IPAD System. He first makes the dictionary entry; name, generic abstract, etc. Then he gives a specific abstract, status and supplies the actual code. The System enters this into the library.

Block CM-6. Build or Modify Code?--The user selects whether he is building new code by merging existing CM's (Block CM-13), or by modifying existing code (Block CM-7).

Block CM-7. Edit in Place?--The user may desire to edit-in-place, that is, edit without preserving the existing contents (Block CM-8), or edit a copy (Block CM-12).

Block CM-8. Search Library for Dependent CM's, OM's, Jobs--If the user has chosen to edit-in-place, the System searches the library for any CM's, OM's, or jobs that depend on the CM about to be edited.

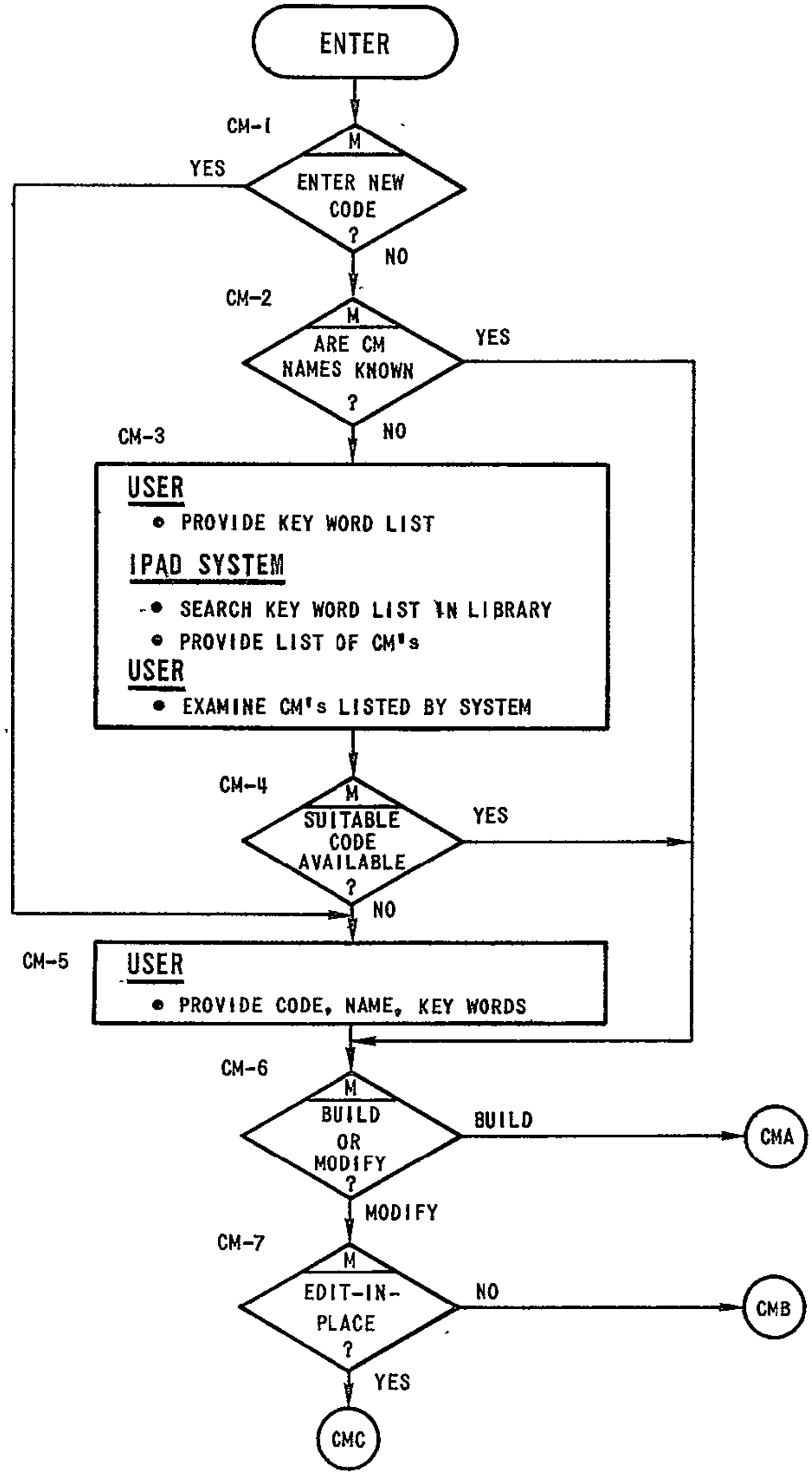


Figure A.4. Procedure: Build or Modify Computational Module (CM)

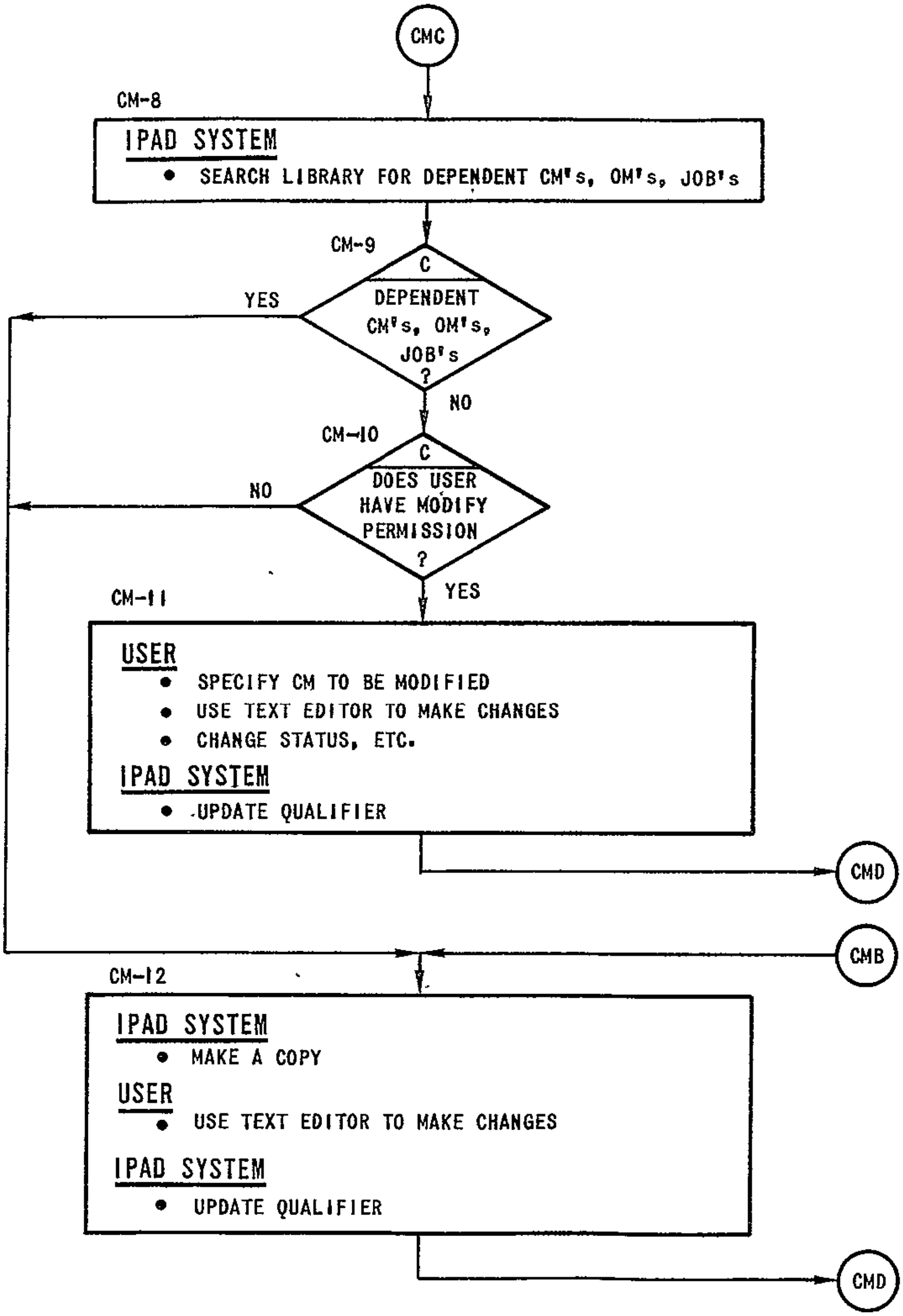


Figure A.4 (Cont'd)

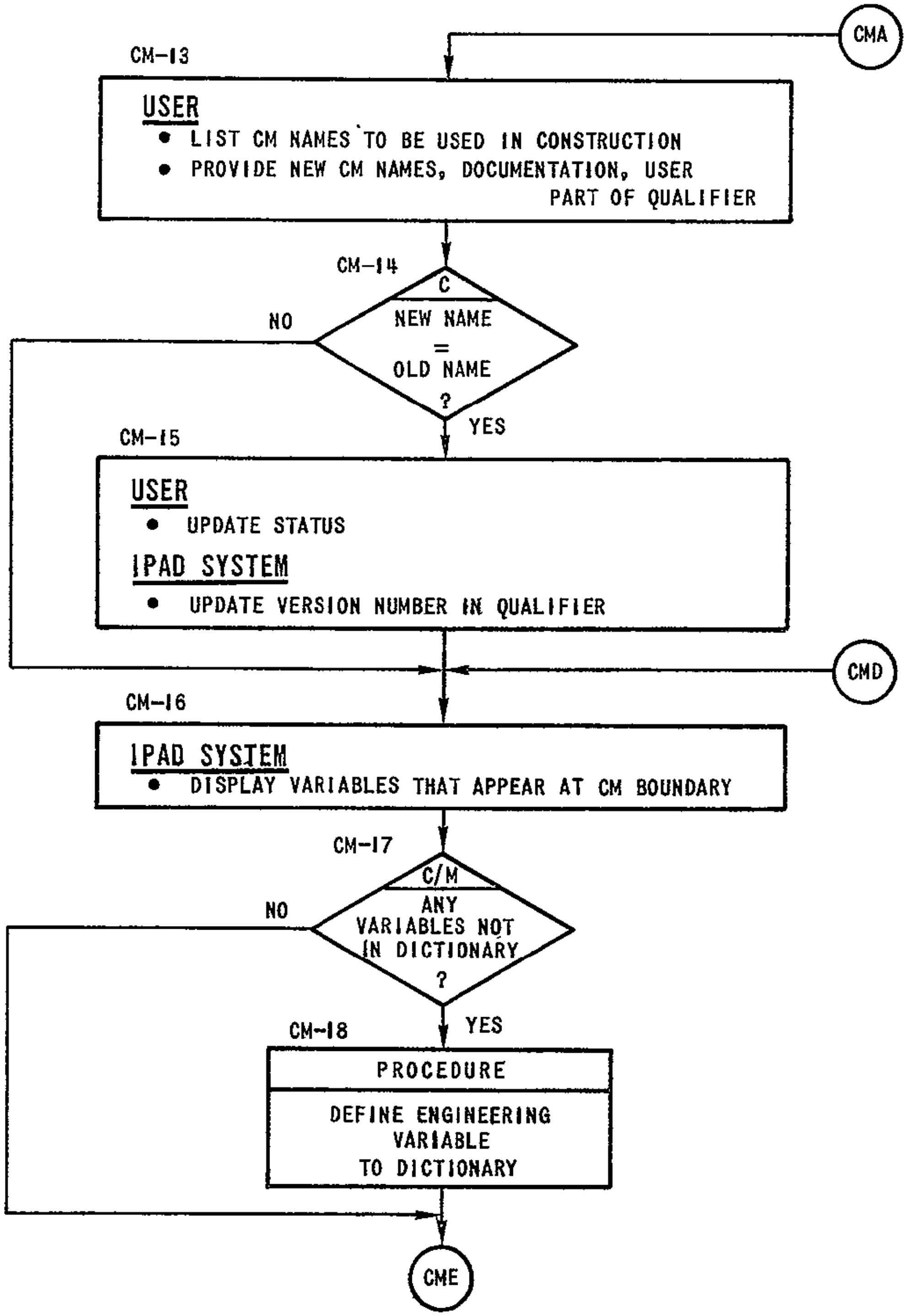


Figure A.4 (Cont'd)

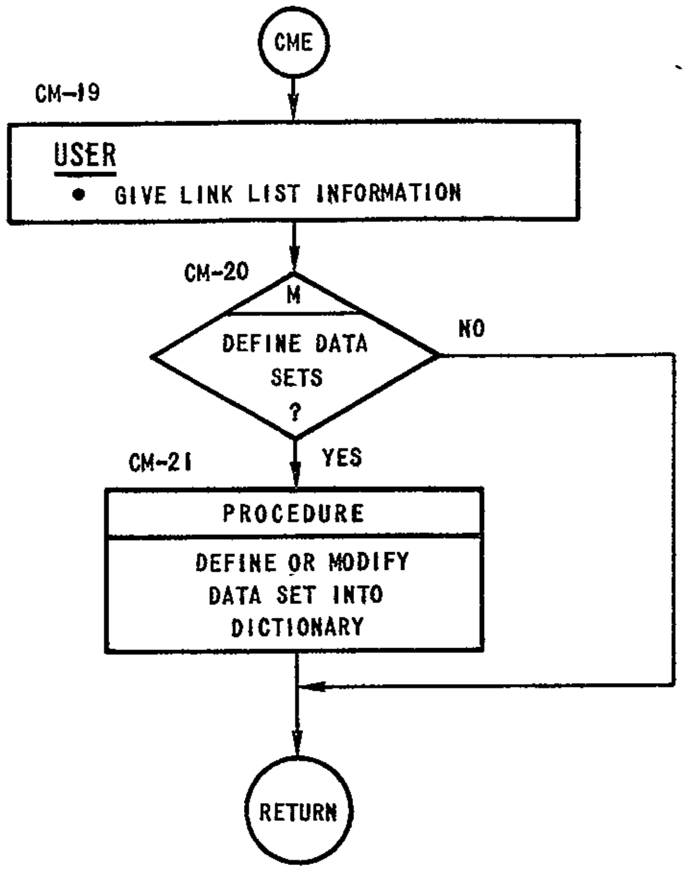


Figure A.4 (Cont'd)

Block CM-9. Dependent CM's, OM's, Jobs?--If the System has found dependent CM's, OM's or jobs, the user will not be allowed to edit-in-place, so the activity jumps to Block CM-12.

Block CM-10. Does User Have Modify Permission?--The user will be allowed to modify only when he has permission code to do so. Otherwise, his modifications must be made to a copy, and the activity moves to Block CM-12.

Block CM-11. Specify CM and Make Changes--The user specifies the CM to be changed, uses the text editor in the EDIT command to make changes, and when satisfied, updates the status, document and user part of the qualifier. The System completes the update of the qualifier and stores the CM in the library. The activity jumps to Block CM-16.

Block CM-12. Copy CM, Make Changes, Store--The System must make a copy of the code to be modified, as the user cannot or does not want to edit-in-place. The user then employs the text editor under the EDIT command to make changes. Once these changes are done, the user corrects the dictionary information, the code is qualified and stored in the library. The activity goes to Block CM-16.

Block CM-13. List CM's to be Used in Building--This block represents the building of a new CM from existing CM's, using the editing features under EDIT. The user provides a new name, dictionary documentation and user part of the qualifier.

Block CM-14. New Name = Old Name?--The System determines if the new name is the same as the old name. If not, the activity jumps to Block CM-16.

Block CM-15. Update Status and Version Number--If the new name is the same as the old name, then the user corrects the status of the CM, and the System increments the version number and stores the CM in the library.

Block CM-16. Display Variables That Appear at CM Boundary--The system displays the variables that appear at the boundary of the CM. These will be displayed in their local name. The user translates to the engineering variable equivalent, and prepares a list linking the local name to the engineering name. This is called the link list.

Block CM-17. Any Variables Not in Dictionary?--During the preparation of the link list, it may be discovered that certain variables are not defined to the dictionary in the engineering sense. Blcok CM-18 will be done for those missing variables. Otherwise, the activity skips to Block CM-19.

Block CM-18--Procedure: Define Engineering Variables to Dictionary--This is the procedure presented in section A.2.

Block CM-19. Give Link List Information--The user tells the System the link list for the CM, and this becomes a part of the qualified portion of the code.

Block CM-20. Define Data Sets?--The user may desire to define data sets for this CM. If not, the activity of this procedure is terminated. If so, the activity continues to Block CM-21.

Block CM-21. Procedure: Define or Modify Data Set Into Dictionary--This is the procedure presented in section A.3, and in it the user prepares data set definitions that relate to this CM. This activity marks the termination of the procedure for building or modifying CM's.

A.5 PROCEDURE: BUILD OR MODIFY OPERATIONAL MODULE

Operational modules (OM's) are organized networks of CM's. These networks and the CM's in them are in executable form. OM's may either be built as new or may be developed by modifying existing OM's. The network for this activity is shown on figure A5.

Block OM-1. Build or Modify?--The user chooses whether to build (Block OM-3) or modify (Block OM-2).

Block OM-2. Modify Existing OM's--The user identifies the OM's that are going to be modified. The System lists the CM's and displays the CM network in that OM. The user then lists new or replacement CM's and makes changes to the CM network, using the CONSTRUCT command. Once the user has completed the modifications, the System makes these cahnges to a copy of the original OM. The activity skips to Block OM-4.

Block OM-3. Build New OM--The user lists the CM's and their network to be used in defining a new OM.

Block OM-4. Select Paths, List Data Set Connections--The user must now determine if the CM networks perform the intended functions and that the data sets for the OM are properly defined, since data sets will be employed at OM boundaries during execution. The user selects paths thru the logic networks of the OM, and the System lists the data sets that appear as input, output or are internally connected along that path. Variables that have not yet been assigned to a data set are also identified.

Block OM-5. Are Required Data Set Names Known?--If all connections are correct, and there are no variables that are not assigned to a data set, then the activity skips to Block OM-11.

Block OM-6. Search for Suitable Data Sets--The user must supply some additional data sets to the OM. This process begins by searching the dictionary for presently defined data sets that will be satisfactory. These are the unqualified data sets, without any contents at this time. The user first gives a key word list, and the System searches the dictionary for data sets

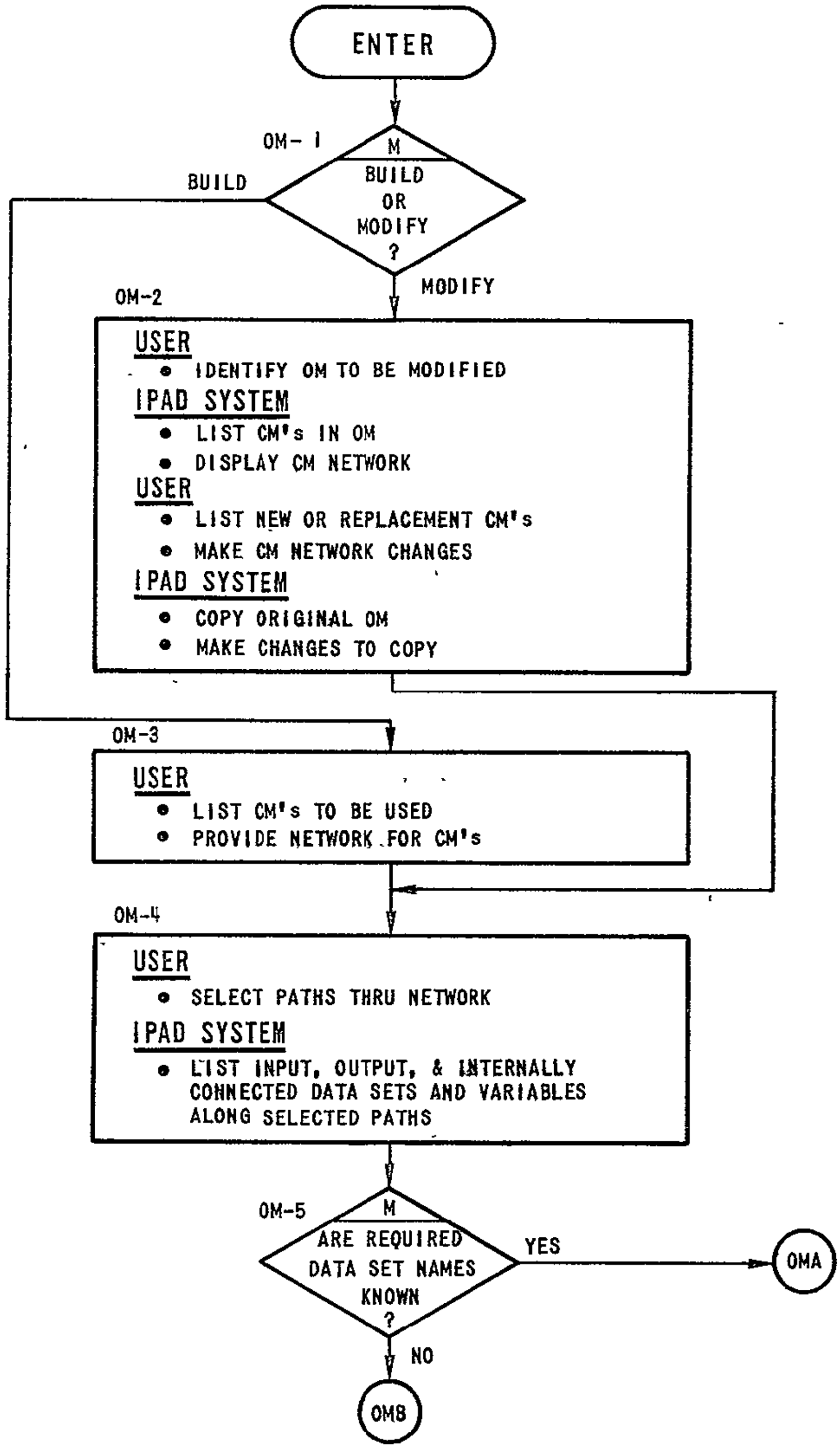


Figure A.5 Procedure: Build or Modify Operational Module (OM)

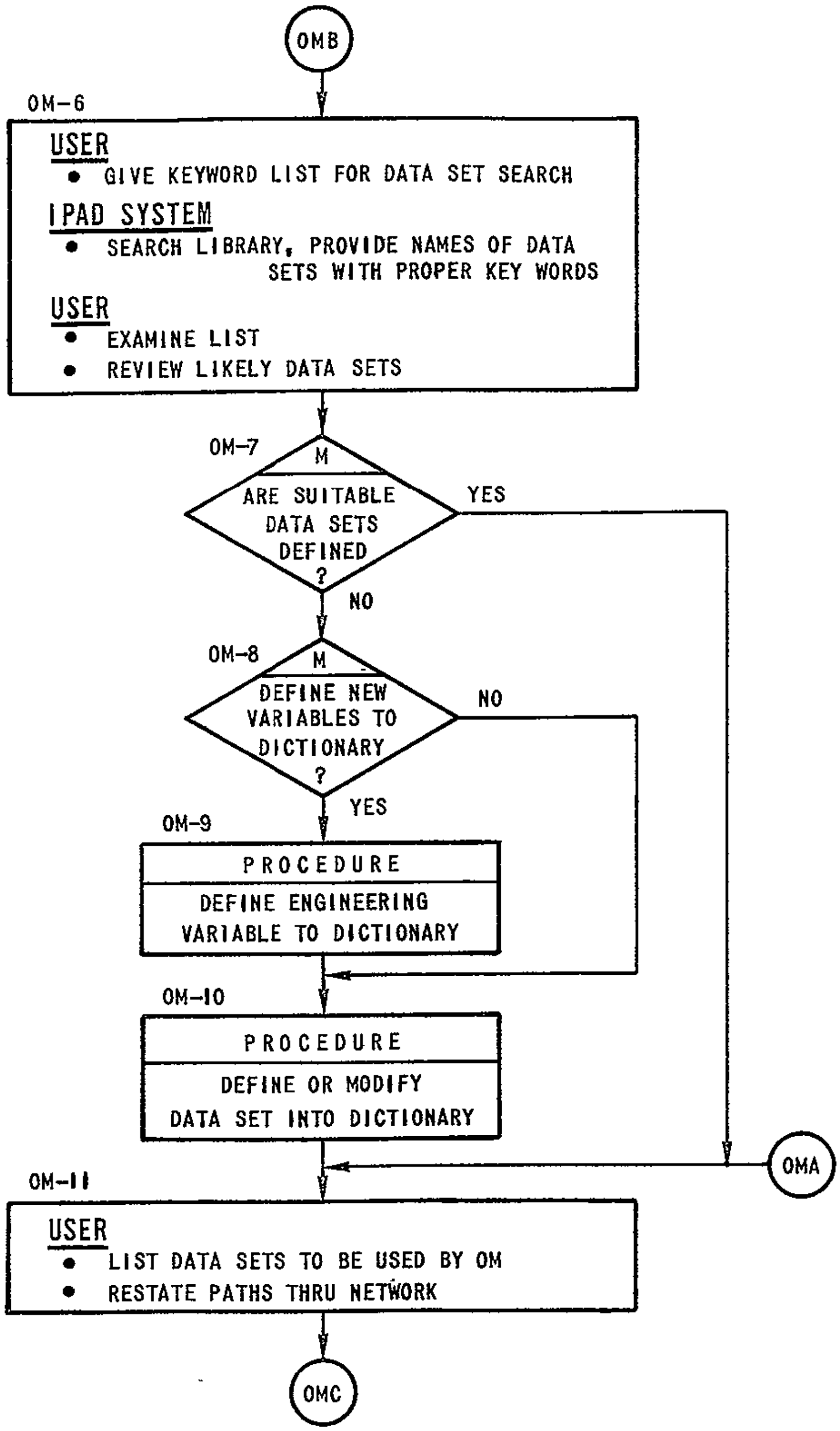


Figure A.5 (Cont'd)

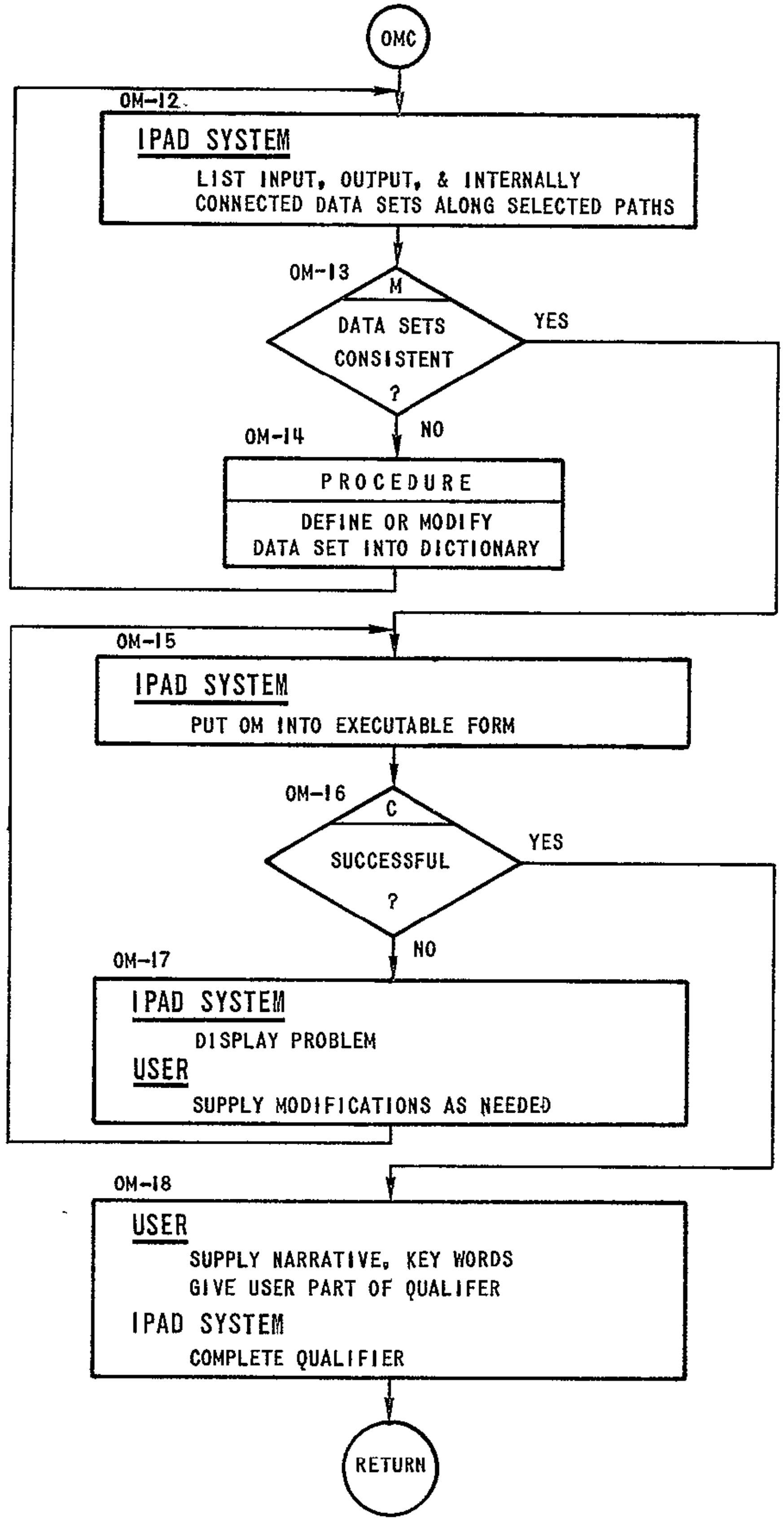


Figure A.5 (Cont'd)

with the same key words and provides their names to the user. These are examined and reviewed.

Block OM-7. Are Suitable Data Sets Defined?--This decision by the user requires that both suitable data sets and complete assignment of variables to the data sets is available, in order to skip to Block OM-11.

Block OM-8. Define New Variables to Dictionary?--Before defining new data sets, there may be additional engineering variables to define. If not, the activity goes to Block OM-10.

Block OM-9. Procedure: Define Engineering Variable to Dictionary--The user employs the procedure presented in section A.2 to make new entries in the dictionary.

Block OM-10. Procedure: Define or Modify Data Sets Into Dictionary--The user performs the procedure presented in section A.3 to make new data set entries into the dictionary. These are unqualified at this time.

Block OM-11. List Data Sets, Restate Paths Thru Network--The user repeats the activity of Block OM-4, where he selected certain paths thru the CM network.

Block OM-12. List Input, Output and Internally Connected Data Sets--The System lists the data sets that appear as input or output at the OM boundary or are internally connected, for the paths the user has selected.

Block OM-13. Data Sets Consistent?--The user examines the information provided in Block OM-12. If there are no inconsistencies, for example a variable appearing as input that the user expected to be internally connected, the activity skips to Block OM-15.

Block OM-14. Procedure: Define or Modify Data Set Into Dictionary--The user employs the procedure presented in section A.3 to supply the needed data sets. The activity then returns to Block OM-12 to check again for correctness.

Block OM-15. Put OM Into Executable Form--The user is now satisfied that the data connections will be correct, so he requests the System to put the OM into executable form.

Block OM-16. Successful?--If the OM was successfully put into executable form, then the activity skips to Block OM-18.

Block OM-17. Display Problem, Supply Modifications--The System displays the problems of the OM, the user supplies the corrections, and the activity returns to Block OM-15 for another try.

Block OM-18. Supply Narrative, Key Words, Qualifier--The OM is now in executable form, and the user is satisifed with the data flow and CM operations along the network paths he has examined. The OM is therefore ready for entry into the dictionary and the library as a qualified item. This completes the procedure for building or modifying OM's.

A.6 PROCEDURE: BUILD OR MODIFY JOB

The job is the item that is actually executed. It is made of a network of OM's, either by building a new network or modifying an existing one. The network for this activity is shown on figure A6.

Block J-1. Build or Modify?--The user chooses whether to build (Block J-3) or modify (Block J-2).

Block J-2. Identify Job and Modify--The user identifies the job to be modified. The System then lists the OM's in the job and displays their network. The user exmaines the network and lists new or replacement OM's and gives network modification information. The System copies the original job and makes changes to the copy. The activity now skips to Block J-4.

Block J-3. Build New Job--The user lists the OM's to be used and gives their network.

Block J-4. Select Paths Thru Network, List Input, Output and Internally Connected Data Sets--The user selects paths thru the OM network of the job. The System lists data sets that appear as input, output or are internally connected for the selected paths.

Block J-5. Network as Expected?--The user examines the network for the sequence of the OM's. If the sequence is not as expected, the activity skips to Block J-7.

Block J-6. Data Sets as Expected?--The user examines the list of input, output and internally connected data sets for the chosen paths. If there are no problems, the activity skips to Block J-9.

Block J-7. Modify Network?--If there are problems, then the user must either modify the network, or exit the procedure

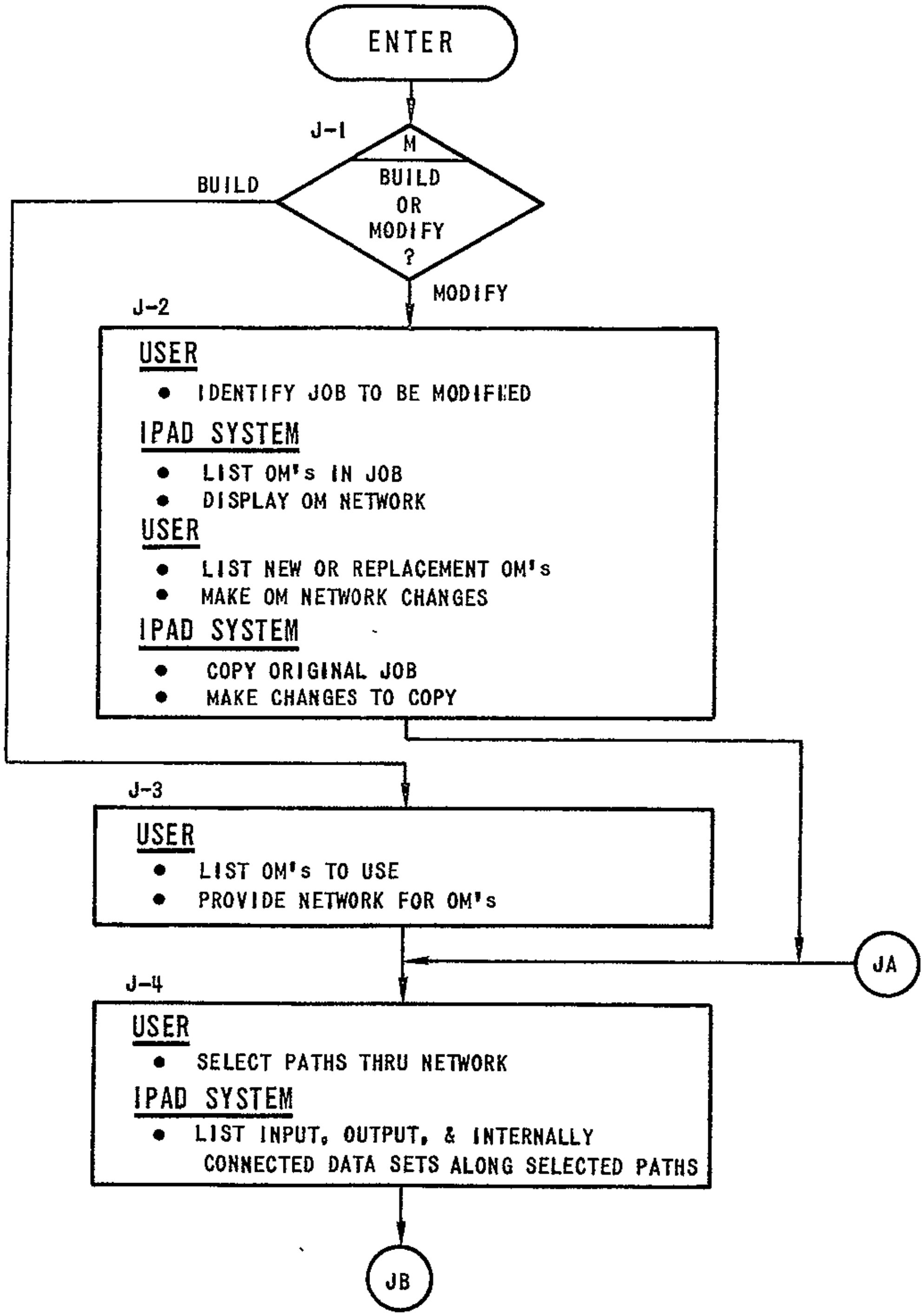


Figure A.6 Procedure: Build or Modify Job

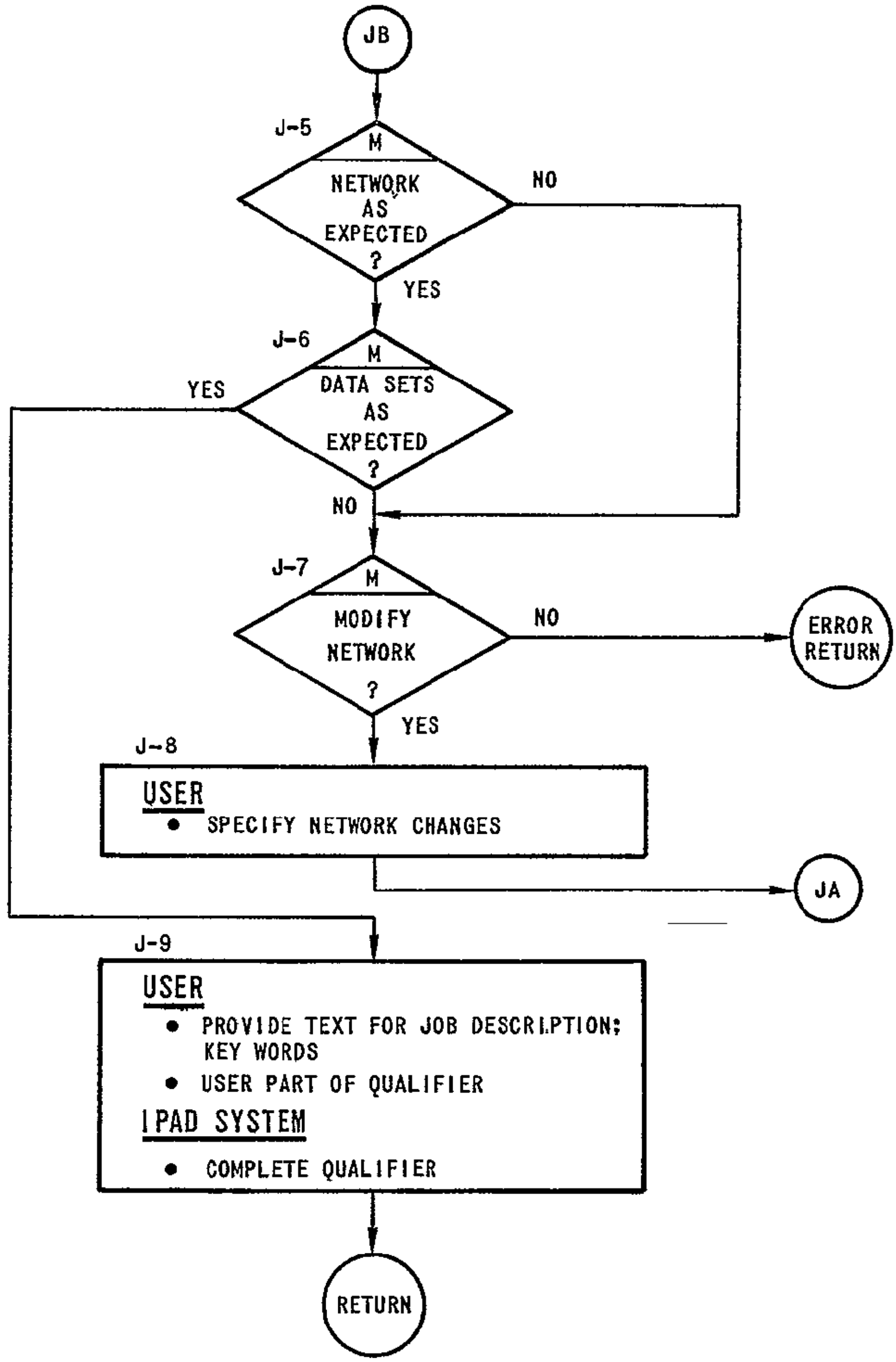


Figure A.6 (Cont'd)

and replan his job. The network modification is done in Block J-8.

<u>Block J-8. Specify Network Changes</u>--The user changes the network, and returns to Block J-4 to repeat the network checkout process.

<u>Block J-9. Supply Narrative, Key Words, Qualifier</u>--At this point, the user has determined that the job is satisfactory. He supplies the dictionary and library information, the job is qualified, then stored in the library. This completes the procedure for building or modifying jobs.