NASA Conference Publication 2254

NASA Administrative Data Base Management Systems

Proceedings of a conference held at
Jet Propulsion Laboratory
Pasadena, California
May 26-27, 1982

NASA

NASA Conference Publication 2254

NASA Administrative Data Base Management Systems

Proceedings of a conference held at
Jet Propulsion Laboratory
Pasadena, California
May 26-27, 1982

NASA

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1983

FOREWORD

On February 2, 1982, Mr. L. N. Lushina, Director NASA Information Systems Division (Code NS-11) issued a call for papers for an Administrative Data Base Management Systems (DBMS) Technology Conference to be held in May 1982 at the Jet Propulsion Laboratory, Pasadena, California. The call for papers was distributed to the members of the NASA Intercenter Committee for ADP and to NASA Headquarters Offices having an interest in the use and development of Adminstrative Data Bases. The purpose of the conference was to serve as an open forum for the constructive exchange of information among NASA technical personnel who deal with NASA Administrative DBMS needs. Because this conference would be the initial meeting of NASA's Administration Data Base implementers, the conference was to be carefully evaluated as to its benefit to the participants and as to whether or not future conferences on this subject should be recommended.

Mr. James D. Radosevich, NASA Systems Management and Evaluation Branch (Code NSD-10) was appointed conference chairman. The conference arrangements at JPL were the responsibility of Herbert D. Strong, JPL Computing and Information Services Office, 207. Ms. Kathyn Leaman, NASA Resident Office at JPL, was in charge of the conference registration.

The conference was held at the Jet Propulsion Laboratory on May 26, and 27,1982. Presentations were made by thirteen of the forty-four people who attended. In addition the attendees had the opportunity to tour the JPL Information Processing Center facilities and to attend a dinner meeting at which a presentation was made by Dr. Alfonso Cardenas ,Professor, Computer Science Department, University of California at Los Angeles. Dr. Cardenas spoke on the subject of Evolution Toward Data Base Information Systems: Challenges and Opportunities. The general reaction from the NASA attendees was that the conference was worthwhile and provided an opportunity for them to meet with and share experiences with other NASA personnel who are concerned with Administrative Data Base System issues.

CONTENTS

ADMINISTRATIVE DATABASE MANAGEMENT SYSTEMS
TECHNOLOGY CONFERENCE
MAY 26-27, 1982
JPL BLDG. 180 - ROOM 101

AGENDA

May 26, 1982

9:00 Welcome

Fred Bowen, Manager, NASA Resident Office-JPL
Dr. Rick Green, Manager, JPL Computing and Information Services Office

Opening Remarks and General Announcements
(Distribute Center Reports and Copies of Papers)

Jim Radosevich, NASA HQ Information Systems Division
Herb Strong, JPL Computing and Information Services Office

9:30 Data Bases as an Information Service (JPL/Don Vincent)

10:00 Comparison of Scientific and Administrative DBMS (JPL/Dr. Joseph Stoltzfus)

10:30 Break

10:45 User Interface to Administrative DBMS Within a Distributed Environment (MSFC/Lowell Martin)

11:15 Panel Discussion (Moderator - Jim Radosevich)

11:45 Lunch

1:00 DOCU-TEXT - A Tool Before the Data Dictionary (JPL/Barbara Carter)

1:30 Corporate Information System Development Approach (MITRE/Peter Rozett)

2:00 Break

2:30 Data Base Machines (MITRE/Malcolm Stiefel)

3:00 Panel Discussion (Moderator - Jim Radosevich)

3:30 Tour of JPL Data Processing Facilities (Woodbury Road)

4:30 Adjourn

6:30 Dinner Meeting - Speaker Dr. Alfonso Cardenas
Subject - "Evolution Toward Data Base Information Systems: Challenges and Opportunities"
Location - Peppermill Restaurant
Cost - $15 per person

May 27, 1982

9:00	Comparative Analysis of DBMS for KSC Institutional Support (CSC/Rod Smith)
9:30	RAMIS (GSFC/Jim Head)
10:00	Break
10:15	RIMS (JSC/Jon Symes)
10:45	Integrated Program for Aerospace Vehicle Design (IPAD) DBMS (NASA HQ/Sam Venneri)
11:15	Panel Discussion (Moderator - Jim Radosevich_
11:45	Lunch
1:00	Planning the Future of JPL's Management and Administrative Support Systems Around an Integrated Data Base (JPL/Mike Ebersole)
1:30	Data Base Management System - A Federated Approach (JPL/Robert Iverson)
2:00	Break
2:15	SISRS (JSC/Ken Martindale)
2:45	Agencywide Systems, NASA Administrative DBMS Goals (NASA HQ/Jim Radosevich)
3:15	Panel Discussion (Moderator - Jim Radosevich)
4:00	Adjourn

DATABASES AS AN INFORMATION SERVICE

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

D. A. Vincent
Computing and Information
Services Office
Jet Propulsion Laboratory

DATABASES AS AN INFORMATION SERVICE

D. A. Vincent
Jet Propulsion Laboratory

ABSTRACT

For database information to be valuable to a broad range of users, it is essential that access methods be provided that are relatively unstructured and natural to information services users who are interested in the information contained in databases, but who are not willing to learn and use traditional structured query languages. Unless this ease of use of databases is considered in the design and application process, the potential benefits from using database systems may not be realized.

The relationship of databases to information services, and the range of information services users and their needs for information is explored and discussed.

I. INTRODUCTION

Much of the attention directed to the subject of database management systems tends to focus more on database technology rather than on the needs and concerns of the range of potential and present users of such systems. Although this technological focus is a necessary part of bringing good database management systems into being, all of these efforts may be for naught if the human side of database utilization is neglected. How well Database Management System (DBMS) capabilities meet the needs and concerns of the people who make up the potential and present user community will determine whether or not a DBMS application is successful since it is the user need which the system must serve. Because of the importance of the special needs and concerns of the user community to the success of DBMS applications, it is essential that the nature of these users and their needs for information be understood by DBMS designers and due consideration given to these factors as part of the database system design. This viewpoint can appropriately be considered as basic requirements for the design of a database system as an information service.

II. DATABASES IN AN INFORMATION SERVICES CONTEXT

Information Services can be defined as follows:

> Information services consist of capabilities and support functions which enable and facilitate the preparation, processing, storage, retrieval and transfer of information.

This broad definition includes database as well as other services which can be accessed by users from their local office workplace. Figure 1 illustrates the relationships of user systems to information services. This simple model includes database services from the information service users

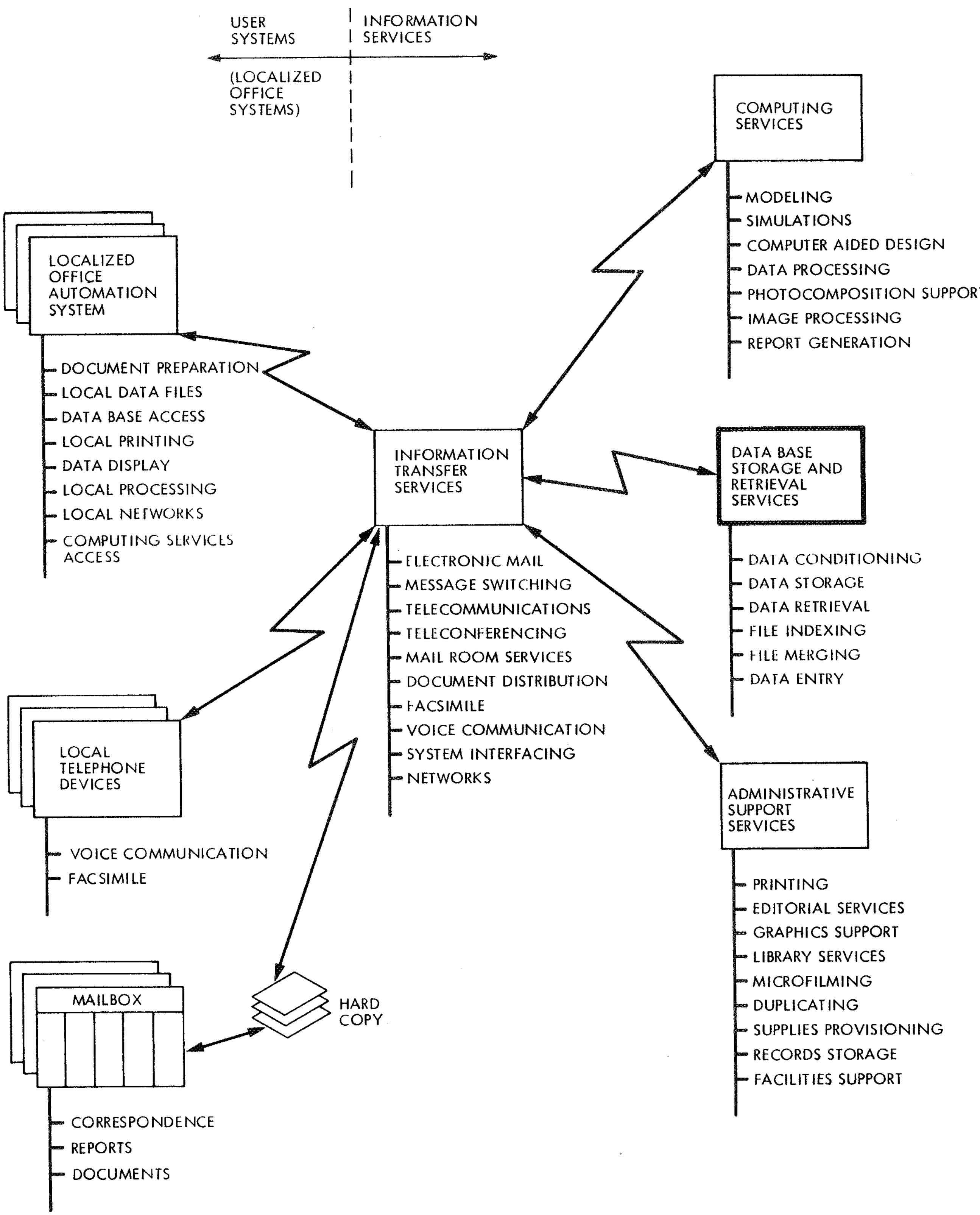


Figure 1. Relationship of User Systems to Information Services

viewpoint. Recognizing this, and realizing that the user of information services is not the typical specialist hired to work at a terminal, the DBMS designers and implementers must provide methods for access to the data and information contained in those databases that are convenient and natural to the users normal day-to-day activities. Within this context the database services should provide for one or more of the following access and/or output capabilities:

1. Terminal access to the database using natural english language queries and instructions concerning the subject area of the database.

2. Quantitative and/or graphic presentations of summary information derived from the content of the database.

3. Prompts and menus which guide the user through the steps needed to acquire explicit information to met his needs, or to allow the generation of custom listings in report form. Whenever possible this capability should be supported by touch sensitive screens, simplified pointing devices or other aids which allow users without computing skills to easily acquire information.

4. Access strategies which allow users seeking information to easily browse the file to determine if their information needs can be satisfied.

5. Features which allow users to recover easily from errors made during an access session, i.e., a high degree of user fault tolerance.

6. Provisions which allow the information services user to enter parameters and variable data that enable specific user functions to be performed. This feature may allow limited user data entry for other information services users depending on the application being satisified by the database.

7. Traditional modes of terminal or batch access to the database for use by computing specialists responding to information services user requests.

Although some of the above described capabilities may pose implementation problems, their consideration by the database designer is essential if utilization of such systems are to reach their full potential.

Database access systems providing some of these extended capabilities are presently available in the commercial marketplace.

III. THE INFORMATION SERVICES USER

The information services user of databases can range from top level managers seeking summary and trend information that can improve their ability to make good decisions on business matters to the machine operators who perform specific structured tasks to produce repetitive reports and documents. The requirements for database access of this wide range of users can be very different indeed. Figure 2 illustrates the range of users vs the uses for information derived from database systems.

One can readily recognize that information services users at the upper level of the pyramid whose need for information is high will most likely be the least equipped to use structured query languages because their interaction with the database systems will tend to be sporadic and penetrate many subject areas. On the other hand, machine operators and data specialists near the lower levels of the pyramid will be most likely to interact frequently with a given database and thus develop a greater degree of skill in using structured access languages.

Specifically, characterization of the information services user can lead us to a better understanding of the reasons that access to information contained in databases must be provided for in ways that accommodate the day-to-day working needs of these users.

1. Managers

The managerial user of database information is most often concerned with broad areas of responsibility. His needs for information will usually range from very specific kernels of information about a discrete subject area to summary, trend and relational information derived from the database content. In general, users at this level will not directly work at a terminal unless the information they seek is easily and conviently available to them without a significant commitment of their time to learn how to use the system. In most cases the information will be acquired for them by professionals and intermediaries who package the data for their use.

As professional workstations become more prevalent and the managerial user can easily and simply acquire database information in terms of his day-to-day concerns, direct use of databases by managers can be expected to increase.

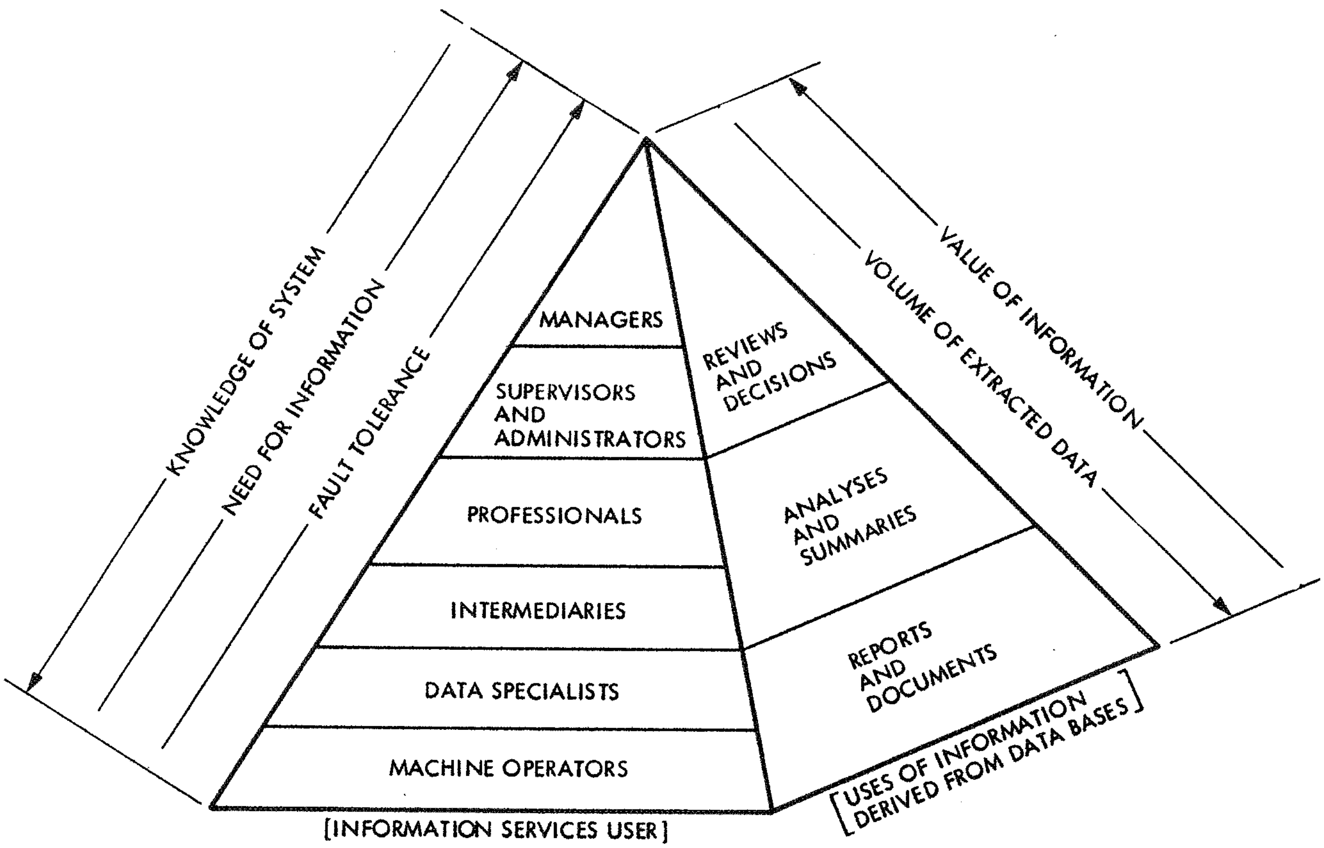


Figure 2. Types of Information Services User vs Uses of Information Derived From Data Bases

2. Supervisors and Administrators

The supervisory and administrative users of database information can be expected to be more focused on a given area of concern than managerial users. Their needs can be expected to range from acquiring specific kernels of information to comprehensive reports about a database subject area. In some instances this type of user may require trend or graphic information to respond to planning needs or for presentation to the managerial level.

Generally this user is not likely to work directly at a terminal using structured query languages. However, if access to the database information is made easy and in terms of his day-to-day concerns the likelihood of his directly accessing the database is increased. In most cases this user will use database information provided to him by professionals and intermediaries who acquire the data outputs which he requests.

3. Professionals

The professional user is probably the highest level user who will be motivated to become highly skilled in using structured query languages to acquire database information. This motivation often stems from the fact that his needs for information require that he interact with the database system to perform his assigned tasks. In spite of this motivation to use structured query languages he can benefit greatly from an ability to access the database system in terms natural to his everyday work concerns, and by being able to acquire trend, relational and graphic representations of the information in the database.

The professional user can be expected to have a variable need for database information depending on the tasks which he performs. Under these conditions his ability to maintain a high degree of skill in using structured query languages may be difficult to maintain at a high level. Accordingly, database access methods which are natural to his everyday working conditions can increase his overall working effectiveness.

4. Intermediaries

Intermediaries can be administrative staff or secretarial users whose function is to provide support to managerial, supervisory and professional workers. In this role their need for information from databases is dictated by their principal's specific needs. In some cases this user will repackage information acquired from database systems to meet a particular request from the principals which they support. Their modes of access to the database system can be by direct terminal access, or through interactions with data specialists and/or machine operators. In some cases intermediaries learn the required query languages and develop the information summaries required by the principals they support. As with other users, intermediaries can benefit from access methods which are natural to their day-to-day activities.

5. Data Specialists

The data specialist generally uses the database system to provide services to other users. In this role, he is usually familiar with structured query languages and with the details of the database system structure. He is also likely to use the full range of capabilities of the database system to provide services to the larger family of users. In some instances the data specialist is involved with maintaining the system and expanding its capabilities to make it more useful to the larger family of general users.

6. Machine Operators

The machine operator user of database systems performs machine or terminal operation tasks in response to specific instructions received from other users of the system. The level of knowledge of this type of user is primarily that needed to respond to requirements levied by information services users or data specialists. Products that result from the efforts of the machine operator are often reports and documents derived from the database content. In some instances, the machine operator performs data entry tasks and assists data specialist efforts to improve the capabilities of the database system for the community of information services users.

IV. Conclusions

As office systems capable of interacting with computers become more common to the office workplace, the potential number of users of databases as an information service will increase. Many of these information services users who need database information will not be trained in computing technology and may not be willing to learn and use traditional structured query languages. Therefore, it is essential that the capabilities of database systems include access methods which allow those information services users to acquire information in ways that are natural to their everyday working concerns.

Unless this ease of use of databases is considered in the design and applications process, the potential benefits from using database systems may not be realized.

COMPARISON OF SCIENTIFIC AND ADMINISTRATIVE DATABASE MANAGEMENT SYSTEMS

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

Joseph C. Stoltzfus
Jet Propulsion Laboratory

COMPARISON OF SCIENTIFIC AND ADMINISTRATIVE DATA BASE MANAGEMENT SYSTEMS

One of the difficult choices for the administrator of an information system is the selection of a data base management system to support a set of applications. Since it is in general impractical even to test a variety of DBMS packages, the decision must be made based on comparison of established requirements, the often confusing and incomplete technical information supplied by vendors, and experience with other applications. Although each vendor attempts to make a truly general-purpose DBMS, every available DBMS product is a compromise which supports some applications better than others.

The requirements for the data base management system to support a particular application depend on the fundamental logical structure of the data. This paper identifies some characteristics found to be different for scientific and administrative data bases, and details some of the corresponding generic requirements for the DBMS.

The requirements discussed here are especially stringent for either the scientific or adminisrative data bases. For some, no commercial DBMS is fully satisfactory, and the data base designer must invent a suitable approach. For others, commercial systems are available with elegant solutions, and a wrong choice would mean an expensive work-around to provide the missing features.

Security is a matter of special concern to the manager of an administrative data base. Personal and financial data of varying degree of confidentiality must be stored and access must be controlled. In an employee record (for example) certain items (name, phone number) may be available to all, while salary or insusrance data would be restricted to a few users. The right to update (change) information must be even more closely controlled, and accidental changes could be as serious as malicious updating.

The requirement on the DBMS is that permissions be controlled for single records and even single items in a record. And it must be impossible to construct confidential data from open information (such as finding a salary from the total in a group of two). Control of permissions for single records and items is provided by some commercial DBMS packages, other controls must be part of the data base design.

In contrast, for scientific data systems a simple password system is often adequate, access is controlled for a large block of data, and stolen data is a rare occurrance.

Automatic Recovery after system failure is important for many administrative DBMS applications, since an organization may be unable to function without access to the data base. Although there are important exceptions, temporary failure of a scientific data base can be corrected by human intervention.

The Amount of Data Retrieved by a single query is likely to be large for a scientific data base. A meteorologist might well request the wind vectors in the North Atlantic for a certain day and receive hundreds of values measured by the Seasat scatterometer. A request for names of all employees at JPL would be unusual in an administrative DBMS. Since the cost of retrieving so much data (or more) might be excessive, the DBMS is required to provide some means of estimating the cost in advance. However, the user would probably accept a delay of several minutes or longer for the response.

A typical query to an administrative data base might well retrieve a single record, such as the current obligation of account 356-ABCDE. For this small amount of data, the time to get a response would be important to the user, but the cost would be small.

Updates are a common activity in an administrative DBMS, and maintaining currency in the data base is a high priority. Delays in posting charges to an account make control difficult, and for certain applications such as banking can cause serious losses. A DBMS should make prompt entry of updates easy and provide integrity checks to minimize error.

The updates in a scientific DBMS are relatively rare, and the principal investigator or data base administrator responsible for a data set might disallow updates altogether.

We use the term Homonymns to describe the similar or identical words used to denote scientific concepts. The term "altitude" may mean geometric height above the ground, or above the earth's susrface, or altitude determined by atmospheric pressure, or temperature, or density. "Temperature" may might mean water or air temperature, air temperature may mean as measured by a thermometer at the earth's surface or remotely sensed, if measured by thermometer different instruments may have different calibrations of little or great significance. Other kinds of homonymns arise from the use of different units, such as fractional days vs. hours, minutes, and seconds, or latitude in minutes or fractions.

Such homonyms are common in science data bases, and a DBMS might support such variations in terminology by "wild cards": *temperature* is used by one DBMS to select any character string containing the word temperature. This is not to suggest that terms are ill-defined in scientific data bases, but rather that nuances of terminology are common, and the users require a means of selecting data with various similar meaning.

In administrative data bases, homonymns rarely occur, since the data base administrator will be careful to avoid any possible confusion in meaning. "Checkbook balance" has a well-defined meaning, and variations would hardly be tolerated.

The existence of Continuous rather than Discrete variables is common in scientific data bases. Of course continuous variables occur in administrative data bases as well, but it is usual (for money amounts) to round off to whole cents or whole thousands or even millions of dollars. Even then, in administrative data bases the keys used for retrieval are usually discrete, such as account number or name.

In a scientific data base, it is often desirable to use a continuous variable as a key (e.g. time). This leads to unpredictable results when certain queries are formulated, e.g. "where time=36.33min" is not the same as "where time=36:20 minutes". At least one vendor plans to include a time data type in a DBMS, but at present there is no simple solution.

Spatial Relationships are an important part of scientific data bases, but are not managed easily in general-purpose DBMS. Concepts such as "adjacent", "overlap", "nearby", "part of", "within" are used to define data retrieval, but are not supplied as standard built-in functions. These concepts are even more difficult to implement for three dimensions.

Functional Dependencies between two or more items in a data base occur in scientific data bases, particularly for position and time. The most important example is the data from satellites, where the position of the satellite at successive times of measurement is determined by the laws of motion. In principle, including both position and time in the data base is redundant, however users commonly select data either by time or position, and no commercial DBMS supports the transformation.

A final difference between scientific and administrative DBMS is the need for Restructuring the Data Base. For scientific data, the logical relationships among the data is just what the scientist wishes to determine. The types of data emphasized in different investigations will be changed, and at least the conceptual view of the data will change to match new understandings. A DBMS can be selected which facillitates changes of conceptual view or even actual physical storage structure.

Again in contrast, changes in logical relationships in an administrative data base will be avoided, because the relationships are a matter of policy decision rather than discovery. Thus a DBMS can be chosen which is optimized for other features than flexibility.

In administrative data bases, homonymns rarely occur, since the data base administrator will be careful to avoid any possible confusion in meaning. "Checkbook balance" has a well-defined meaning, and variations would hardly be tolerated.

The existence of Continuous rather than Discrete variables is common in scientific data bases. Of course continuous variables occur in administrative data bases as well, but it is usual (for money amounts) to round off to whole cents or whole thousands or even millions of dollars. Even then, in administrative data bases the keys used for retrieval are usually discrete, such as account number or name.

In a scientific data base, it is often desirable to use a continuous variable as a key (e.g. time). This leads to unpredictable results when certain queries are formulated, e.g. "where time=36.33min" is not the same as "where time=36:20 minutes". At least one vendor plans to include a time data type in a DBMS, but at present there is no simple solution.

Spatial Relationships are an important part of scientific data bases, but are not managed easily in general-purpose DBMS. Concepts such as "adjacent", "overlap", "nearby", "part of", "within" are used to define data retrieval, but are not supplied as standard built-in functions. These concepts are even more difficult to implement for three dimensions.

Functional Dependencies between two or more items in a data base occur in scientific data bases, particularly for position and time. The most important example is the data from satellites, where the position of the satellite at successive times of measurement is determined by the laws of motion. In principle, including both position and time in the data base is redundant, however users commonly select data either by time or position, and no commercial DBMS supports the transformation.

A final difference between scientific and administrative DBMS is the need for Restructuring the Data Base. For scientific data, the logical relationships among the data is just what the scientist wishes to determine. The types of data emphasized in different investigations will be changed, and at least the conceptual view of the data will change to match new understandings. A DBMS can be selected which facillitates changes of conceptual view or even actual physical storage structure.

Again in contrast, changes in logical relationships in an administrative data base will be avoided, because the relationships are a matter of policy decision rather than discovery. Thus a DBMS can be chosen which is optimized for other features than flexibility.

The Conclusion of these comparisons is that selection of a DBMS must be based on the requirements for the information system. There is no unique distinction between scientific and administrative data bases or DBMS. The distinction comes from the logical structure of the data, and understanding the data and their relationships is the key to defining the requirements and selecting an appropriate DBMS for a given set of applocations.

USER INTERFACE TO ADMINISTRATIVE DBMS WITHIN A DISTRIBUTED ENVIRONMENT

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

Lowell D. Martin
NASA/George C. Marshall
Space Flight Center

Robert D. Kirk
Boeing Computer Support
Services, Inc.

ABSTRACT

A Data Base Management System (DBMS) has been implemented for the Communications Office of MSFC to control and report on Communication Leased Service Contracts issued by this office. The system user executes online programs to update five files residing on a UNIVAC 1100/82, through the forms mode features of the Tektronix 4025 terminal and IMSAI 8080 microcomputer. This user can select the appropriate form to the Tektronix 4025 screen, and enter new data, update existing data, or discontinue service. Selective online printing of 40 reports is accomplished by the system user to satisfy management, budget, and bill payment reporting requirements.

REQUEST FOR COMPUTER SYSTEM

In the early part of 1978 the MSFC Communications Office requested that a computer system be established to develop a method of automating their recordkeeping process of tracking all Communication Leased Service Contracts under their responsibility, and the requesting and approval of certified funds from the Financial Management Office (FMO). A feasibility study was conducted, and it was determined that a computer system could be established to eliminate most of their manual operations necessary to track Communication Leased Service Contracts, obtain certified funds from FMO, and produce reports from the data base to assist them in their daily activities. Approval was received from the MSFC Computer Services Office to proceed on a computer system. The resulting system is called Communications Cost Tracking (CCT) System.

IMPLEMENTATION PLAN

Plans were made to develop a computer system in three phases to fulfill user requirements. These phases have all been implemented and are as follows:

- Phase I - Establish the data base and produce five reports. This would be accomplished on a batch-run basis.
- Phase IA - Develop three interactive online update programs and provide for online printing of reports.
- Phase II - Generate an accounting file from the updated master file, automate the bill payment process, and produce management and budget reports.

PHASE I IMPLEMENTATION

Phase I consisted of developing an update program using the free-form method of input data preparation by the system user. The updated master file from this program was used by a separate funding program to extract those records which required certified funds to be approved by the FMO. A report program used these extracts to print the Communication Service Authorization (CSA) Form (MSFC Form 461). See Figure 1. Four other programs produced reports from the updated master file to satisfy user reporting needs for their daily activities. This phase was implemented in October 1978.

PHASE IA HARDWARE SELECTION

Phase IA consisted of the development activities required to implement three interactive online update program using a forms mode update feature, and the online printing of all reports from this system at the user's site. The initial research effort revealed that no software was available to aid in our development activity. The initial development activity involved the selection of two identical sets of hardware which would be compatible for forms mode processing and online printing of reports. To accommodate forms mode processing, the hardware must include a smart terminal. A Tektronix 4023 terminal was initially reviewed, but it was determined that this terminal would not fully satisfy our forms mode requirement. A Tektronix 4025 terminal (CRT) with 64K memory was then reviewed and it was determined that this terminal would satisfy our forms mode needs.

The next hardware to be selected was a microcomputer to interface and handle all communication traffic between the host computer and the Tektronix 4025 terminal with capability to support two printers. Partly because of availability, but mostly because of compatibility, it was determined that the IMSAI 8080 microcomputer with a dual floppy disk could be effectively used

FIGURE 1.

COMMUNICATION SERVICE AUTHORIZATION

PAGE 1 OF 1

TO:	1 FROM:	CLASSIFICATION	ACTION
AMERICAN SATELLITE CORP ATTN: WILLIAM T RANDALL DIR C&P 1801 RESEARCH BLVD ROCKVILLE MD 20850	NATIONAL AERONAUTICS AND SPACE ADMINISTRATION GEORGE C. MARSHALL SPACE FLIGHT CENTER Marshall Space Flight Center, Alabama 35812	2 ROUTINE	3 NEW FY

AUTHORITY FOR A COMMUNICATIONS COMPANY TO ESTABLISH OR PERFORM SERVICE FOR OFFICIAL USE

COMMUNICATION SERVICE AUTHORIZATION		BILLING NUMBER: 6	AUTHORITY
CONTRACT NO.: 4 H0006361C	ISSUE DATE: 5 10/01/81	958265	National Aeronautics and Space Act of 1958, P.L. 85-568 (42 U.S.C. 2451 ET SEQ.). Approved July 29, 1958.
MODIFICATION NO.: 7 000	EFFECTIVE DATE: 8 10/01/81	SERVICE DATE: 9 09/25/80	MILEAGE: 10 00000

In accordance with established Federal, State or other regulatory Commission Tariffs, of which this authorization forms a part, authority is hereby given to establish or perform services as prescribed below for:

GEORGE C. MARSHALL SPACE FLIGHT CENTER
Marshall Space Flight Center, Alabama 35812

ITEM NO.	DESCRIPTION OF SERVICE	QUANTITY 11	USOC 12	RATE PER MONTH PER UNIT	RATE PER MONTH TOTAL
	INSTALLATION, RENTAL, & MAINTENANCE OF COMMUNICATIONS SERVICE & FACILITIES FOR PERIOD OCTOBER 1, 1981 THROUGH SEPTEMBER 30, 1982 IN ACCORDANCE WITH FEDERAL OR STATE APPROVED TARIFFS.				
	FROM: MSFC 453 CENTREX AL TO: RKWL B 1 RM AA-17 CANOGA PARK CA				
1.	SATELLITE CIRCUIT	1	ASCKT1	759.93	759.93
2.	CUSTOMER PACKAGE	1	CUSTPACK	56.00	56.00
3.	LOCAL LOOP FACILITIES	1	LLF	37.00	37.00
			TOTAL MONTHLY	CHARGE	852.93
			TOTAL	CHARGE	$852.93

CIRCUIT FILE INFORMATION DCN V-2-AN-06361-010 S01

REQUESTOR: 13	PURPOSE OR USE: 14	TYPE OF CIRCUIT: 15	CONFIGURATION: 16	MODULATION RATE: 17	Conditioning: 18	Restoration Priority: 19
JSC	FAX NET	FAX	SATELLITE	4.8 KBS	00	00

SUBMIT BILLS FOR SERVICE TO: AN01
DIRECTOR, COMMUNICATIONS OFFICE
GEORGE C. MARSHALL SPACE FLIGHT CENTER
Marshall Space Flight Center, Alabama 35812

CORRESPONDENCE REFERENCE 20: 958265-R029-3-4345

REMARKS: 21 HIGH SPEED FACSIMILE

LOCAL CONTACT FOR DETAILS:

ACCOUNTING AND APPROPRIATIONS DATA: 22
308-01-00-3975-AN-2-004-139-2321

DCN: 23
V-2-AN-06361-010 S01

INTERNAL REFERENCE
CSAR NO.: 24 TELEPHONE ORDER NO.: 25

OFFICE MAKING PAYMENT: BF52
FINANCIAL MANAGEMENT OFFICE
GEORGE C. MARSHALL SPACE FLIGHT CENTER
Marshall Space Flight Center, Alabama 35812

DISTRIBUTION: 26
3 COPIES AN01 (1 SIGNED ORIGINAL & 2 COPIES)
2 COPIES VENDOR
1 COPY BF40/J. CAMPER (SIGNED ORIGINAL)
1 COPY BF20/J. PALMER
1 COPY RESOURCE OFFICE
1 COPY PSCRD
1 COPY TCC

ACCEPTED (Name of Contracting Firm): 27 AMERICAN SATELLITE CORP

SIGNATURE OF CONTRACTOR REPRESENTATIVE: 28 DATE: 29

SIGNATURE OF AUTHORIZING OFFICIAL: 30

CONTRACTING OFFICER: 31 E.D. HILDRETH

MSFC - Form 461 (Rev. February 1982) (PART A)

FOR INTERNAL MSFC USE ONLY

TO: 32
AMERICAN SATELLITE CORP
ATTN: WILLIAM T RANDALL DIR C&P
1801 RESEARCH BLVD
ROCKVILLE MD 20850

COMMUNICATION SERVICE AUTHORIZATION		BILLING NUMBER: 35
CONTRACT NO.: 33 H0006361C	ISSUE DATE: 34 10/01/81	958265
MODIFICATION NO.: 36 000	EFFECTIVE DATE: 37 10/01/81	SERVICE DATE: 38 09/25/80

FACILITIES COST: 39	MILEAGE COST: 40	NON-RECURRING COST: 41	TOTAL COST: 42
$852.93	$.00	$.00	$852.93

ACCOUNTING AND APPROPRIATIONS DATA: 43
308-01-00-3975-AN-2-004-139-2321 MISC COST $.00

DCN: V-2-AN-06361-010 S01 EST YEARLY COST $10,235.16 44

CERTIFICATION OF FUNDS: 45
I certify that funds in the amount of $5,117.58 are available under appropriation 802/30108 to cover the items or services listed on this document.
PREV FUNDS $5,117.58 TOTAL FUNDS $10,235.16

FOR: DIRECTOR, FINANCIAL MANAGEMENT OFFICE DATE

MSFC - Form 461 (Rev. February 1982) (PART B) to handle this interface. The next hardware item to be selected was a GE-340 printer. This printer was compatible with the hardware already selected and was capable of print speeds of up to 300 lines per minute, thus providing the printing speed for online printing requirements at the user's site. Two printers would be utilized, one to print reports on regular computer paper, and the second to print the CSA form. This would allow the system users to fulfill their printing needs without changing paper and would also provide in-house backup printer capability.

The last hardware item to be selected was a Tektronix 4631 Hard Copy Unit. This provided the system users with a means of deriving hard copy output of the contents currently contained on the Tektronix 4025 screen. This would also be helpful to the system programmers for obtaining hard copies of the screen for later resolution of problem areas.

With the equipment selection now finalized, a 9.6K baud multipoint communication line was selected with two drops. This would provide the system users with the data access and computer turnaround necessary for responsive interaction between their terminals and the host UNIVAC 1100 computer.

PHASE IA SOFTWARE DEVELOPMENT

The software development necessary to create and drive the interactive online forms mode update programs and to provide for online printing of reports started in September of 1978. This consisted of three distinct parts: development of an Interactive Communication Module (ICM) to establish an operating system for the IMSAI 8080; development of forms mode forms for the online update process; and development of COBOL programs for the online forms mode update and online print programs. The first item was the development of ICM for the interface operation of the IMSAI 8080. This was necessary for two reasons. First the IMSAI 8080, had to configure the communication traffic to and from the host computer to the Tektronix 4025 terminal emulating UNIVAC 200 protocol. Secondly, the IMSAI 8080 would be required to decipher and intercept special communication messages from the COBOL programs on the host computer. This would allow for the following:

- o Form could be drawn on the Tektronix 4025 screen.
- o Proper spacing for printed reports would be determined.
- o Booting and rebooting of this operating system on the IMSAI 8080 to connect each terminal to the host computer would be established.
- o Routing and processing of specific communication traffic to the proper hardware device (CRT, printer, or disk) would be accomplished.

The second item was the actual development of the forms which would be required for online updating. Three forms were necessary because three different files would be involved. The first form (see Figure 2) was developed to update the master file. This form required the ability of the Tektronix 4025 terminal to scroll the form up or down on the screen. This was necessary because the size of this form was larger than the screen size of the Tektronix 4025. The second form (see Figure 3) was developed to update the Vendor Address file and required no scrolling on the Tektronix 4025 screen. The third form (see Figure 4) was developed to update the Uniform Service Order Code (USOC) file and required no scrolling on the Tektronix 4025 screen.

FIGURE 2.

COMMUNICATIONS SERVICE AUTHORIZATION

CONTRACT	ISSUE DATE / /	BILLING NO
MOD NO.	EFFECTIVE DATE / /	SERVICE DATE / /
MILEAGE	CLASSIFICATION	LAST ACTION
LOCATION-FROM:		
LOCATION-TO:		
REQUESTOR	PURPOSE	END DATE / /
TYPE CIRCUIT	CONFIGURATION	
MODULATION RATE	CONDITIONING	RESTOR PRIORITY
CORR REF	TELEPHONE ORDER NO - - /	
REMARKS	CSAR NO -	
ACCOUNTING CODE - - - - - - - -	FUND CONTROL	
DCN - - -	MISC COST $.	FAC COST $.
MILAGE COST $.	NON-RECURRING COST $.	DEL-SW
TOTAL COST $.	AMOUNT CERTIFIED $.	VENDOR
EST YEARLY COST $.	TOTAL YTD FUNDED $.	

USOC	QTY	USOC	QTY	USOC	QTY	USOC	QTY
				MORE USOCS			

VALID ACTION CODES ARE:

0 = MENU	3 = DELETE	6 = CHG EFF DATE	9 = DISPLAY
1 = INSTALL	4 = ACC CHG	7 = BAD REC	10 = STOP
2 = MODIFY	5 = NEW FY	8 = REC CHG	11 = NEXT RECORD

FIGURE 3.

```
                 CSA VENDOR ADDRESS

          VENDOR CODE

          SERVICE ADDRESS
          LINE 1

          LINE 2

          LINE 3

          LINE 4

          BILLING ADDRESS
          LINE 1

          LINE 2

          LINE 3

          LINE 4


>
VALID ACTION CODES ARE :
 A = ADD          C = CHANGE
 D = DELETE       L = LOCATE & DISPLAY
 S = STOP
ENTER ACTION CODE
>_
```

FIGURE 4.

CCT SYSTEM USOC TRANSACTION FORM		
ACTION CODE	VENDOR CODE	EFFECTIVE DATE
USOC	COST	DESCRIPTION

```
>

VALID ACTION CODES ARE :
 A = ADD          C = CHANGE
 S = STOP         X = NEXT RECORD
 D = DELETE       L = LOCATE & DISPLAY
ENTER ACTION CODE
>
_
```

After these three forms were developed, they were written onto a floppy disk on the IMSAI 8080. This would allow for the accessing and drawing of these forms from the floppy disk onto the Tektronix 4025 screen, based on the updating requirements of the system user.

The third item was the development of the COBOL programs and one FORTRAN program required for the online updating of the above three files, and the online printing of reports. Each file to be updated required a separate program due to MSFC requirements restricting programs used in terminal processing to a maximum of 45K memory on the host computer.

Incorporated into each of the three update programs were all applicable Tektronix 4025 commands required for forms mode updating. These commands were used by the Tektronix 4025 to establish: a work area for the forms; a monitor area for communication with the system user; the commands required for forms mode processing; and the positioning of the cursor in the form. The applicable form was drawn on the screen by passing the form number to the IMSAI 8080 from each update program being executed by the system user. The IMSAI 8080 would recognize this communication traffic as a command for drawing the form on the Tektronix 4025 screen. Control would then be passed to a FORTRAN program residing on disk in the IMSAI 8080 disk unit. This program would draw the form and then turn control back to the COBOL update program being executed.

After the desired data has been entered into the form by the system user, COBOL coding in each update program allows for a full-screen transmission of the data contained in the form back to the host computer for processing, editing, and updating of the files. After editing is completed, the data is transmitted from the host computer back to the Tektronix 4025 screen for verification or correction of data prior to writing the data to the file on the host computer. This allows for correction of erroneous data by the system user.

The existing report generators were modified and all subsequent programs were prepared to accept special form-feed and vertical tab characters from the report program being executed. These characters would be intercepted by the IMSAI 8080, and would be passed directly to the GE-340 printer. In this way, it was possible to skip to new pages or to special vertical tabs positioned in implicit lines within a report. This made it possible to print reports online on the GE-340 printer at the system user's site.

A special user FORTRAN program was written to work in conjunction with ICM. This program established the function keys on the Tektronix 4025 to provide for full-screen transmission, to allow positioning of the cursor to lines within a form, to provide Site-ID codes, and to produce hard copy output of the screen. These keys are used by the system user to provide for the above functions by merely depressing a key.

PHASE IA IMPLEMENTATION

All of the software for the system had been written, compiled, and was ready for system testing. The first item to be accomplished was the creation of index sequential files for the three files used by this system. A small COBOL program was written to accomplish this task. Index sequential files were selected because they could be read randomly by the three online update programs, or read sequentially by the report programs in this system.

The next item was to obtain ICM in a relocatable format. This relocatable was linked with the relocatable from the user FORTRAN program using standard IMSAI procedures. This provided for the loading of an IMSAI 8080 operating system for each of the two sites involved. System testing was started, utilizing forms mode updating, full-screen transmission of data from the form to the host computer and the online printing of reports. Because of the complexity of these items, systems testing required approximately 3 months to completely check out the 10 programs contained in the system. During this system test period, the system user participated actively to obtain the necessary training required to run this system. Also, the system user was familiar with the data contained in the three files of this system, thus making it much easier to validate test results from system testing.

Parallel runs were made on the online and batch-run systems for approximately 1 month. This was done to ensure system integrity. After these parallel runs were completed, the system user approved the online system, and started exclusive use of the online updating and reporting system. Phase IA was implemented in March 1979.

PHASE II SYSTEM ENHANCEMENTS

Through the normal daily updating and reporting process, the system user requested that system changes be implemented for ease of operation. This included using standard dates and the system changes which would let the computer accomplish the routine tasks of specific updating tasks originally placed on the system user. These changes were implemented, and many other changes have since been implemented to make the system more user-friendly.

Development work was now begun to construct and update an accounting file from the latest master file. The accounting file would be a historical file of all activity that occurred during each fiscal year, and would be utilized for the preparation of bill payment and other budget reporting needs.

A new forms mode form (see Figure 5) was prepared in conjunction with a new online update program. This would allow the system user to manipulate the bill payment switches to satisfy current bill payment criteria. In this manner, selective bill payment by each carrier involved could take place. Also, adjustments to monthly bills could be easily processed by the system user.

Once the accounting file was implemented, system work commenced on other enhancements. The first enhancement implemented was the development of an automated bill payment program. This program is used to pay the specific monthly bills for each carrier involved, and produce reports to be used by the FMO to make the actual payment of these bills. These bill payment reports have replaced the MSFC Form 1575 previously prepared on a manual basis by the Communications Office, and are used by the FMO for payment of monthly bills.

A second enhancement was implemented to provide the Communications Office with the ability to process data for all NASA Centers located throughout the United States. The current master file now contains data on these records, and CSA forms are produced for the carrier and for the funding of these items.

A third enhancement implemented was the conversion of the Communications Order System from a batch process to an online updating and reporting process. A new form (see Figure 8) was prepared for this updating function. This system maintains an inventory on communication supplies and equipment and is used in the preparation of budget reports for the Communications Office.

FIGURE 5.

COMMUNICATIONS SERVICE ACCOUNTING					
CONTRACT NO. [redacted]		BILLING NO. [redacted]		MOD NO. [redacted]	
MONTH	MONTHLY-COST	PAY-STATUS	MONTH	MONTHLY-COST	PAY-STATUS
OCT	$[redacted]	[redacted]	NOV	$[redacted]	[redacted]
DEC	$[redacted]	[redacted]	JAN	$[redacted]	[redacted]
FEB	$[redacted]	[redacted]	MAR	$[redacted]	[redacted]
APR	$[redacted]	[redacted]	MAY	$[redacted]	[redacted]
JUN	$[redacted]	[redacted]	JUL	$[redacted]	[redacted]
AUG	$[redacted]	[redacted]	SEP	$[redacted]	[redacted]
YRLY COST $[redacted]		YTD FUNDS $[redacted]		VENDOR [redacted]	

PAY STATUS
1 = TO BE PAID 2 = PAID 3 = NO ACTION
M = DISPLAY MENUE OF ACTION CODES

VALID ACTION CODES ARE:
M = MENU D = DISPLAY
C = RECORD CHANGE N = NEXT RECORD
S = STOP
ENTER ACTION CODE _

FIGURE 6.

COMMUNICATIONS ORDER TRACKING

ORDER NUMBER

WORK ORDER CODE

DCN

ESTIMATED AMOUNT $

ACTUAL AMOUNT $

REMARKS

FORM30

VALID ACTION CODES ARE:

A = ADD D = DELETE

C = RECORD CHANGE N = NEXT RECORD

M = MENU DS = DISPLAY

S = STOP

ENTER ACTION CODE

>_

A fourth enhancement implemented was the modification to the automated bill payment programs to provide for the consolidation of monthly bills from one account per page to up to 36 accounts by account number, on multiple pages where necessary, for each carrier involved. This greatly reduced the amount of printed output going to the FMO on a monthly basis, and provided the FMO with a more orderly and efficient means of paying these monthly bills. Also, a pre-edit program was implemented to provide the system users with the ability to verify bills to be sent to the FMO prior to the actual payment. Another change implemented into these programs ensured the year-to-date amount of paid bills did not exceed the total amount funded for the fiscal year. This eliminated making corrections to bills already processed for payment by the system user, and prevented erroneous bill payment data from going to the FMO for payment.

A fifth enhancement was the implementation of approximately 35 additional report generators to assist the system user in support of daily operations, and the Communications Office in budget preparation. Any or all of these reports can be produced from the current master file by the system user on an as-needed basis. The budget reports can be produced from either the current master file or the current accounting file.

PLANNED FUTURE SYSTEM ENHANCEMENTS

Some of the planned enhancements for the future are:

- Generation of additional budget reports.
- Preparation of the Program Operation Plan (POP) budget for the Communications Office.
- Provision of graphics capability for special reporting.
- Provision for hard copy output of graphics at the user site.

Planned future enhancements are not limited to the above items. However, these are only those major items which have been requested by the system user.

SUMMARY

Since the inception of this system, it has always been the intent to provide the system users with a means of accomplishing their tasks in the most efficient manner, with the least amount of manual intervention. The system now in operation provides the system users with a means of saving many labor hours over the old manual method. The number of records contained in the current master file have almost tripled since the original master file was established in 1978. Processing these extra records would have required additional man-hours under the old manual system. The current online system workload is being processed with less manual effort than when the system was implemented in 1978.

Some of the major labor savings features of this system are:

- o Computer programs now establish a new Fiscal Year master file from the previous Fiscal Year's master file.
- o User requested changes to the Accounting Code for the new Fiscal Year master file are easily processed.
- o Special funding runs are made for requesting specific funds to be certified by the FMO.
- o Use of the CSA Form eliminates the need of a MSFC Form 404 to be manually prepared by the Communications Office on Communication Leased Service Contracts for requesting certification of funds.

- The Consolidated Bill Payment Report saves time and labor hours for both the Communications Office and the FMO.

- Budget reports currently being produced saves labor hours during budget preparation activities by the Communications Office.

DOCU-TEXT - A Tool Before the Data Dictionary

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

Barbara Carter
Jet Propulsion Laboratory
FTS - 792-3851

I. WHAT IS DOCU-TEXT?

DOCU-TEXT is a proprietary software package developed by Diversified Software Systems, Inc. This product aids in the production of documentation for a data processing organization and can be installed and operated only on IBM computers. In organizing information that ultimately will reside in a data dictionary, DOCU-TEXT has proven a useful documentation tool in extracting information from existing production jobs, procedure libraries, system catalogs, control data sets and related files.

DOCU-TEXT reads these files to derive data that is useful at the system level. The ouput of DOCU-TEXT is a series of user selectable reports. These reports can reflect the interactions within a single job stream, a complete system, or all the sytems in an installation. Any single report, or group of reports, can be generated in an independent documentation pass.

II. WHAT ARE WE AIMING FOR?

Information on existing systems is being collected that, because of volume and desired access to information, will necessitate the use of a data dictionary. In addition, the query power of a data dictionary is desirable for both forward and backward tracking of information.

Figure 1 depicts the decomposition of information and the types of relationships we are interested in documenting and maintaining. The term "System" as it is used in this paper

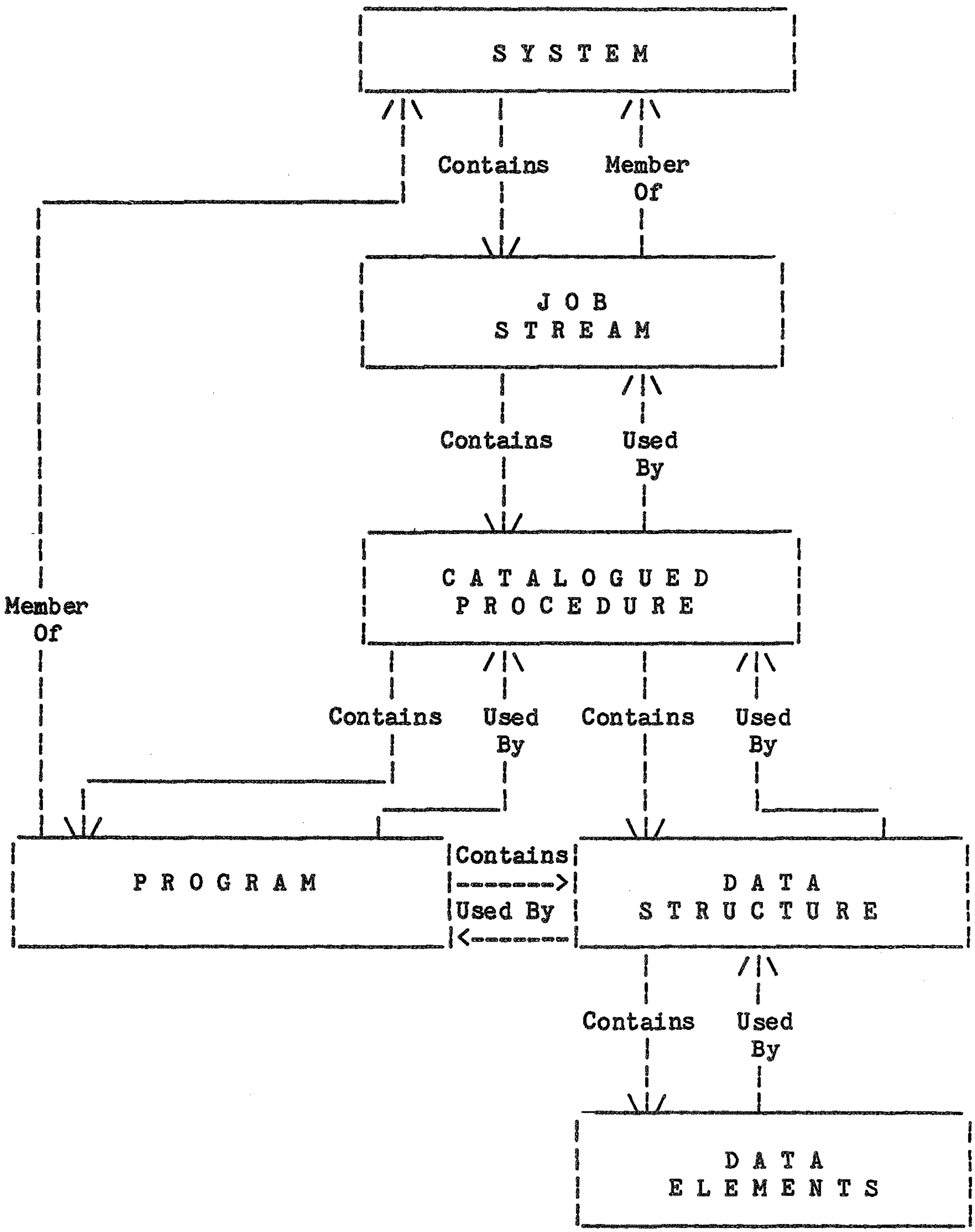


FIGURE 1
INFORMATION DECOMPOSITION

refers to an entity (or entities) that is thought of as a single function. An example of a system is Payroll; or Check Generation, which is a sub-set of Payroll. Each of these are considered systems; systems may be further decomposed into other systems. Systems contain job streams.

A job stream(s) is initiated by operations to control the processing of a particular system(s). A job stream belongs to one top level system. A typical job stream consists of a "Job Card" and one or more calls to catalogued procedures.

Catalogued procedures consist of Job Control Language (JCL) that identifies the programs and data structures to be used in a particular job stream and also the order in which the processing takes place. Catalogued procedures may be used by one or more job streams.

Programs consist of actual source code. Programs (load modules) may be used by one or more catalogued procedures, but they are members of a single top level system. Programs contain one or more data structures.

Data structures are used in catalogued procedures as well as by individual programs. In most instances, the data structure (file) used by the program and the catalogued procedure are the same. However the situation might exist where a program which is run weekly uses one data structure (file) and then the same program is run monthly using a different data structure (file). Data structures may be decomposed into data elements.

III. HOW DOES DOCU-TEXT FIT INTO THE PICTURE?

A user of DOCU-TEXT must identify which "System" he wishes to analyze. He does this by identifying which job streams are to be included in the analysis.

A job stream(s) is the basic input to DOCU-TEXT. It is the standard, unmodified execution card deck(s) or disk file(s) used by operations for regular production. Up to 10,000 separate job streams can be grouped together and documented as a system.

Figure 2 shows the overall DOCU-TEXT system.

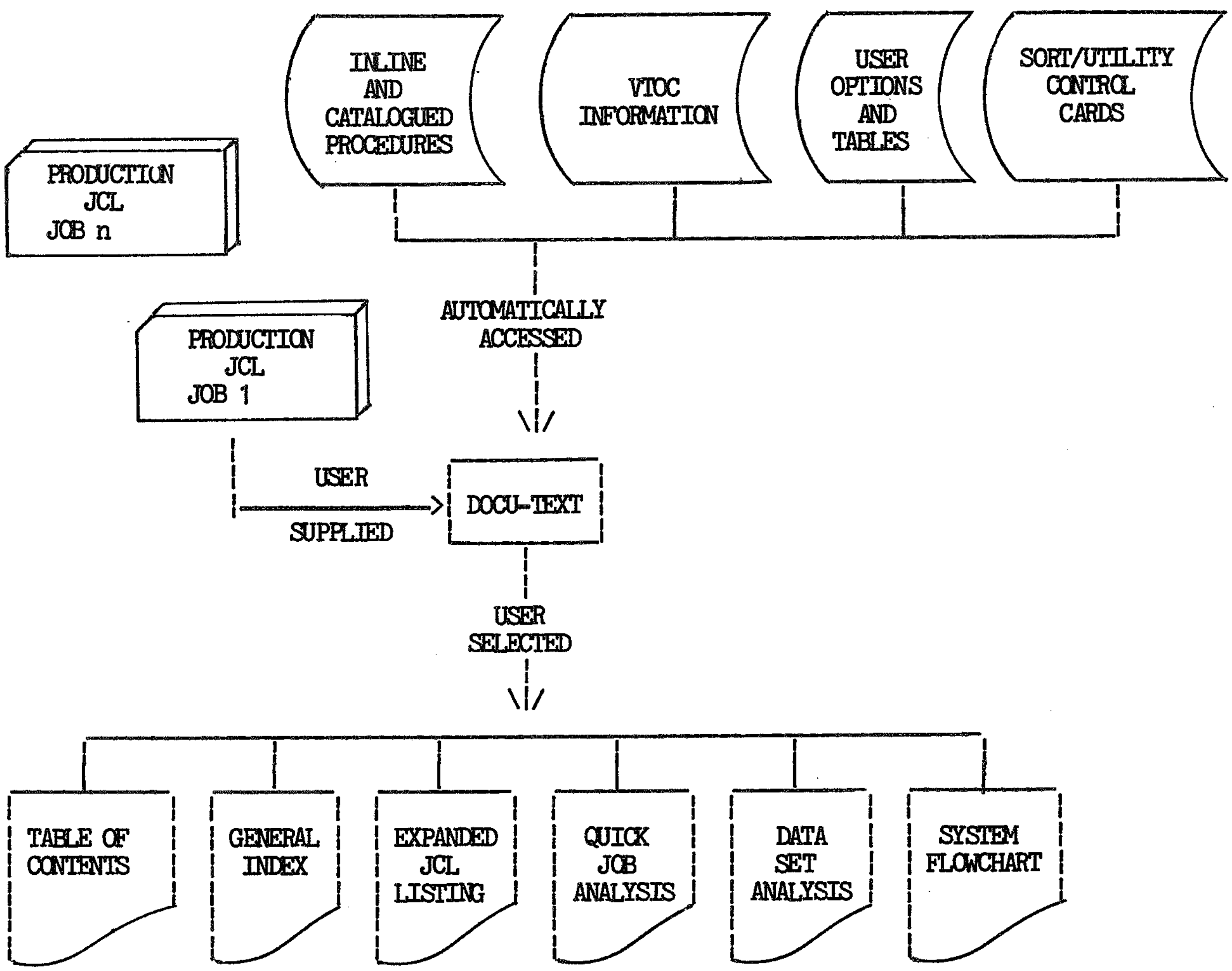


FIGURE 2
THE DOCU-TEXT SYSTEM

DOCU-TEXT is able to automatically generate lists that, prior to the installation of this package, had to be tediously compiled manually. One example of this might be a list of all the programs for the Payroll System; of, on a grand scale, a list of all data sets used in all the Financial Systems.

DOCU-TEXT does not analyze data structures down to the data element level. It also does not analyze individual source code satements for data structures. The data structures referenced by DOCU-TEXT are file names associated with catalogued procedures and not record layouts used by individual programs. It does, however, produce a number of reports, some of which are concise and some which provide greater detail for analyzing job streams, catalogued procedures, programs and data structures.

IV. DOCU-TEXT REPORTS

The following reports are available for use by the data processing organization:

- o Table of Contents
- o General Index
- o Expanded JCL Listing
- o Quick Job Analysis
- o Data Set Analysis
- o Systems Flow Char

TABLE OF CONTENTS REPORT

Figure 3 shows a sample of the Table of Contents report.

TIME 09:28:10

FAIRCHILD CAMERA

*** EXPANDED JCL LISTING ***

JOB	JOB-STEP	PROC	PROCSTEP	PGM	PAGE	JOB	JOB-STEP	PROC	PROCSTEP	PGM	PAGE
J150A002	C150A002	C150A002	STEP01	IEBGENER	1	J150A025	C150A025	C150A025	STEP050	P150B20	8
J150A020	C150A020	C150A020	STEP010	DELETDS	2		C150A025	C150A025	STEP060	P150B07	9
	C150A020	C150A020	STEP020	P150A025	2		C150A025	C150A025	STEP070	SORT	11
J150A025	C150A025	C150A025	STEP010	DELETDS	5		C150A025	C150A025	STEP080	P150C0[illegible]1	11
	C150A025	C150A025	STEP020	P150A025	5		C150A025	C150A025	STEP090	SPLITX	12
	C150A025	C150A025	STEP030	P150B10	7		C150A025	C150A025	STEP100	SPLITX	13
	C150A025	C150A025	STEP040	SORT	8		C150A025	C150A025	STEP110	SPLITX	13

*** QUICK JOB ANALYSIS ***

JOB	PAGE	JOB	PAGE	JOB	PAGE	JOB	PAGE	JOB	PAGE	JOB	PAGE
J150A002	10001	J150A020	10002	J150A025	10003						

*** DATA SET ANALYSIS ***

FIGURE 3
TABLE OF CONTENTS

The Table of Contents report is similar to the Table of Contents for a regular reference manual. Each report is treated as a separate chapter. The line entries contain the entity name and the exact page number where the entity can be found. Besides providing a ready reference for the content of each report, it can present the components of an entire system in less than a page.

GENERAL INDEX REPORT

Figure 4 shows a sample of the General Index report. The General Index report contains a sorted list of every job name, catalogued procedure name, program name and file name in the system being documented. This index can have many uses. For example, if you needed to know whether a particular program or file was used in a system, you could reference the General Index report for that system. Another use might be to verify adherence to naming conventions. Each section is sorted alphabetically and contains the page numbers in each of the reports where the named entity can be found.

EXPANDED JCL LISTING

Figure 5 shows a sample Expanded JCL Listing. The Expanded JCL Listing contains the production JCL, procedure library members and sort/utility control cards. Lines requiring symbolic replacement are listed first in unmodified form followed with the modified form. Satements used as overrides are listed. Lines

*** JOB INDEX ***

NAME	JCL LIST	FLOW CHART	AUTO RUN	QUICK JOB	NAME	JCL LIST	FLOW CHART	AUTO RUN	QUICK JOB	NAME	JCL LIST	FLOW CHART	AUTO RUN	QUICK JOB
J150A002	1	00001		10001	J150A020	2	50002		10002	J150A025	5	50004		10003

*** PROC INDEX ***

NAME	JCL LIST	FLOW CHART	AUTO RUN	QUICK JOB	NAME	JCL LIST	FLOW CHART	AUTO RUN	QUICK JOB	NAME	JCL LIST	FLOW CHART	AUTO RUN	QUICK JOB
C150A002	1	00001		10001	C150A020	2	50002		10002	C150A025	5	50004		10003

*** PGM INDEX ***

NAME	JCL LIST	FLOW CHART	AUTO RUN	QUICK JOB	NAME	JCL LIST	FLOW CHART	AUTO RUN	QUICK JOB	NAME	JCL LIST	FLOW CHART	AUTO RUN	QUICK JOB
DELETDS	2	50002		10002	P150B07	9	50009		10004	SORT	11	50010		10004
DELETDS	5	50004		10003	P150B10	7	50006		10003	SPLITX	12	50012		10004
IEBGENER	1	50001		10001	P150B20	8	50008		10003	SPLITX	13	50013		10004
P150A025	2	50003		10002	P150C011	11	50011		10004	SPLITX	13	50014		10005
P150A025	5	50005		10003	SORT	8	50007		10003					

*** FILE INDEX ***

DATA SET NAME	DEVICE	DSORG	PAGE	DATA SET NAME	DEVICE	DSORG	PAGE
&*UNNAMED FILE 00001	D	PS	30002	FCI1.A00150.F0150.F004(XXXXXXXX)	D	PS	30005
&*UNNAMED FILE 00002	D	PS	30003	FCI1.A00150.F0150.F005(XXXXXXXX)	D		30006
&*UNNAMED FILE 00003	D	PS	30003	FCI1.A00150.F0150.F006(XXXXXXXX)	D		30006
&*UNNAMED FILE 00005	D	PS	30003	FCI1.A00150.F0150.F041(XXXXXXXX)	D		30006
&*UNNAMED FILE 00006	D	PS	30003	FCI1.A00150.F0150.F051	D	PS	30007
&*UNNAMED FILE 00013	D	PS	30004	FCI1.A00150.F0150.F056	D	ISAM	30007
&WORKHORS	D	PS	30004	FCI1.A00150.L0150.L008(XXXXXXXX)	D		30007
CMCTAPE	T	PS	30004	FCI1.A00150.L0150.L040(XXXXXXXX)	D		30008
DATA SET NAME NOT AVALABLE	R	PS	30001	FCI1.A00821.F0821.F000(XXXXXXXX)	D		30008
DUMMY	X	PS	30004	FCI1.A65003.F0051.F100	D	PS	30008
FCI1.A00140.F0140.F071	D	ISAM	30005	FCI1.CTLCARDS(XXXXXXXX)	D	PDS	30009
FCI1.A00150.F0150.F000(XXXXXXXX)	X		30005	FCI2.A00150.F0150.F007	X	PS	30009

FIGURE 4
GENERAL INDEX

```
DOCU/TEXT-200 VERSION 4.39                    *** EXPANDED JCL LISTING ***                          DATE RUN 12/04/80 PAGE      2
COPYRIGHT 1979. ALL RIGHTS RESERVED.                                                                TIME      09:28:10
DIVERSIFIED SOFTWARE SYSTEMS, INC.          FAIRCHILD CAMERA
**=====================================================================================|GENERATED INDEX ***
*** JOB JCL | PROCLIB | SORT/UTILITY CARDS                                              | JOB JOB PRC  DS***
**=====================================================================================|     STP STP    ***


  //J150A020 JOB (00150,B03),PROD,                                                          2   0   0
  // REGION=1028K,                                                                          2   0   0
  // TYPRUN=HOLD,                                                                           2   0   0
  // CLASS=G                                                                                2   0   0
  //*******************************************                                             2   0   0
  //*        PARM CARD VOL SER                                                              2   0   0
  //*******************************************                                             2   0   0
  //*LOGONID A00150                                                                         2   0   0
  //* APOLLO HC                                         INSERTED                              2   0   0

  //C150A020  EXEC C150A020,WTR2=V,                                                         2   1   0
  // PARM.STEP020=(01,518)                                                                  2   1   0
  $$STEP020.CARDIN DD *                                                                     2   1   0
  /*                                                                                        2   1   0
        //C150A020 PROC MSGS=A,                                    00000010                 2   1   0
        //       FCI2=FCI2,                                        00000020                 2   1   0
        //       SYSDAV=SYSDAV,                                    00000030                 2   1   0
        //       WTR2=Y,                                           00000040                 2   1   0
        //       WORK=WORK                                         00000050                 2   1   0
        //*******************************************************  00000060                 2   1   0

        //STEP010  EXEC PGM=DELETDS                                00000070                 2   1   1
        //*******************************************************  00000080                 2   1   1
        //SYSPRINT DD  SYSOUT=&MSGS                                00000090                 2   1   1   1
        --SYSPRINT DD  SYSOUT=A
        //SYSIN    DD  DSN=FCI1.CTLCARDS(U150A024),                00000100                 2   1   1   2
               FCI2.A00150.F0150.F007                                        00010000       2   1   1   2
               FCI2.A00150.F0150.F054                                        00020000       2   1   1   2
               FCI2.A00150.F0150.F055                                        00030000       2   1   1   2
                                                                                            2   1   1   2
        //             DISP=SHR                                    00000110                 2   1   1   2
        //*******************************************************  00000120                 2   1   1   2

        //STEP020  EXEC PGM=P150A025                               00000130                 2   1   2
        //*********************************************************00000140                 2   1   2
        //BATCHDRS DD  DSN=&WORKHDRS,                              00000150                 2   1   2   1
        //             DISP=(,DELETE,DELETE),                      00000160                 2   1   2   1
        //             UNIT=SYSDAV,                                00000170                 2   1   2   1
        //             MSVGP=&WORK,                                00000180                 2   1   2   1
        --             MSVGP=WORK,
        //             SPACE=(13024,(600,150),RLSE,,ROUND),        00000190                 2   1   2   1
        //             DCB=(RECFM=FB,LRECL=148,BLKSIZE=13024)      00000200                 2   1   2   1
        //CARDIN   DD  DUMMY,                                      00000210                 2   1   2   2
        //             DCB=BLKSIZE=80                              00000220                 2   1   2   2
        $/STEP020.CARDIN DD *                                                               2   1   2   2
        //CMCTAPE  DD  DSN=CMCTAPE,                                00000230                 2   1   2   3
        //             DISP=OLD,                                   00000240                 2   1   2   3

JOB (J150A020) CONTINUED.
```

FIGURE 5
EXPANDED JCL LISTING

requiring override replacement are listed first in unmodified form followed by the override statement. Sort/utility contol cards are listed.

QUICK JOB ANALYSIS

Figure 6 shows a sample of the Quick Job Analysis report. The Quick Job Analysis report provides an edited list of each job, catalogued procedure, program and file in a system. The major feature of this report is its concise format. Many steps can appear on a single page and each file requires at most one line.

DATA SET ANALYSIS

Figure 7 shows a sample of the Data Set Analysis report. The Data Set Analysis report shows where a particular file is used. This report is organized alphabetically by file name. Temporary files, such as SORTWK, can be eliminated from this report. Files can specifically be selected.

SYSTEMS FLOW CHARTS

Figure 8 shows a sample of System Flow Chart. The System Flow Chart pictorially shows each step in a system. The characteristics of each file are shown. The step information shown is that in effect when the step is actually executed

```
**==========================================================================================** GENER INDEX
** JOB  | PROC  | PGM   |  DDNAME   DEV DATA SET NAME                        FILE IDENTIFICATION         ** JOB JOB PRC
**==========================================================================================**     STP STP

J150A025                                                                                            3   0   0

     C150A025 C150A025                                                                              3   1   0
       DEFAULT  SYMBOLICS - CTLLIB=FCI1.CTLCARDS  WTR2=Y  MSGS=A  ZERO=0  ONE=+1
       DEFAULT  SYMBOLICS - TWO=+1  THREE=+1  WORK=WORK  FCI1=FCI1  FCI2=FCI2

           DELETDS  STEP010                                  FLOW CHART PAGE  50004                 3   1   1
                    I  SYSIN     D  FCI1.CTLCARDS(U150A025)

           P150A025 STEP020                                  FLOW CHART PAGE  50005                 3   1   2
                    I  BATCHDRS  D  &*UNNAMED FILE 00003
                    I  CMCTAPE   T  CMCTAPE
                    I  CONTROL   D  F150.F99
                    I  DATEFILE  D  FCI1.A00150.F0150.F051
                    I  ERRORS    X  DUMMY
                    I  REFRTWRK  D  &*UNNAMED FILE 00005
                    I  WORKFILE  D  &*UNNAMED FILE 00006
                    I  XMITIN    D  FCI1.A00150.F0150.F004(0)
                    O  FPRINT    R     REPORT
                    O  GOODATA   X  FCI1.A00150.F0150.F000(+1)
                    O  NOPSFILE  D  FCI1.A00821.F0321.F000(+1)
                    O  REPORT    D  FCI2.A00150.F0150.F107
                    O  SHPALERT  D  FCI1.A00150.F0150.F005(+1)
                    O  SHPREJS   R     2-PART
                    O  SYSOUT    R     REPORT
                    O  SYSOUX    R     REPORT
                    O  SYSDBOUT  R     REPORT
                    O  TYPER     D  FCI2.A00150.F0150.F154
                    O  TYPEC     D  FCI2.A00150.F0150.F155

           P150B10  STEP030                                  FLOW CHART PAGE  50006                 3   1   3
                    I  SYS050    X  DUMMY
                    I  XMITIN    D  FCI2.A00150.F0150.F155
                    I  DATEFILE  D  FCI1.A00150.F0150.F051
                    O  SYS020    D  FCI2.A00150.F0150.F102
                    O  SYSOUT    R     REPORT

           SORT     STEP040                                  FLOW CHART PAGE  50007                 3   1   4
                    I  SORTLIB   D  SYS1.SORTLIB
                    I  SORTIN    D  FCI2.A00150.F0150.F102
                    I  SYSIN     D  FCI1.CTLCARDS(U150AS40)
                    O  SORTOUT   D  FCI2.A00150.F0150.F103
                    O  SYSOUT    R     REPORT

           P150B20  STEP050     UPDATE CUSTOMER MASTER ISAM F150  FLOW CHART PAGE  50008            3   1   5
                    I  SYS021    D  FCI2.A00150.F0150.F103
                    I  CUSMA     D  FCI1.A00150.F0150.F056
                    I  DATEFILE  D  FCI1.A00150.F0150.F051
                    O  SYS052    R     REPORT
                    O  SYSOUT    R     REPORT
```

FIGURE 6
QUICK JOB ANALYSIS

```
DOCU/TEXT-200 VERSION 1.65                      ***  DATA SET ANALYSIS  ***                   DATE RUN  12/04/80    PAGE  30007
COPYRIGHT 1979. ALL RIGHTS RESERVED.                                                          TIME      09:28:10
DIVERSIFIED SOFTWARE SYSTEMS, INC.          FAIRCHILD CAMERA

**=================================================================================================** GENER INDEX
** JOB     |JOBSTEP  |PROC      |PROCSTEP |PROGRAM  |PROGRAM ID       |F/C P. |DDNAME   |DISP                | FILE ID       ** JOB JOB PRC
**=================================================================================================**     STP STP

**DISK******************************************>   FCI1.A00150.F0150.F051   <*************************************DISK**
**                                                                                                 **
**  DSORG VOLSER RECFM RETPD CREDT EXPDT PROTECTION                              BLK/LRECL SPACE   **
**  PS    PRODX1 F           76176 00000 UNPROTEC                              00064/00064         **
**-------------------------------------------------------------------------------------------------**

J150A002  C150A002  C150A002  STEP01    IEBGENER                    50001 I SYSUT2   OLD                              1   1   1
J150A020  C150A020  C150A020  STEP020   P150A025                    50003 I DATEFILE SHR                              2   1   2
J150A025  C150A025  C150A025  STEP020   P150A025                    50005 I DATEFILE SHR                              3   1   2
J150A025  C150A025  C150A025  STEP030   P150B10                     50006 I DATEFILE SHR                              3   1   3
J150A025  C150A025  C150A025  STEP050   P150B20   UPDATE CUSTOMER   50008 I DATEFILE SHR                              3   1   5
                                                  MASTER ISAM F150


**DISK******************************************>   FCI1.A00150.F0150.F056   <*************************************DISK**
**                                                                                                 **
**  DSORG VOLSER RECFM RETPD CREDT EXPDT PROTECTION                              BLK/LRECL SPACE   **
**  ISAM  PRODX1 FB          80338 00000 UNPROTEC                              01560/00156         **
**-------------------------------------------------------------------------------------------------**

J150A025  C150A025  C150A025  STEP050   P150B20   UPDATE CUSTOMER   50008 I CUSMA    OLD                              3   1   5
                                                  MASTER ISAM F150
J150A025  C150A025  C150A025  STEP060   P150C011                    50011 I CUSMA    SHR                              3   1   8


**DISK*************************************>   FCI1.A00150.L0150.L008(XXXXXXXX)   <*******************************DISK**
**                                                                                                 **
**  DSORG VOLSER RECFM RETPD CREDT EXPDT PROTECTION                              BLK/LRECL SPACE   **
**        MS0219 F           76185 00000 UNPROTEC                              00000/00000 (6118,(180,20),RLSE,,ROUND)     **
**-------------------------------------------------------------------------------------------------**

GDG MEMBER +1
J150A025  C150A025  C150A025  STEP060   P150B07   CREATE BATCH      50009 O FOLP     NEW,CATLG,DELETE                 3   1   6
                                                  ERROR XMIT FILES
```

FIGURE 7
DATA SET ANALYSIS

```
DOCU/TEXT-200 VERSION 1.65                         ***  SYSTEM FLOW CHARTS  ***                        DATE RUN  12/04/80    PAGE  50006
COPYRIGHT 1979. ALL RIGHTS RESERVED.                                                                   TIME      09:28:10
DIVERSIFIED SOFTWARE SYSTEMS, INC.           FAIRCHILD CAMERA                                          GENERATED INDEX        3   1   3

  JOB -J150A025                                  TIME-4
  PROC-C150A025  C150A025
  PGM -P150B10   STEP030
                         PARM-
                         COND-(0,NE)

DUMMY                        XXXXXXXXXXXXX
        PS          SYS050   XX          XX
   80                        XX  DUMMY   XX********
                             XX          XX       *                                 XXXXXXXXXXX  FCI2.A00150.F0150.F102    FB
                             XXXXXXXXXXXXX        *                                (X         X(   SYSDAV  PS              SYS020
                                                  *                       ******->(X  DISK  X(      6160/80     NEW,CATLG,DELETE
                                                  *                       *N-50007 (X         X(   (6160,(225,40),RLSE,,RO
FCI2.A00150.F0150.F155   FB    XXXXXXXXXXX        *       XXXXXXXXXXXXXX       *         XXXXXXXXXXX
SYSDAV  PS          XMITIN    (X         X( P-50005*      X            X       *
12960/80    OLD,DELETE,KEEP   (X  DISK X(   *************->X  P150B10   X********
(12960,(60,10),RLSE,,RO        (X        X(       *       X            X       *       XXXXXXXXXXXXX      REPORT              FBA
                               XXXXXXXXXXX        *       XXXXXXXXXXXXXX       *       XX          XX        PS              SYSOUT
                                                  *                       ******->XX REPORT  XX    133/133    COPIES=05
                                                  *       J-REGION = 2048              XX      _--XX/
FCI1.A00150.F0150.F051    F    XXXXXXXXXXX        *       SYSUDUMP = A                 *XX___/
DISK    PS          DATEFILE  (X         X( P-50005*
00064/00064 SHR               (X  DISK X(   ********
                       PRODX1  (X        X( N-50008
                               XXXXXXXXXXX
```

FIGURE 8
SYSTEM FLOW CHARTS

V. SUMMARY

DOCU-TEXT is a useful tool in organizing and supplying certain system information that is necessary to establish a data dictionary. If a tool such as this were not available, many hours would be lost in manually trying to capture the information.

DBMS UTILIZATION: A Corporate Information System (CIS) Development Approach

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

Pete Rozett
The MITRE Corporation

ACKNOWLEDGEMENT

The author wishes to acknowledge that this document contains portions of documents written by other members of the Corporate Information System (CIS) project team since its inception.

CONTENTS

SECTION 2

WHAT IS CIS?

The CIS is an integrated information system intended to tie the corporation together as a functioning entity. In addition to being a major upgraded automated data processing system, the CIS is a management philosophy which recognizes data as a valuable corporate resource and which distinguishes between data and selected data, or information. It further recognizes that different users need different kinds of information. As the CIS evolves, it will offer its users not just after-the-fact data, but timely information in a format that is meaningful and useful to the particular user, so that the information can be applied in planning, controlling, and decision making by all levels of management. In effect, CIS will help the Corporation itself to function as a total, integrated system by typing together administrative activities through information exchange.

The CIS supports the operational, tactical control, and strategic planning functions of the Corporation. Operational functions are the day-to-day processing necessary to support the Corporation's work, such as Purchasing, Payroll, etc. Tactical control involves the effective deployment and utilization of resources against tasks, as performed by project leaders and department heads in budget management and staff planning. Strategic planning implies the selection of objectives and the necessary policies and resources. This executive decision-making could, for example, include work program selection or other hypothesis testing.

Achieving the goals of the CIS requires more than reprogramming of the automated portion of the system. It includes:

- improving discipline and procedures for manual processes;
- improving coordination among the various parties involved;
- educating the users of the system;
- creating a mechanism to communicate new requirements; and resolve operational problems;
- improving communication between corporate headquarters and remote operations and sites;
- providing timely and direct access to relevant information;
- reducing development lead time for new and changing requirements;
- improving the quality of reports;
- reducing the quantity of reports;
- providing an adequate level of responsibility reporting.

Effective responsibility reporting recognizes that the level of detail of the information required varies with the level of the manager. For example, the project leader needs detailed expenditure data for the day-to-day operations of his project. Senior management, on the other hand, need summary and trend data to coordinate their broader area of responsibility.

SECTION 3

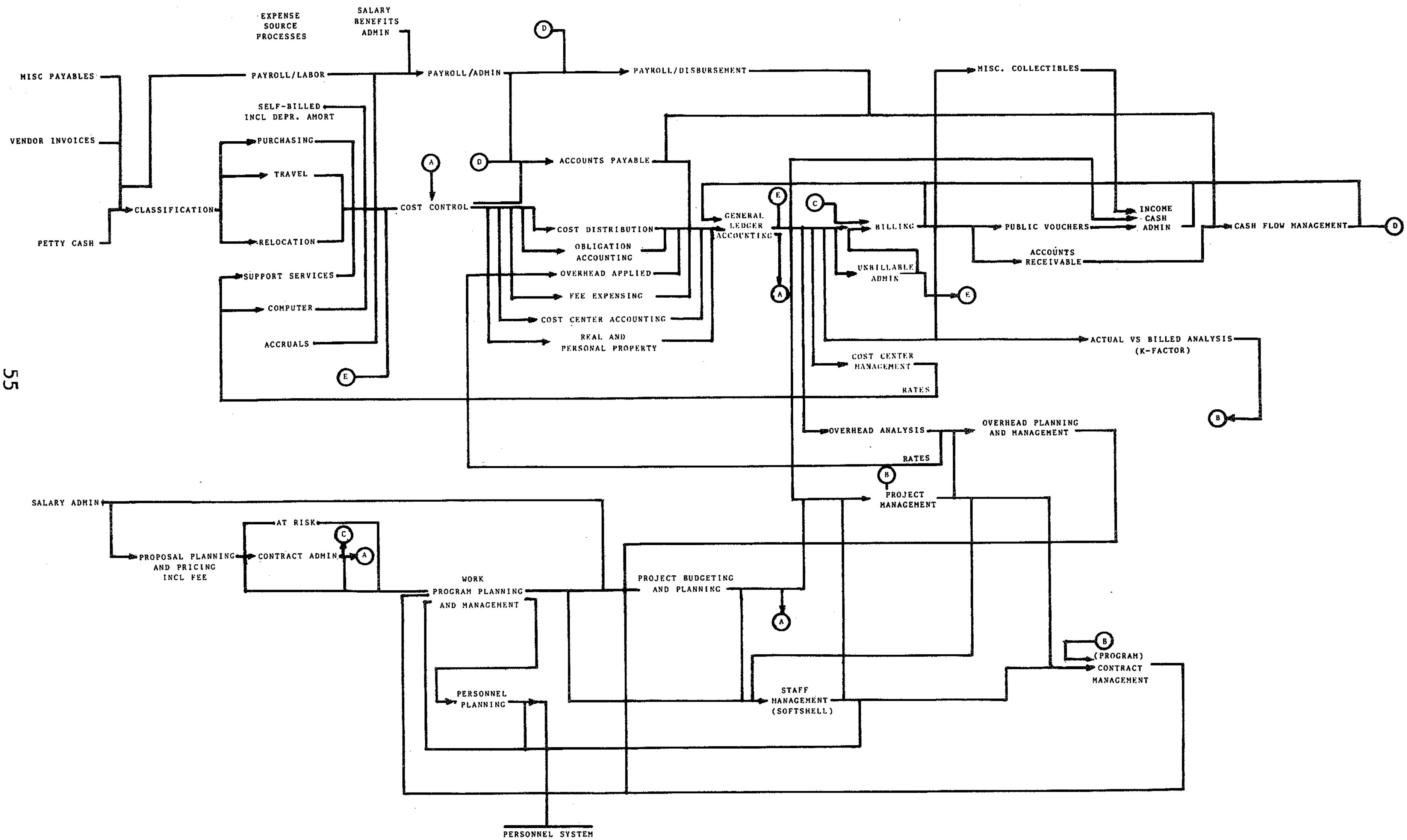
HOW BIG A PROJECT?

Building an integrated information system across functional boundaries is a mammoth undertaking. Figure 1 outlines the functional scope of the project. Within this scope, 28 functional subsystems have been defined. Development is scheduled to last five years.

Seven million (1981) dollars have been budgeted for development of the CIS. This budget is within the accepted three to four percent of gross sales most commercial operations spend on data processing support. Approximately 25 staff personnel will be involved.

Figure 1

CIS—FUNCTIONAL SCOPE



SECTION 4

WHY IS IT BEING DEVELOPED?

CIS PREDECESSOR

The predecessor of our CIS, the Budget and Accounting (B&A) System, evolved over more than twenty years. Automated support was provided to various accounting activities as the computer resource migrated from NCR posting machines to our current IBM 3031 central processor with associated peripherals. A major effort to integrate these automated accounting activities across functional boundaries was undertaken about twelve years ago. However, that effort focused on detailed accounting operations and line management on the premise that collecting clean, basic data and maintaining its integrity was of uppermost importance. Upper level management and client reporting requirements were not fully recognized at the time.

The current generation of applications which comprise the B&A System may be characterized as follows:

- its processing mode is mostly batch;
- its software is application-specific and developed in response to a particular request;
- files are primarily specific to the applications but, when possible, are shared or referenced;
- the members of its data base are usually sequential;
- processing occurs at discrete intervals, as does the reporting;
- special customized reports are produced upon request.

CHANGING REQUIREMENTS

Since the original system design, three major changes have taken place. First, changes in the characteristics of the DoD work have resulted in new reporting requirements. Second, corporate growth has included significant support to clients outside the DoD community, tending towards smaller contracts of shorter duration which impose different controls and management responsibilities. Finally, there has been a substantial growth in our Washington Center operations, resulting in the geographic separation of a significant portion of corporate activities from corporate offices in Bedford.

PROBLEM IDENTIFICATION

These changes put a strain in the information system. In response to pleas from project leaders and managers for more relevant information on a timely basis, the Corporate Treasurer chartered the Corporate Information System Requirements and Design (R & D) Team to examine the information system in order to:

- o Define existing problems;
- o Establish requirements;
- o Prepare a preliminary system redesign specification, assuming one was needed.

To fulfill its charter, the team conducted extensive interviews with all levels of financial and administrative management, with operating division senior management, and with representative groups of technical division management. The team studied the work of previous committees and MITRE's Information Services Group, and, in addition, investigated commercially available business information systems.

The team's findings were published in an internal document intended for top-level management. Their description of deficiencies included policies and procedures as well as automated support. In addition to new requirements, the team identified fundamental problems, growth related problems and management support problems permeating the system structure.

As an adjunct to their efforts, the team initiated a study of software alternatives conducted by MITRE's Information Services Group (ISG). Their study indicated that the automated portion of the B&A System was based on outdated file access techniques due to the unavailability of effective file management software at the time of its design and development. Using conventional file software had bred several problems:

- o embedded record layouts in the application programs;
- o shared files;
- o scattered data;
- o redundant data.

Programming resources were devoted primarily to the maintenance of current applications and, as a result, were not always available to implement applications requiring substantial design and programming effort. Some areas had either no support or inadequate support. History files were either unavailable or too cumbersome to be useful, making historical profiles and trend analyses difficult to compile. Ad hoc queries and new reporting requirements were sometimes difficult to implement because of unintegrated tables and files and uncoordinated use of data element values.

An enhancement to one application program which created a change in any file record format or an expansion in fields, could force modification to all programs which accesed the file, whether or not they used the affected fields. The team stated bluntly that the B&A System was fragile and inflexible, difficult and cumbersome to modify, and caused delays in satisfying special information requests.

RECOMMENDATIONS

Taking into account MITRE's dynamic environment, the ISG team stated that any improvement to the automated portion of the system should strive towards an independance of data and application programs. It should improve the ability to meet special information requests, reduce new development lead time and integrate databases. To accomplish these goals, they strongly recommended the utilization of a state of the art Data Base Management System (DBMS).

The CIS Requirements and Design Team recommended replacing the B&A System with a more flexible, user-oriented Corporate Information System built around a commercial DBMS. A corporate decision was made to endorse the CIS project.

The R&D team then identified and prepared information requirements for all levels of management through a series of formatted report mockups which served as a coordination and communication media. In this form, the requirements were also used to identify data elements needed in computer files and by manual procedures to complete successful system design, implementation and use.

SECTION 5

HOW IS IT BEING DEVELOPED?

EVOLUTIONARY APPROACH

An evolutionary approach has been chosen to build the new system since this effort will extend over several years. The implementation of any portion of the CIS cannot negatively affect the level of financial processing support that users currently receive. Components of the existing Budget and Accounting system may be modified to support or interface to new subsystems but must continue to operate as usual until phased out by new CIS components. A transitional approach for the implementation of each CIS subsystem is defined which includes parallel operation of the new subsystem with the old manual or automated process. User demonstrations and training for the new system are provided before parallel operation or user test. Implementing one CIS subsystem at a time allows end users to verify the new capability and adjust to new procedures.

The evolutionary approach to CIS design and implementation also allows improved application development productivity. Small, readily-implemented subsystems have been defined to assure tighter control for CIS project management efforts and to improve individual task productivity. CIS development personnel are able to apply experience gained on early applications to later, more demanding subsystems. With the use of a phased development process, concurrent development activities allow more effective utilization of the development resources. Staff requirements during each subsystem's life cycle vary but can be overlapped so that tasks are ready for people at the same time people are ready to perform the tasks.

CIS PROJECT STRUCTURE

The overall CIS project structure is shown in Exhibit I. The CIS project reports to corporate management, providing corporate-level involvement and control. There are Bedford and Washington Center components to each subproject.

The System Engineering (SE) role is to assure that CIS development continues in accordance with corporate objectives. System Engineering is responsible for continuing definition of user requirements and their prioritization, for coordination of change control procedures, and for review of functional requirements and specifications. System Engineering acts as the primary user interface and coordinates user training, documentation, and acceptance testing.

EXHIBIT 1

CIS PROJECT GROUP RESPONSIBILITIES

SYSTEM ENGINEERING
- Primary User Contact
- Requirements Identification and Prioritization
- Configuration Control

DATA SYSTEMS INTEGRATION
- Data Analysis and Data Base Design
- Data Administration
- Data Base Software

DESIGN
- Functional Requirements Analysis
- Functional Subsystem Design

DEVELOPMENT
- Technical Subsystem Design
- Application Construction and Integration

MAINTENANCE
- Subsystem Test
- Production Support
- Configuration Detail

Data Systems Integration (DSI) accomplishes most of the tasks associated with data base administration and is responsible for designing and implementing the CIS data base. This includes determining data requirements with Design, selecting a subsystem data conversion strategy with Development, performing the physical data base design and initial data load for the subsystem, and managing and administering the data on a day-to-day basis. Data Systems Integration is also responsible for data base software support.

The Design effort includes responsibility for maintaining the overall CIS design framework and for converting requirements into detailed subsystem designs. Design is responsible for evaluating the impact of new user requirements on the CIS as well as on existing development schedules and resources. For each subsystem, Design defines the functional requirements and functional design by analyzing and redefining as necessary existing manual and automated support.

Development is responsible for the detailed technical design and construction of subsystems, for definition of test criteria for system testing, and for transition to operational use. Development also provides technical support for user training and procedures.

Maintenance is responsible for the production support for CIS applications. As part of the subsystem development process, Maintenance personnel perform subsystem testing and support user testing and parallel operations. Upon satisfactory completion of these steps and delivery of documentation, Maintenance accepts the subsystem into production and coordinates transfer of processing responsibility to the Computer Center Operations Staff as necessary.

PROJECT CONTROLS

In order to keep the project under control and to be able to track progress, management controls have been built into the CIS development approach.

Change Management

To most effectively and efficiently develop and implement the CIS within planned schedules, all available development resources must be utilized to the fullest capacity. To accomplish this, change control standards and procedures have been defined which monitor and evaluate user requests for system change. Users requesting changes to the CIS must quantify and document the benefit of the change. Procedures have been established to review each user request to determine if that request is:

- A new CIS development effort which will require rescheduling and reprioritization of the development schedule;

- A modification/enhancement to a production mode CIS subsystem;

- A modification/enhancement to a current application which would impact planned CIS subsystems; or

- A non-CIS related application.

An impact statement is prepared by System Engineering and Maintenance in order to process the request and formally respond to the user. If a commitment is made to satisfy the request it is scheduled through the development process or implemented by Maintenance, depending on the nature of the request.

Milestone Tracking

To ensure the development and implementation of all subsystems on time and within budget, the CIS project leader requires information in a consolidated fashion in order to monitor the progress of each individual subsystem as well as the progress of the CIS in total. To support this requirement, a milestone schedule has been prepared which:

- Identifies each subsystem;

- Specifies planned milestone dates for each subsystem;

- Specifies personnel assignments and responsibilities.

Major milestone activity dates are completed prior to the start of each subsystem by the individuals who have been assigned responsibility for each area of Design, Development, and Data Systems Integration. From the start of the subsystem, the individual with responsibility for a given milestone must issue any revisions to previously submitted dates on a biweekly basis. Similarly, as each milestone is met, the responsible individual must supply the actual date to the project leader.

A milestone chart is maintained for every CIS subsystem scheduled for development. These charts allow the project leader to maintain a current knowledge of the status of all CIS subsystems which have been scheduled and to recognize and take action on potential problems.

Reviews and Approvals

Several levels of management reviews and checkpoints, as well as direction, participation, and decision-making by the users is provided. The user interacts with the development process at established milestones to review requirements and specifications during the definition and design phases of subsystem development. Close user involvement is necessary as subsystem design may require modifications to current operating policies, practices and procedures, as well as certain managerial responsibilities before implementation. An active user role is necessary to define and document new procedures and practices.

CIS project personnel review each subsystem during each phase of its development. Document reviews in combination with subsystem review meetings are employed to coordinate comments and discussion items from various project groups. Approval by the cognizant CIS project representative indicates that all problems and conflicts have been resolved.

General project reviews are performed at periodic intervals by CIS management. A formal 12-month schedule is maintained which shows scheduled elapsed time for major design, data base, and development efforts by subsystem. Progress against the plan is maintained and summarized at intervals for corporate and division managers.

DEVELOPMENT CYCLE

For development scheduling purposes, the CIS is defined in terms of implementable subsystems. Implementable subsystems are relatively uniform in size to allow each subsystem to be performed within a manageable time span, such as six months from functional specification to delivery.

The general approach to defining an implementable subsystem starts with a natural functional area or a cut across more than one function. The scope is then refined by consideration of the interfaces with the current operational system and the interdependencies between the newly defined subsystems with regard to data, software, and user procedures. Whenever it becomes apparent that the development process for a subsystem will require significantly longer than six months, the subsystem is redefined into phases such that products can be delivered at the end of each phase.

The CIS is being developed through a traditional process which takes each subsystem from initial requirements analysis through production status. For each phase there are well-defined objectives, major tasks, products, approvals, and completion criteria. Each stage has review points and provides a plan for the next step. Controls and standard reviews are built into the process to provide internal and external verification of the subsystem design and development. The process itself is monitored and adjusted through experience in order to increase future effectiveness.

Implementable subsystems are defined based on the current definition of overall system requirements. Each is developed according to its own established schedule, and managed by means of a standardized phased approach.

Task Initialization

The first stage in the definition of a new subsystem is the initial requirements determination. System Engineering, based on input from users, is responsible for defining and documenting the CIS user information needs. System Engineering:

- Defines and limits the objectives and scope of the requirements.

- Specifies which user area functions are to be included or excluded.

- Describes overall information requirements.

- Performs an initial cost/benefit analysis based on estimated impact on user and development resources.

- Prioritizes this task and, if necessary, reschedules the implementation of subsystems, based on user needs or development requirements.

- Defines certification criteria for this subsystem.

The task initialization document formalizes the information requirements and communicates between the user and development areas the objectives and scope of the subsystem. The scope of the subsystem is established by determining what functions of the user area are to be included in (and excluded from) the development effort. Overall information requirements are described for those functions contained within the boundaries of the subsystem.

Design and Development personnel support the creation of the Task Initialization document by evaluating the requirements in terms of necessary resources (personnel, data, hardware, and software configuration, etc.). System Engineering and Design define a tentative schedule based on prioritization and resource impact. Upon completion of the Task Initialization document and its approval by the users and the CIS team, the new task is included into the development schedule.

Functional Requirements Analysis

The functional requirements definition stage is scheduled following the acceptance of the Task Initialization document. Design personnel detail what the user requires based on an analysis of the subsystem. The subsystem designer:

- Defines the functions to be addressed by the subsystem in user terminology, including relationships between functions and interfaces with other user areas.

- Defines detailed information requirements, including volume and frequency, for each function.

- Analyzes and states performance objectives, including accuracy, validation, timing and flexibility, as well as audit and control requirements.

- Provides a more detailed cost/benefit analysis, as well as milestone dates and timing requirements.

The functional requirements document defines in the user's terminology the business functions to be addressed by the subsystem. Detailed information requirements are set forth for each business function. The relationships between functions within the user area and their interfaces with other user areas are also defined. Key milestone dates, critical dates and necessary implementation requirements such as user training, manual/form preparation, subsystem conversion procedures and user testing and acceptance procedures are also included in this document.

Functional Design Specification

The functional design specification activity follows the approval of the Functional Requirements. This activity is the initial step in the design of a subsystem. The functional design specifications defines how the subsystem will meet the user requirements. Design personnel:

- Define each system input and output, including its purpose and usage, source, major data elements, proposed format and special considerations, such as privacy/security requirements.

- Define data requirements for both the manual and automated portions of the subsystem, and detail the subsystem data analysis performed by Data Systems Integration. Data analysis will define logical data groupings and relationships based on data requirements and functional specifications, and ownership (control and updating) for each data element. This activity will form the basis for the subsequent data base design.

The functional design specification document describes the subsystem as it will appear to the user: the external design of the subsystem. It demonstrates how the subsystem will meet the requirements of the user area by delineating functional information flows throughout the subsystem and interfaces with other subsystems. Detailed transaction and report formats are described as part of the information flow. A copy of each input form and output report is included in the document. Data requirements, manual and automated, are defined in this document also. Each data input and output to the subsystem is defined by describing its purpose, usage, source, and characteristics.

Review and approval of the Functional Design Specifications document is required by the user as well as by CIS project personnel. In addition, the Corporation's Internal Auditor reviews the specifications for auditability.

Technical Subsystem Design

The technical subsystem design stage follows approval of the functional design specification document. This activity defines the internal design of the automated portion of the subsystem. The Development group:

- o Defines, in technical terms, the subsystem objectives, flow, and specifications for each automated component.
- o Specifies processing functions, inputs and outputs, controls, and performance characteristics.
- o Specifies subsystem building blocks: compiler languages, query/report language components, other support required.
- o Defines the logical data base structure for this subsystem. Logical and physical data base design is performed by Data Systems Integration during this phase and is incorporated into the global data model.
- o Defines the data migration approach or bridging technique developed with Data Systems Integration to accomplish conversion of old system conventional file data to the data base, and how new data will be assimilated into the system.

The technical subsystem design document defines the subsystem's internal design to technical personnel in terms of independent subsystem components and interactions between components. At this point, each component of the subsystem is decomposed into groups of computer program modules which will achieve the function(s) of the component. Data requirements are stated here as the automated inputs and outputs that are necessary for each program. Included in this document is a data conversion plan that describes how the subsystem will extract data from existing subsystems and maintain that data in parallel with existing subsystems during the development process.

The review and approval of the Technical Subsystem Design document concludes the design phase of the development process and allows actual construction to begin.

Program Specification

After approval of the Technical Subsystem Design, the construction phase of the development process begins with the creation of Program Specification documents. The purpose is to document the program design step prior to construction. The processing logic requirements of the program together with the description of the partitioning of the program into modules is provided for later use by Development and Maintenance personnel. A code inspection procedure has been defined whereby adherence to the program specifications can be validated and modules inspected for possible errors as well as violation of programming standards.

The program specifications document identifies each module of each program to be developed within the subsystem. Each program is described in terms of its processing requirements and its relationships to other programs. All data inputs and outputs for each program are defined in this document.

Coding and Integration

This stage continues the construction phase and consists of the development and integration of automated subsystem components into a final product ready for subsystem test. A Development Team consisting of approximately three to six members, including the team leader:

- Builds the subsystem and performs component testing.
- Implements any necessary data migration mechanisms through modifications to old system components.

The Development Team is responsible for adherence to development standards and guidelines. Enforcement of standards occurs during the Technical Subsystem Design and Program Specification reviews, the Code Inspection, and the review by Maintenance and System Engineering personnel during Subsystem and User Test.

User Procedures and Manuals

System Engineering and Design are responsible for the development and publication of user procedures in coordination with the development of the various subsystems. Development supports the documentation of the automated portions of the subsystems. Users participate with CIS personnel in the preparation of procedural manuals. CIS personnel:

- Review user requirements for CIS procedural manuals.
- Review proposed user procedural manuals with proposed training tools to ensure consistency.

- Obtain user approval of proposed manual structure and content (by broad topics). Indicate planned milestone dates for the development and production of user procedural manuals. Issue draft of documentation for review and approval.

- Publish manuals.

- Develop and publish additions/revisions/deletions on an on-going basis.

The User Procedures document serves as a policy and procedure manual. It provides an overview of subsystem functions and a detailed reference to subsystem procedures. Interactive procedures are described for subsystems having "on-line" components.

User Training

Training requirements are defined for each CIS subsystem. System Engineering and Design are responsible for the development of training tools and training seminars. Development supplies automated and technical support where possible. Training tools and programs are developed under the following guidelines:

- Training tools for one CIS subsystem are developed in consideration of other related CIS subsystems. Where particular subsystems share similar training needs, a training package may be designed to satisfy more than one subsystem's training requirements.

- Training tools are designed for the benefit of new personnel as well as for personnel involved in conversion from current procedures to new CIS procedures.

- Various approaches to training techniques are investigated. Alternative approaches include written and pictorial guides/manuals, terminal-based automated training sessions, or audio/visual programs.

- Training sessions are designed in conjunction with the training tools being produced.

Subsystem Test

A controlled test is performed by Maintenance on the automated portion of the subsystem and compared to either manual procedures or previously automated systems. Test procedures to be followed for all newly developed subsystems are:

- A test file and sample transactions are developed; all error conditions included in transaction data and control reports are reconciled.

- Control procedures are implemented to include auditable controls over input for volume and amount fields maintained by the user, an audit trail of input processing actions maintained by the program, and reconciliation of manual or automated controls for each run.

- Reports must reflect disposition of all transactions. For accounting processes control totals are reported.

- Test results are documented and verified by Internal Audit.

Maintenance and Computer Center Operations acceptance of the subsystem depends upon delivery of production documentation, processing manuals, and user manuals together with placing programs into a production library and turning over programs to Maintenance, and job streams, tapes, etc., to Operations.

The Subsystem Test Plan document defines the subsystem functions to be tested and describes the tests to be performed. Milestones and schedules for testing are detailed in this document along with the resource requirements needed to accomplish testing. The impact on personnel, software, and equipment resources is addressed here. A demonstration plan is included in this document as a means by which the subsystem can be proven acceptable for maintenance purposes.

Post-implementation Support

The post-implementation phase encompasses review and evaluation of the effectiveness of the subsystem, how closely the subsystem has met the original design specifications, and the definition of any enhancements or modifications which should be made to make the system perform more efficiently. Maintenance to CIS subsystems is applied only after authorization, and is performed in a manner consistent with the development of the subsystem, i.e., programming standards are observed.

SECTION 6

WHAT IS THE DATABASE ENVIRONMENT?

SOFTWARE TOOLS

A database environment is created when your data is easy to relate and manipulate; when data relationships are easy to establish and recognize; when access to data is convenient. To facilitate the development of such an environment, certain software tools have been acquired. These tools include a data base management system (DBMS), a data dictionary, a data communications (teleprocessing) monitor, and a very-high-level language which includes screen formatting/report writing capabilities.

Data Base Management System

The heart of our database environment is provided by our DBMS. Due to its simplicity of design, solid performance, and ease of use, MITRE has procured the ADABAS data base management system from Software AG. ADABAS utilizes an inverted list structure to provide access to the data based on element value. There are no paths or pointers to follow thus providing flexibility in establishing data relationships.

The data and the access lists are stored separately. This approach allows for analysis of data values without ever accessing the data. Histograms of values and number of occurrences are easy to perform and facilitate analysis and data cleaning.

ADABAS can handle a variety of processing modes (sequential access, keyed access, etc.). It can handle comprehensive databases. It's easy to enlarge or modify existing files or to add files. changes to one file do not effect any other file in the database. Very little database reorganization is required.

ADABAS is particularly effective in supporting an environment of dynamic growth and will accommodate a gradual phase-in of changing corporate data with its flexible file structure. The ADABAS package also includes the NATURAL programming and query language, backup and recovery software, security facilities, a data dictionary, interfaces to several transaction processors, and a multi-use capability.

Data Dictionary

A data dictionary is a repository of information which describes an organization's systems, procedures, programs, data, and interrelationships. A comprehensive dictionary serves as a source of documentation for user procedures as well as for automated systems, and assists in data base design and change control. CIS implementation utilizes the ADABAS data dictionary which is integrated with and dependent upon ADABAS.

The ADABAS data dictionary provides good support for DBMS functions such as data definition, loading, descriptive query names, and userview generation, but has minimal support for describing and interrelating non-ADABAS data such as descriptions of user organizations, procedures, terms, manual reports, and necessary changes and extensions to the current data dictionary capabilities to improve its ability to define, locate, access, validate, and control the data resource.

NATURAL Programming/Query Language

NATURAL is a high-level programming language and an integral part of the overall ADABAS DBMS. It has extensive screen formatting and report writing facilities and provides easy access to the database. An entire screen can be formatted with a single coded statement. NATURAL handles all access calls to the database automatically without direct programmer intervention.

NATURAL is actually a combination programming and query language. Information can be displayed at a terminal with a single command or complex Boolean searches and iterative processing loops can be performed. Syntax rules and error messages, however, are confusing to novice users. Therefore, only preformatted queries have been implemented in the CIS to date.

Teleprocessing Monitor

The multi-user feature of ADABAS handles data base access between competing users but does not solve the problem of transaction file access by multiple terminals for widespread data entry. After investigating several teleprocessing monitors, we are currently running bench-mark tests on COMPLETE, a TP monitor, marketed by Software AG, designed to run efficiently with NATURAL.

DATA BASE ADMINISTRATION

Data Systems Integration has primary responsibility for maintaining the physical data base and supporting all activities related to the CIS data base. Activities are underway to define and implement tools for data dictionary maintenance, disk space management, and ADABAS performance monitoring, as well as data base support procedures and capabilities.

Database Support Tools

On-line and batch processing facilities are used to maintain the data dictionary. Future enhancements include upgrading the dictionary's reporting features to reflect MITRE-defined information. The internal dictionary format has been expanded to satisfy MITRE conventional-file information requirements.

Software which provides information on data base disk space usage by file has been developed. A file mapping program provides a concise summary of the position and size of ADABAS files and available free space. This type of information facilitates the scheduling of data base maintenance.

Software has been developed which provides summary statistics of the ADABAS log report to aid in monitoring data base performance. The log report contains a record of each processed ADABAS command; the summary statistics program condenses this information into a convenient, manageable format for analysis.

Data Inventory

An extension of the data dictionary concept is that of a Data Resource Catalog. Such a catalog contains descriptions of the data resources of the corporation and provides processing tools to update and report on these descriptions. Since MITRE already has a considerable amount of automated data and computer programs, a catalog of this information is essential for the identification of important interrelationships which will affect the development of the CIS data base and associated subsystems.

The task of creating and maintaining this catalog is known as the Data Inventory. It involves expanding the data dictionary to encompass job streams, transactions, program modules, source library members, etc. Five major sources of information relating to the current system are being processed for input into the catalog: the actual control language used in production, the source libraries, the production modules, a report and frequency file, and a manual element directory. Inconsistencies are being resolved and report programs are currently being generated to aid in data base design.

Restart/Recovery

To ensure data base recoverability, restart/recovery requirements are identified and planned for during subsystem design. This planning should ensure that the major causes of failure are anticipated, evaluated and resolved as part of the basic system design process.

Restart and recovery procedures fashioned after those recommended by Software AG for the data base have been implemented. To provide general recovery, the data base is saved periodically (currently weekly).

Data Privacy/Security and Access Controls

Having corporation-sensitive data on-line in a database requires an upgrade of procedures to prevent unauthorized access and/or update of this data. Several levels of access controls have been implemented. The maximum data protection facilities of the operating system are used in conjunction with the data base protection available in ADABAS. Use of all data on the data base is controlled via passwords. Any user, whether through an on-line query language or an application program, is required to identify himself with a valid password. Issuing and modifying passwords is currently the responsibility of Data Systems Integration. Each password allows access or update to only a selected portion of the data base.

The privacy/security mechanisms for any data on the data base is defined by Design and Data Systems Integration based on subsystem requirements originally defined by System Engineering. Minimum protection is maintained at the file level; field protection and record protection (by value) is implemented when necessary. Because of the processing overhead involved in these higher level protection measures, justification is rigorous. For extremely sensitive files, encryption of data can be accommodated if the performance trade-off is acceptable.

ADABAS automatically checks the access and update authority of a user's password. Any unauthorized attempt to open a file is noted on the ADABAS log and the user is locked-out of the data base. All violations are closely monitored by Data Systems Integration to ensure that a valid level of data protection is maintained. Should data compromise be suspected, CIS management and System Engineering are notified.

Password mechanisms have been developed which dynamically identify users and provide valid passwords at execution time. Such procedures allow passwords to be modified at frequent intervals with minimal impact on the programming staff and data base users. The frequency of password modification is at the discretion of Data Systems Integration.

Data Base Testing

Independent test and production data bases are provided to support concurrent development testing and production operation without conflicts or risk of temporary loss of data base availability. The test data base is comprised of a scaled-down version of the production data base together with any new data not yet in production status. The target size of the test data base is 10% of the production data base size.

The Development group plans and coordinates subsystem testing with Data Systems Integration. The scope of each subsystem determines the pre-testing preparation, which may include unloading a single file, saving an entire data base, or implementing test logging facilities.

DATA BASE INTEGRITY

The CIS data base is a corporate resource shared by many areas of the Corporation and administered centrally by Data Systems Integration. Standards have been established to control the definition of the data and its documentation, the view of the data base that each particular user accesses, and the manner of that access.

Data Element Definition and Values

To achieve consistency in naming conventions, each unique data element has a unique name and definition. If the same data element occurs in many logical files, it will always have the same name and be referred to in host language programs by that single name. Synonyms are used only externally to aid users. To ensure that all application programs use standard names, the structures into which data is read from the data base is stored by Data Systems Integration in a central library.

Userviews and Data Access

Access by application programs to data in the CIS data base is controlled through the use of "userviews". A userview is a powerful concept central to the data base approach which provides a logical view of the data, based on program requirements alone. ADABAS allows specified data elements to be extracted from one or more physical files, transparent to the application program. Monitoring and controlling access is necessary to improve data file independence, ease future subsystem maintenance and enhancements, and provide an inventory of data element access/update by program.

ADABAS provides userviews via the data dictionary and its conventional language macro interface (ADAMINT) data access modules. Standard userviews are created for ad hoc applications and queries, and simplify the access to data by users since knowledge of the physical structure of the data base is not needed. ADAMINT modules also provide a mechanism by which a user is given direct access to a subset of a data file even when access to the entire file is not authorized

Data Base Checkpoints

The ADABAS protection log is required during all production ADABAS sessions. This log contains the before and after images of all updated information in the data base. Application programs, by means of checkpointing, establish reference points in the log. If a hardware or software failure occurs during data base updating, Data Systems Integration is able to restore the data base to any previous checkpoint.

All batch update programs are required to be restartable from checkpoints. It is the responsibility of Development to ensure that information necessary to achieve this restart (internal subtotals, positioning in the input file, ability to overwrite portions of any output file, etc.) and the technique for accomplishing the restart are designed into the application program. When multiple batch jobs will be updating the same subset of data base files, synchronous checkpoints are mandatory. To reduce data base recovery complexities, simultaneous on-line and batch updating is prevented through the use of standard operating system facilities.

On-line update programs identify the end of a logical transaction by issuing an end transaction (ET) call to ADABAS. Should a hardware or software failure result in the unscheduled termination of ADABAS, any incomplete logical transactions are automatically removed from the data base when ADABAS again becomes active. This automatic back-out processing insures that the integrity of the data base is maintained.

SECTION 7

WHAT IS OUR STATUS?

Four CIS subsystems have been completed and are servicing the corporation today. Four more are scheduled for completion by the end of our fiscal year on 31 July 1982.

Development costs to date have been high due to initial procurement expenses and mandatory familiarization time for project members to acquaint themselves with the new database environment and their new roles. Having anticipated this phenomena, the CIS project remains on schedule and within its established budget.

SECTION 8

WHAT IS THE FUTURE?

IMMEDIATE PLANS

Immediate plans provide for the design and implementation of selected subsystems in an orderly fashion as we move towards our goal of building the 28 identified subsystems by the end of fiscal year 1986. Our plans remain flexible, however, as corporate requirements and priorities are subject to change.

One of the payoffs of a DBMS supported information system, is that each subsystem adds and draws data from one integrated database. Initial development costs are high as service and control files are established. Once they are in place, subsequent development efforts build on what is already available. Along with the expertise gained by our staff from early implementations, we anticipate subsequent development efforts to be easier and less expensive.

Adapting the Process

Following traditional development and documentation procedures has proven to be a time consuming process. To speed the approach, we are currently investigating the effect of combining some of the planning documents (e.g. Functional Requirements and Functional Design) into one. The use of prototype subsystems as a means of communicating ideas to the users is starting to be explored as well. This approach could quickly provide the framework of a system to the user and make it easier for him to envision how his subsystem will ultimately operate.

LONG RANGE PLANS

Longer range plans address such matters as the impact of extensive CIS development and implementation on the Bedford Computer Center, the feasibility/design of distributing CIS processing across mainframes and minicomputers, and the interconnection or overlap of MITRE's word processing activities with the CIS. The modular, evolutionary approach to building the CIS will allow adaptation to state-of-the-art technology in these areas. The CIS will not include "pioneering" development but will be flexible so that distributed processing can be accomplished as hardware and system software suppliers produce the necessary capabilities for multi-machine, inter-process communication.

DATABASE MACHINES

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

M. L. Stiefel
The MITRE Corporation

1. INTRODUCTION

Over most of its thirty-year history, the data processing community has had a strong interest in techniques that would make computers easier to use in an ever-widening circle of applications. Much of this interest has focused on the employment of database management system (DBMS) as a tool that would allow users to manipulate and retrieve data without wiring programs for each new situation, and that would allow programmers to devote their energies to optimizing the performance and functional capability of the DBMS.

The notions were given a substantial boost in 1970 with Codd's seminal paper (1) introducing the concept of a relational DBMS that would allow users to access data in their own terms, without reference to the physical location of the data on a specific mass storage device.

Unfortunately, it quickly became apparent that, when a relational system was implemented, it was woefully slow, because of the relatively large number of time-consuming disk accesses needed to satisfy all but the most trivial retrieval tasks (i.e., those requiring access to only a single record or to a very small file).

Since 1970, of course, substantial progress has been made in mass storage technology, yielding devices with high capacities and high data transfer rates. At the same time, computer technology has advanced; processors are now fast enough to handle relational algorithms, and main memory is now large enough to hold relational DBMS programs without overlays and without the need to page to and from virtual storage.

Despite these advances, the host computer remains a bottleneck in many DBMS applications where transaction traffic is significant and where the transactions are fairly complex. The problem becomes especially severe when two or more host computers attempt to share a common database; in such instances, a buildup in transaction cues can occur in the computer controlling the database, and response time for the users can be adversely affected.

In light of these lingering difficulties, researchers and designers have turned their attention to an architecture in which a specially designed computer, called a database machine (DBM), handles the DBMS on behalf of a host, in much the same manner that a front-end processor handles mundane communication chores for a host.

In this paper we describe the functions and performance characteristics of DBMs, including machines currently being studied in research laboratories and those currently offered on a commercial basis. Finally, we discuss the cost/benefit considerations that must be recognized in selecting a DBM, and we explore the future outlook for database machines.

2. DBM FUNCTIONAL DESCRIPTION

Figure 1 contrasts the configuration of a system containing a DBM with the configuration of a conventional data processing system. A DBM may be regarded as a computer (typically a minicomputer) that is specially configured to maintain a DBMS, particularly a relational DBMS, working in stand-alone fashion or in conjunction with one or more host computers. As a back-end processor, the DBM accepts commands from any host to update existing entries in the database, to add entries, to delete entries, or to retrieve data that satisfies user-specified criteria. As a stand-alone processor, the DBM accepts commands directly from user terminals and returns requested data to the user terminals.

In each case, a method is provided for translating user requests, stated in a high-level, non-procedural language, into specific low-level commands for access to specific records.

a. In a batch environment, when a host is being used, the user request is compiled; the compiler provides the translation between the user language and the DBM commands, using host-resident software that is furnished with the DBM. The DBM responds to the commands when the compiled batch program is executed and returns the requested data to the batch program for further processing.

b. In an on-line environment, an on-line interpreter provides the translation service. The interpreter may reside in the host or in the DBM. The DBM returns the requested data through the interpreter to the user. As above, the interpreter is part of the DBM package.

As we have noted, the primary purpose of the DBM is to reduce the load on the host and thereby improve system throughput. The DBM may also incorporate special hardware that facilitates the searching of a relational database file -- i.e., a sequential file in which the records are not necessarily sorted in any specific sequence -- at high speed. In particular, the DBM may incorporate a so-called associative memory, in which the data to be manipulated is identified by its contents, rather than by its location. Furthermore, the hardware may include mechanisms that permit the searching of many tracks in parallel.

This function is illustrated in Figure 2. A conventional sequential search, in Figure 2a, is contrasted with a parallel search, in Figure 2b. In Figure 2b, all tracks of a disk are searched at once, and the time needed to retrieve a specific record or group of records is reduced accordingly. In conjunction with the parallel search capability, the associative memory is coupled to a parallel set of processors that compare the contents of each record in the database with the selection criteria established in the user-supplied commands.

FIGURE 1

(a) Conventional Data Processing System

(b) System Incorporating a Data Base Machine

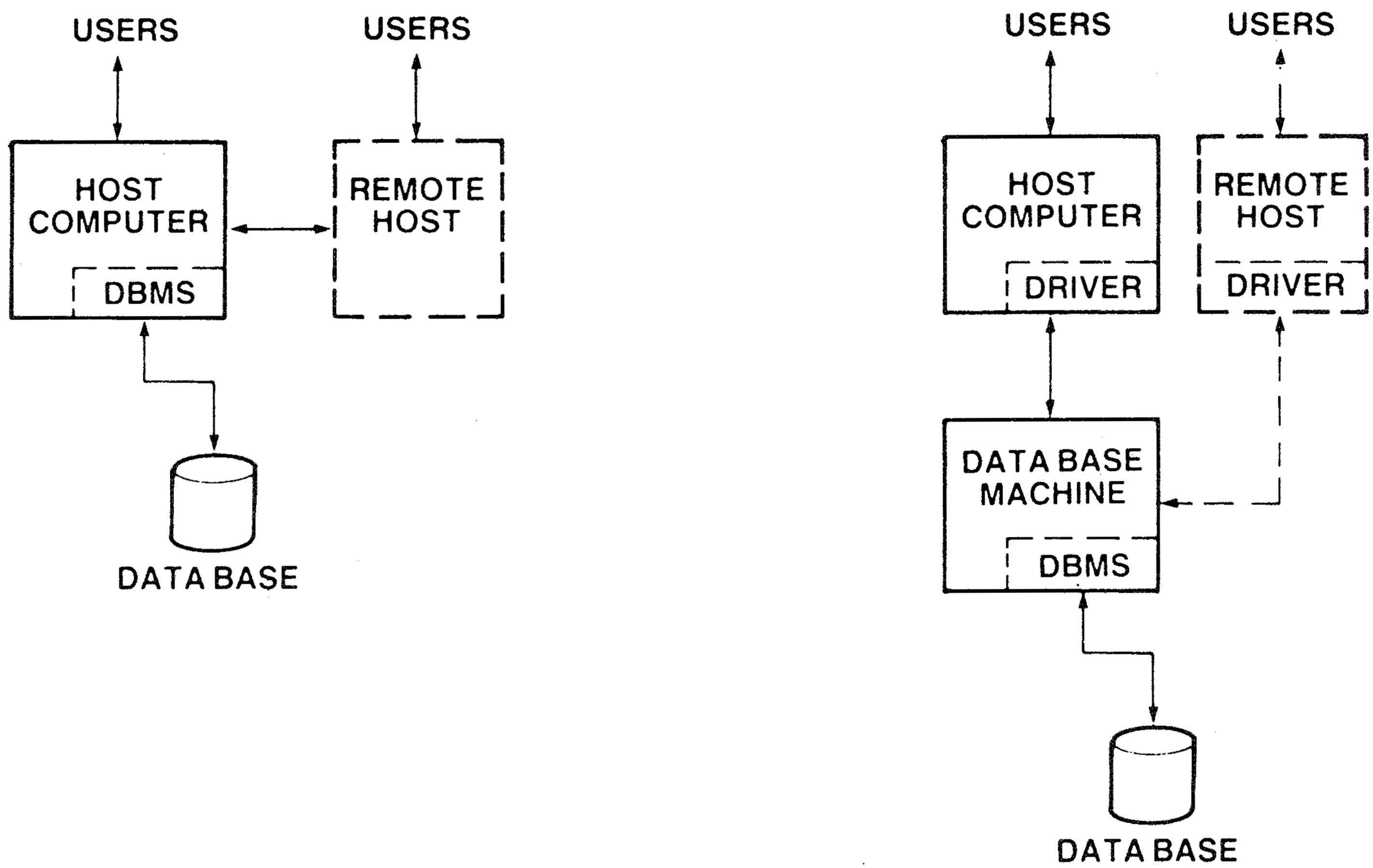


FIGURE 2

Comparison of Read Capabilities

(a) ● **Conventional System Scans a File Sequentially**

- One Track at a Time
- In One Cylinder at a Time

[C1 T1] [C1 T2] • • • [C1 T9] [C2 T1] • • • [C2 T9] • • •

(b) ● **Data Base Machine Scans Files in Parallel**

- All Tracks at a Time
- In One Cylinder at a Time

C1 T1	C2 T1	...	• All 9 Tracks of C1 Read First
C1 T2	C2 T2	...	• All 9 Tracks of C2 Read Second
⋮	⋮		
C1 T9	C2 T9	...	• ETC.

In some systems, a semiconductor cache memory is interposed between the disk and the comparator processors; the cache is searched before the disk is accessed. If the cache is large enough -- typically, one to two megabytes -- then many transactions can be accomplished without reference to disk, considerably reducing the response time.

From this description, it becomes evident that the DBM can be an effective tool in applications where the transaction load is heavy, and where the transactions typically require retrieval of several records from a fairly large file. It is equally evident that the DBM is not cost-effective in every application; any decision to consider addition of a DBM to an existing system must weigh the benefits to be derived against the cost of adding the DBM.

3. CURRENT DATABASE MACHINES

ICL introduced the first commercially available DBM, called the Content Addressable File Store, in 1977, but the device did not gain widespread attention. Today, two U.S. companies offer commercial DBM products: Britton-Lee, of Los Gatos, California, and Intel Corporation, of Austin, Texas. In addition, a fair amount of DBM research is being conducted in in the U.S. and other countries.

3.1 Britton-Lee

The Britton-Lee machine, called the Intelligent Database Machine, or IDM (Figure 3), was first introduced in 1980. The IDM Series now comprises four models, costing $60,000 to $150,000, that interface with various hosts.

The IDM incorporates interfaces to host processors or to intelligent terminals (2). The architecture includes up to four serial I/O processors, each connecting up to eight asynchronous lines, with each line running at up to 19,200 baud. In addition, up to four parallel I/O processors may be included, each operating at 250K bytes per second.

The standard database processor holds up to 3 megabytes of main memory. The optional database accelerator, with a 100-nanosecond instruction time, can search a 2-kilobyte disk page while it is being transferred into memory, and in most cases can complete processing of a page by the time the page has been completely read in. The database accelerator works with only one page at a time; the IDM does not support parallel disk searches.

The system also provides up to four disk controllers, each controlling up to four user-furnished SMD Control Data Corporation 9760- compatible drives. The system disk storage capacity is 32 gigabytes.

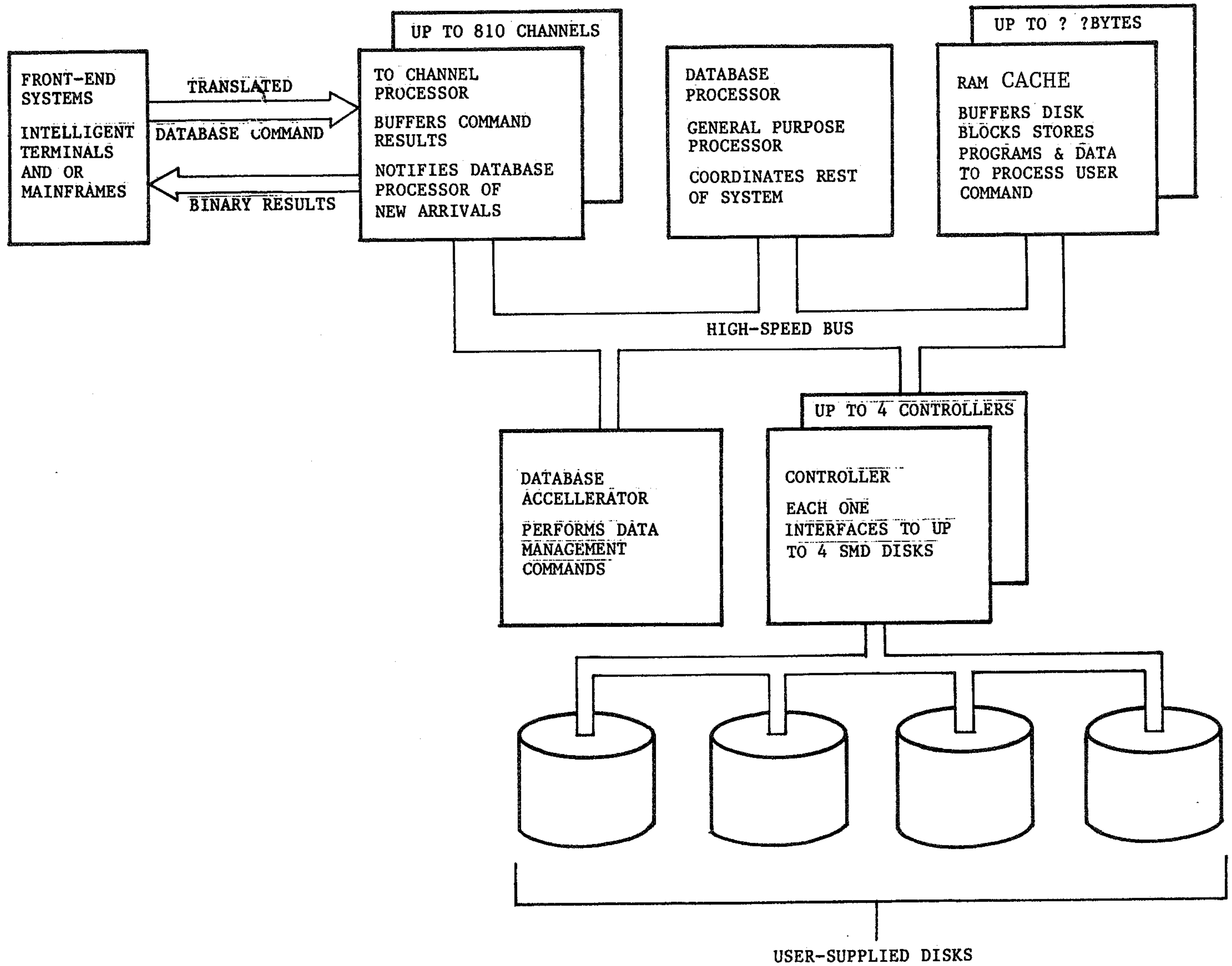
The IDM runs a relational database management system incorporating an integrated data dictionary that is used to define the structure of the DBMS. The DBMS holds up to 50 databases, each containing up to 32,000 separate relations (files). Each relation can hold up to two billion tuples (records). Each tuple can contain up to 250 attributes (fields) with a maximum combined length of 2,000 bytes.

Functions of the DBMS include transaction management, optional logging of database changes, indexing of data for fast access and update, crash recovery, protection against unauthorized users, and data definition and manipulation facilities.

According to the vendor, when the database accelerator is used, nominal throughput is 8 transactions per second for the IDM 200 and 30 transactions per second for the IDM 500.

FIGURE 3

IDM Hardware Architecture



Britton-Lee does not provide the host-resident software that supports the IDM interface; instead, it remains for the host machine vendor, or the original equipment manufacturer (OEM) who is packaging the IDM in a larger system, to write the interface software. Britton-Lee provides a specification of the IDM command set to the OEM; Britton-Lee also provides specifications for parsing statements in a high-level, host-resident query language, called IDL, into IDM commands.

The OEM then creates a turnkey system for the end user, containing the complete set of IDM-resident and host-resident software.

3.2 Intel

The Intel iDBP, introduced in 1982, consists of an 8086-based microcomputer, with 128K to 1M bytes of memory, that controls Winchester or SMD-compatible disk drives and tape drives (3). Like the IDM, the iDBP searches only one disk track at a time. Host interfaces are implemented via serial, parallel, and Ethernet links, making the iDBP suitable for local area network applications.

Intel has not yet announced a price structure for iDBP, but its prices are likely to be comparable to those of the IDM Series, including the cost of DBMS software.

iDBP software provides the "kernel" of a file management system or DBMS. For a relational DBMS, iDBP furnishes the relational operators -- join, select, and project. For databases with hierarchical or network architecture, iDBP provides the tools to implement the database structure, set types, and relationships.

Software functions include: data definition, selection, retrieval, update, and deletion; access and concurrency control; transaction logging, recovery, and restart; file and database backup and recovery; and an integrated data dictionary.

Intel, like Britton-Lee, does not furnish host-resident interface software for iDBP; the system integrator provides this software for the end user.

Throughput figures for the iDBP have not yet been released.

3.3 Research Activities

Many U.S. universities have been conducting DBM research. An early product was the CASSM system from the University of Florida (4). Other universities have since made significant contributions, including Ohio State University, with its DBC machine (5); MIT, with its Infoplex system, a DBM based on complexes of multiple microprocessors (6); the University of Wisconsin, with its DIRECT multiprocessor system (7); and the University of California at Berkeley, with MUFFIN, a distributed DBM (8).

Computer vendors, notably Sperry-Univac, have also been active in DBM research. Sperry recently reported on the results of its DBM architecture studies (9). Sperry favors the DBC architecture devised at Ohio State, because DBC can support the older network-structured database management systems as well the relational ones; this feature clearly addresses the vendor's concern for existing customers that already have a DBMS in place and are not willing to convert to a relational structure in order to add a DBM.

Several other U.S. commercial organizations have shown evidence of DBM developments, notably Memorex, Bell Telephone Laboratories, and Honeywell. IBM has thus far maintained a relatively low profile in the field, but may also offer a DBM after the market becomes better developed. Speculation has persisted in the industry that the next generation of IBM computers will be composed of special-purpose processors: one to handle communications, another for number manipulation, and still another for database maintenance and retrieval.

Elsewhere in the world, DBM studies have been reported in Canada, Turkey, Germany, and Japan. The Japanese activity is of particular interest because it may be the precursor of DBM product announcements that will emanate from that country in the next few years.

4. COST-BENEFIT CONSIDERATIONS

The DBM may be suited to any application that lends itself to a DBMS. Within NASA, appropriate DBM applications may be found in many administrative and scientific tasks. The question then becomes whether the DBM is suited to a specific purpose in a specific environment. The answer depends upon the relative procurement and life cycle costs of systems that incorporate DBMs and those that do not.

The proper time to consider the selection of a DBM is when the architecture of a new information system is being designed or studied, or when a change to a new host is being contemplated for an existing system.

In designing a new system -- say, one that replaces an existing manual operation -- the system architect is presumably free to choose the best hardware and software for the job, without the constraints imposed by investments in existing systems. If a DBMS-oriented approach is contemplated, inclusion of a DBM may permit the procurement of a smaller host machine than would be selected otherwise. The total procurement cost could be reduced if the increase in cost due to the DBM is more than offset by savings in the cost of the host. The life cycle cost of ownership may increase, on the other hand, due to the added cost of maintaining the DBM hardware. (Presumably, the cost of owning the DBMS software would be approximately the same whether the software resided on a host or on a DBM.)

A somewhat different situation pertains when an existing system is being upgraded. In this case, the DBM may be added to an existing system as an alternative to replacing the host with a larger machine. Here, software conversion costs must be computed, along with the cost of any additional DBM-compatible hardware that may be needed. For example, we noted in Sections 3.1 and 3.2 above that the IDM and iDBP systems work with SMD disks; if the existing system uses other types of disks, then it may be necessary to replace them, at a cost that could far exceed the cost of the DBM hardware.

Software conversion may entail restructuring of the existing database and changes in data format. In addition, the user may need to write special interface software that permits an existing query language to be used with the DBM commands.

The feasibility of using a DBM -- whether in a new system or a replacement -- also depends on the expected performance gain that the DBM affords. If the measured or predicted query transaction load in the system constitutes 50% or more of the system workload, then a DBM can be effective (10); otherwise, the performance gain realized with the DBM may not be enough to justify its cost.

5. FUTURE PROSPECTS

In the next few years, the stand-alone mode will become increasingly important, as the DBM becomes integrated with local area networks. Each user will have an intelligent terminal, make the appropriate queries to the DBM, and process the returned information in a dedicated manner. Each terminal will rapidly acquire the power of a mainframe, as the trend toward more and more computing power in smaller and smaller packages continues.

Meanwhile, off-the-shelf DBMs will offer very large cache memories and parallel associative processors for very high speed disk searches, which will further increase the throughput of the DBM.

The DBM promises to become the vehicle researchers have sought since 1970 that will make relational database management systems cost-effective elements in large-scale information systems.

LIST OF REFERENCES

1. Codd, E.F., "A Relational Model of Data for Large Shared Data Banks," Communications of the ACM, Vol. 13, No. 6, June 1970, pp. 377-387.

2. Britton-Lee, Inc., "IDM 500 Intelligent Database Machine Product Description," undated.

3. Intel Corp., "iDBP DBMS Reference Manual," May 14, 1981.

4. Su, S. Y. W. and Lipovski, G. J., "CASSM: A Cellular System for Very Large Data Bases," Proceedings International Conference on Very Large Data Bases, Sept. 1975, pp. 456-472.

5. Banerjee, J., Hsiao, D. K., and Kannan, K., "DBC -- A Database Computer for Very Large Databases," IEEE Transactions on Computers, Vol. c-28, No. 6., June 1979.

6. Madnick, Stuart E., "The Infoplex Database Computer: Concepts and Directions," Proceedings of Spring COMPCON 79, pp. 168-176.

7. DeWitt, David J., "DIRECT - A Multiprocessor Organization for Supporting Relational Data Base Management Systems," Proceedings Fifth Annual Symposium on Computer Architecture,, April 3-5, 1978, pp. 182-189.

8. Stonebreaker, Michael, "MUFFIN: A Distributed Data Base Machine," Proceedings of the First International Conference on Distributed Computer Systems, October 1-5, 1979, pp. 459-469.

9. Freeman, Harvey A., and Jordan, John R., "Another Approach to the Data Base Computer," Proceedings of the ACM, 1980, pp. 382-388.

10. Champine, G. A., "Trends in Data Base Processor Architecture," Proceedings of Spring COMPCON 79, pp. 69-71.

COMPARATIVE ANALYSIS OF DATA BASE MANAGEMENT SYSTEMS

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

Rod Smith
Computer Sciences Corporation
Kennedy Space Center, Florida

Background (Spring 1980)

Institutional data processing support was provided by two pre-third generation computer systems, a Honeywell 635 and an IBM 360, both computer systems utilized operating systems which had not been significantly upgraded in nine years (since 1971). The decision was made to replace those two aged computer systems with an IBM 4341 and to upgrade the peripheral equipment. The IBM 360's operating system, MFT, was upgraded to OS/VS1; MFT did not support the new peripherals. While considering the impact of this upgrade, a study was commissioned to determine if the Remote File Inquiry (RFI) system would handle the future requirements of the user community. RFI is a locally written and locally maintained on-line query/update package.

An RFI User Survey Team was formed, consisting of NASA and Contractor data processing personnel. The scope of the study was, via interviews and panel discussions, to determine the current and future on-line requirements of the user community. Additional consideration was given to the types of data structuring the users required.

The survey indicated the features of greatest benefit were: sort, subtotals, totals, record selection, storage of queries, global updating and the ability to page break.

The major deficiencies were: one level of hierarchy, excessive response time, software unreliability, difficult to add, delete and modify records, complicated error messages and the lack of ability to perform interfield comparisons.

Missing features users required were: formatted screens, interfield comparison, interfield arithmetic, multiple file access, security and data integrity.

The survey team concluded the on-line data processing requirements at Kennedy Space Center were as varied and dynamic as the user organizations themselves. The collective requirements span the full range of DBMS capabilities; there are those users requiring batch update with on-line inquiry, and there are users requiring on-line update and inquiry approaching the real-time.

The survey team recommended Kennedy Space Center move forward to state-of-the-art software, a Data Base Management System which is thoroughly tested and easy to implement and use. This DBMS, at a minimum, must provide the users with all the functionality they enjoyed with the RFI system and provide all the additional capabilities required.

Preliminary Study (Fall 1980)

Acting upon the recommendation of the RFI study team; a comparative analysis of DBMSs for IBM computer systems was begun. Alas, there are literally dozens! Therefore, the first phase of the preliminary study was to reduce the number of candidate DBMSs to a manageable quantity. This was accomplished by establishing a set of mandatory requirements a candidate DBMS must possess to receive further consideration. These mandatory requirements were:

- o Must be IBM 4341 hardware and operating system (OS/VS1) compatible.
- o Must provide on-line query/update, data dictionary and TP monitor capabilities or at a minimum, provide interface to other commercially available packages.

- o Must provide data security to record level.
- o Must provide data integrity.
- o Must support data independence.
- o DBMS must be stable and proven, meaning the features evaluated must have been generally available for not less than one year.
- o Must provide host language interface, at a minimum, COBOL.
- o Must possess general industry acceptance, that is, installed and supported at over two hundred installations.
- o Must support multi-user/multi-thread environments.

The result of the first phase of the preliminary study was the five DBMSs satisfying the basic data processing objectives of the KSC user community. These packages and respective vendors were (alphabetically):

ADABAS	-	Software AG of North America
IDMS	-	Cullinane Database Systems
IMS	-	IBM Corporation
SYSTEM 2000	-	Intel Systems Corporation
TOTAL	-	Cincom Systems, Incorporated

Phase two of the preliminary study was intended to narrow the field of competition to those DBMSs which most probably satisfy all of KSC's requirements. A set of preliminary evaluation criteria was established. The five candidate DBMSs were evaluated and rated based on these criteria. As these criteria are general to all Data Base Management System users, a method of weighting was devised based on KSC's specific needs (a by-product of the RFI study). These evaluation criteria were:

1. On-Line Query/Update

A good on-line query/update capability must be provided. Such a capbility is to be a very high-level, English-like language allowing end-users the ability to make simple and complex inquiries and updates with very little programming know-how. The backbone of strong user-responsive data processing systems is a strong user-friendly query processor; such capability requires the following characteristics:

- o Basic relational and boolean logic operators must be functional, allowing for complex record selection.
- o Capability to easily store and retrieve regularly used queries.
- o Ability to scan records based on a partial key must be provided. A partial key is defined as any fragment of the whole key (i.e., a string of characters starting with any character position in the key).
- o Query output must be able to be sorted in any sequence specified with capability to specify page breaks and subtotals on the sort key. Also, grand totals must be provided on request.

- o Record updating must be flexible enough for global updating as well as one-record-at-a-time updating. Global updating refers to modifying all records meeting a specified criteria.

- o Capability to easily format input screens for use in entering update transactions.

- o Capability to edit input data or decode output data; in short, "table processing".

- o The query/update language must be able to access the full data structure capability of the DBMS.

- o Fields in the same record must be able to be compared with each other.

- o Fields in the same record must be able to be processed with each other using all the arithmetical functions available to the query/update processor. The result of the inter-field calculations must be available for reporting and storing for later processing.

- o Query output can optionally be directed to high-speed printer or other terminals.

- o Security and integrity provisions must be the same as for the DBMS itself.

2. Security

Refers to the additional security capabilities beyond those offered by the operating system and file management system. The DBMS must protect the data base from unauthorized access and update of data at the field level (called "field sensitivity").

3. Integrity

Facilities must be provided to insure the integrity of the data and protect it from CPU or operating system malfunction, direct-access hardware failure, application program error or other system disaster. Evaluated is the success of the DBMS to protect the integrity of the data with little or no human intervention. The DBMS must provide for and monitor concurrent updating.

4. Data Structures

The DBMS must support the three dominant data structures - network, hierarchical and recursive. A network structure implies records can be related in a many-to-many relationship; a hierarchical structure implies a one-to-many record relationship. A recursive structure represents a relationship where one record has a relation to one or more records of its same type. Secondary key refers to the ability to randomly access a record by other than the primary key. KSC requires recursive network structure and secondary key capabilities.

5. Ease of Data Model Change

Refers to the ability of the DBMS to allow changes to the data model after the data model has been established. Evaluated is the ease with which fields, records and relationships can be added to and/or removed from the data model.

6. Ease of Use for Data Base Administrator

Refers to the complexity involved in defining the data structure, data relationships and data formats, to the DBMS. This is commonly called the Data Definition Language (DDL). The DDL should ideally be English-like or at least COBOL-like.

7. Ease of Use for Programmer

Refers to the difficulty involved for programmers to access the data needed to meet their requirements. This is commonly called the Data Manipulation Language (DML). The DML should be English-like or a non-procedural language, ideally the same syntax as the query/update language.

8. Data Dictionary

The Data Dictionary (DD) is the hub of the DBMS. Integrated and non-integrated DD alike must provide these basic characteristics:

- o Provide a common centralized collection of data element descriptions (i.e., name, size, where used, owner, characteristics, description, etc.).
- o File and record relationships are maintained by DD.
- o Provide DBA with various reports to manage data resources.
- o Have batch and on-line maintenance capabilities.

9. Documentation

Rates the vendor-supplied documentation at introductory and technical levels in the areas of organization, completeness, readability, indexing and cross-referencing.

10. Adaptability

Refers to the ability of the DBMS to support a broad range of applications.

11. Random Processing

Many of KSC'c on-line queries are directly to a specific record (via key). Therefore, the DBMS is evaluated on its ability to quickly (minimum I/O) and efficiently (minimum CPU usage) access a record randomly.

12. Report Writer

Evaluates the ability of the DBMS to support the definition of sophisticated report formats, including multiple break-points, subtotals, totals, cross-totals, footings and summaries. Higher rating afforded those DBMSs providing for multiple reports for a single data base pass.

13. Vendor

Refers to two vendor-related topics

- o Vendor credibility as demonstrated by financial position and recent growth patterns.

- o Vendor support - DBMSs requiring very little support are rated highest by their users; however, features like "hot-lines" (24-hours-a-day), local representatives, newsletters and user groups are important.

14. Training

Refers to the amount of training required by Data Base Analysts and Programmers to fully utilize the DBMS; and the availability of the training.

The results of the preliminary study, summarized in Appendix A, were the two DBMSs best satisfying the preliminary evaluation criteria. They were ADABAS and IDMS. The decision was made to include TOTAL in the final evaluation because TOTAL was installed at KSC. TOTAL was used by a KSC tenant; who has since acquired his own IBM 4300 and left the center.

Final Evaluation (December 1980)

A four person DBMS Study Team was appointed and began the comprehensive final evaluation process of the three remaining DBMSs. The Study Team refined the preliminary evaluation criteria for use as the final evaluation criteria. The refinements were; to eliminate Adaptability and Training and to add the following six criteria:

o Physical Storage Management

The DBMS will be evaluated based on the following criteria:

- Buffer strategy - Methodology employed to handle the memory-resident buffers.

- The methodology employed to maintain the free space generated by record deletion/modification.

- The methodology employed to handle overflow of data records.

o Program/Data Independence

Refers to the ability of a data base to change while programs remain unchanged. This shall be accomplished without recompilation of host language programs so long as the programs do not access deleted or added data fields or relationships.

o Time-Sharing Subsystem

To facilitate system development and modification, a time-sharing subsystem capable of performing the following functions is required:

- Text entry and editing
- Interface with file management system
- Remote job processing
- Interactive programming

o Host Language Interface

The DBMS must be capable of interfacing with COBOL, Assembler Programming Language and FORTRAN.

o Utilities

The utility programs are to permit efficient loading and subsequent maintenance of the data base. Functions include:

- File loading, unloading and deletion
- Addition of physical storage space for the data base
- Save and restore of files
- Data base status reporting
- Restructuring

o Staff

Refers to personnel required to support the successful operation of a DBMS. Included are the requirements for:

A. Systems Programmers

B. Data Base Analysts

C. Programmer/Analysts

The Study Team assigned a weighting factor to each criterion to demonstrate its importance to the specific requirements of Kennedy Space Center. A total of one hundred points was divided among the eighteen criteria based on the relative importance (to KSC) of the criterion in relation to the other criteria. Appendix B provides a list of the final evaluation criteria and associated weighting factors.

Each vendor was asked to make a technical presentation of its DBMS products. During these informal presentations team members interrogated the vendors on the methods employed by their DBMSs to meet KSC's criteria. Team members requested and received detail technical documentation on all aspects of the vendor's products. After the vendor presentations were completed, the DBMS Study Team met daily for "round-table" discussion and assignment of tasks. Each criterion was fully investigated for each DBMS, usually by two or more team members. Users of the candidate DBMSs were contacted to verify the vendors' statements and to solicit the users' experiences with the DBMS.

The study concluded the best data base management system available in the software marketplace for Kennedy Space Center's institutional data processing user community needs was ADABAS. ADABAS is particularly strong in the areas of On-Line Query/Update, Data Structures, Security, Integrity, Physical Storage Management and Ease of Use.

Procurement (April 1981)

An open procurement procedure was followed by advertising KSC's DBMS requirements and intentions in the Commerce Business Daily and conducting a benchmark.

ADABAS Evaluation (Fall 1981)

Kennedy Space Center's Computer Support Division leased the Software AG products for three months so a thorough evaluation could conclusively determine whether or not the ADABAS software products performed as documented.

The ADABAS Evaluation appraised, (1) all features of ADABAS in terms of KSC's minimum requirements and (2) as many additional features as feasible.

The ADABAS Evaluation Team concluded the products performed as documented and the products are well suited to satisfy the data processing needs of the Kennedy Space Center Institutional Computer Users for the next decade.

APPENDIX A

CRITERION	WEIGHT	ADABAS		IDMS		IMS		TOTAL		SYSTEM 2000	
On-Line Query/Update	4	10	40	7	28	0	0	7	28	8	32
Security	3	9	27	8	24	9	27	6	18	7	21
Integrity	3	9	27	9	27	9	27	8	24	8	24
Data Structures	2	8	16	9	18	7	14	8	16	7	14
Ease of Data Model Change	2	9	18	9	18	7	14	8	16	7	14
Ease of Use for DBA (DDL)	2	5	10	9	18	5	10	7	14	7	14
Ease of Use for Programmer (DML)	2	8	16	9	18	7	14	7	14	7	14
Data Dictionary	2	8	16	10	20	8	16	8	16	8	16
Documentation *	2	8	16	8	16	7	14	7	14	7	14
Adaptability	2	8	16	8	16	8	16	8	16	7	14
Random Processing	1	7	7	10	10	8	8	10	10	7	7
Report Writer	1	9	9	9	9	7	7	10	10	9	9
Vendor	1	7	7	8	8	9	9	7	7	8	8
Training *	1	8	8	8	8	7	7	7	7	9	9
Overall (Maximum = 280)		233		238		183		210		210	

* - Data Pro 1979 User Survey

MATRIX OF DBMS RATINGS

APPENDIX B

FINAL EVALUATION CRITERIA

Criterion	Weighting Factor
On-Line Query/Update	22
Data Structures	9
Security	8
Integrity	8
Physical Storage Management	7
Ease of Data Model Change	7
Ease of Use for Programmer	6
Data Dictionary	6
Ease of Use for DBA	5
Program/Data Independence	4
Time-Sharing Subsystem	4
Documentation	3
Vendor	3
Random Processing	2
Host Language Interface	2
Utilities	2
Staff	1
Report Writer	1

RAMIS

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

Jim Head
Goddard Space Flight Center

Figure 1 was extracted from Datamation in 1980, and I ran across it again recently in a sales brochure in RAMIS.

As you know the cost of programming services has more than doubled, and the price of a byte of memory today is less than the maintenance cost of that byte five years ago. In 1975, it cost $65,000 to rent 1MB of ampex memory. In February, we bought 4MB of IBM memory for $56,000 for our 4341.

When I first programmed, we had to desk check because we only got one shot a day. Now with all the software products available, like ABEND-AID and CAPEX, on-line compiles and testing, it takes all the fun out of programming.

I think every organization goes through the same growing pains and problems that we went through, and we can all find ourselves somewhere in Figure 2.

Around 1966 we are beginning to do Centerwide applications, not just Payroll and Accounting.

The IBM 300/40 was installed in December 1970. This was our hardest conversion, AUTOCODER or COBOL programs. Anyone remember ACCCP?

In the early 70's we were strictly a COBOL shop - quick turnaround on new applications or reports was weeks not days. As you all know, that was not received very well.

Figure 1

SYSTEMS DEVELOPMENT

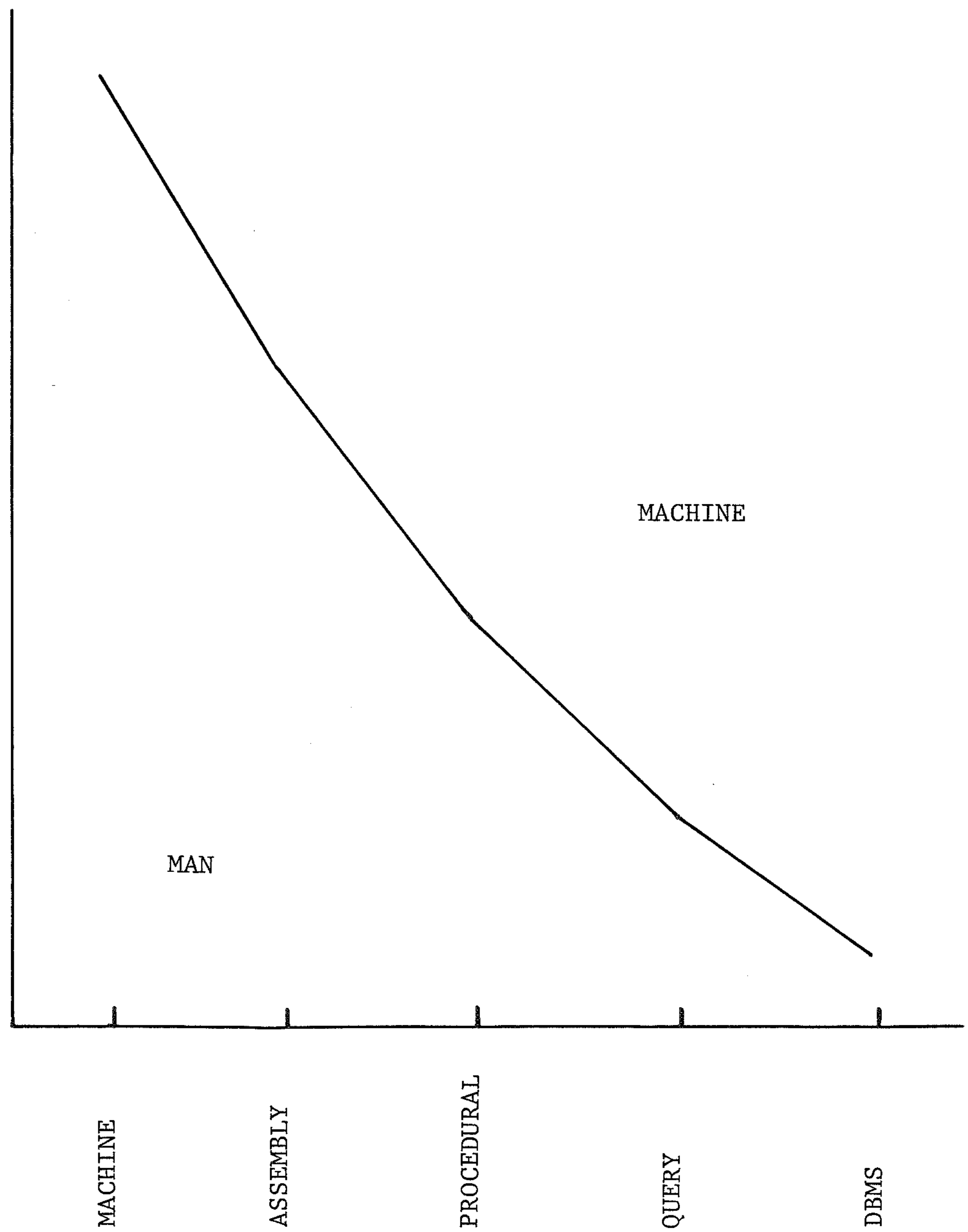


FIGURE 2

GODDARD SPACE FLIGHT CENTER
MANAGEMENT OPERATIONS DIRECTORATE
MANAGEMENT SYSTEMS OFFICE

CHRONOLOGICAL DEVELOPMENT OF THE
ADMINISTRATIVE INFORMATION PROCESSING
FUNCTION AT GODDARD SPACE FLIGHT CENTER

1960 - ADMINISTRATIVE INFORMATION PROCESSING FUNCTION ESTABLISHED AS BUSINESS DATA BRANCH OF FINANCIAL MANAGEMENT DIVISION. TABULATING EQUIPMENT AND PUNCH CARDS USED

1962 - IBM 1401 COMPUTER OBTAINED TO SUPPLEMENT TABULATING EQUIPMENT

1965 - IBM 1410 COMPUTER ADDED TO ALLOW MORE ADMINISTRATIVE FUNCTIONS TO BE PROCESSED

1973 - IBM 360/40 COMPUTER REPLACES IBM 1410

1974 - IBM 360/40 UPGRADE TO 384K

1975 - IBM 360/50 REPLACES 360/40, COMPUTER CAPACITY GREATLY INCREASED, PROCESSING SEVERAL APPLICATIONS CONCURRENTLY POSSIBLE

1976 - COMMUNICATIONS HARDWARE OBTAINED TO PERMIT ONLINE ACCESS TO COMPUTER, INITIALLY USED PRIMARILY FOR SYSTEMS DEVELOPMENT ACTIVITIES

1978 - NEW SOFTWARE (RAMIS) OBTAINED TO PERMIT ONLINE AD HOC INQUIRIES BY USER COMMUNITY

1978 - DEVELOPMENT OF FIRST MAJOR ONLINE SYSTEM STARTS (SMALL PURCHASES TRACKING)

1979 - IBM 360/65 REPLACES 360/50 TO PERMIT FASTER ONLINE PROCESSING AND TO INCREASE TOTAL CAPACITY

1980 - NEW PERSONNEL SYSTEM INSTALLED TO REDUCE COSTS AND IMPROVE SERVICE

1981 - IBM 4341 II REPLACES IBM 360/65

In 1973, we purchased IRS (Inquiry and Reporting System) from Sigma Data, and we could now produce one time ad-hoc batch reports in a timely fashion, and we are still running those same one-time-only reports.

As in Figure 1, there seemed to be a void being filled constantly by friendly languages - more computer generated statements and assistance. RAMIS claims to reduce the required instructions by 40:1 over COBOL.

In August of 1977, we put together a study (Figure 3) on the need for an on-line Inquiry System. We developed the following requirements Matrix and RAMIS, INQUIRE, and EASYTRIEVE qualified for further study. (Figure 4) INQUIRE declined to install their system for a free trial and, lacking record arithmetic, were eliminated from further consideration.

We chose RAMIS because of its English-like language, its flexibility (it will treat non-RAMIS files, either ISAM,USAM, TOTAL, or QSAM with the same nonprocedural language), and its growth potential.

RAMIS was a complete Data Base Management System. We felt that records management was its weakest point, but then in those days EASYTRIEVE didn't have its own capability - so I guess you can say it was another plus for RAMIS.

FIGURE 3

NEED FOR INQUIRY SYSTEM

- CHANGE IN DATA STRUCTURES
- CHANGE IN REPORTING REQUIREMENTS
- REDUCTION IN STAFF LEVELS
- DIFFERENT VIEWS OF RELATED DATA
- USER REPORTING
- REDUCE DEVELOPMENT TIME
- REDUCE MAINTENANCE COSTS

FIGURE 4

ATTRIBUTES

PRODUCTS	COST	FILE PROCESSING abcd	DBMS CAPABILITIES ab	SECURITY abcd	SYNTAX abc	COMPUTATION abcd	QUERY AIDS abc	MODE OF OPERATION ab	QUALIFY FOR FURTHER STUDY
ASI-ST	79.0	1111	01	1000	X 101	1110	111	11	
DSR	52.6	X 1110	X 00	1111	111	1110	111	11	
EASYTRIEVE	20.0	1111	01	1110	X 110	1111	111	11	
GPS	0	X 0100	01	1000	X 110	X 1000	100	11	
INQUIRE	42.5	1111	10	1111	111	X 1100	111	11	
IRS	0	0111	X 00	1000	X 000	1110	X 100	11	
MISURS		X 1000	01	1111	X 011	X 0000	111	11	
MRCS	22.0	1111	X 00	1000	X 101	1111	111	11	
QUERYS	17.0	1111	X 00	1001	111	1110	111	11	
RAMIS	38.5	1111	11	1111	111	1111	111	11	
RFI		X 1000	X 00	1000	X 011	X 0010	X 100	X 01	
SOCRATES	20.0	1111	01	1000	X 100	1110	X 100	11	
MINIMUM REQUIREMENTS	50.0	?111	?1 1?	1???	111	?11?	11?	11	

? NON-ESSENTIAL REQUIREMENTS

X DOES NOT QUALIFY IN CATEGORY

COSTS WERE BASED UPON DATAPRO FIGURES, WHERE APPLICABLE

PRODUCT RATING MATRIX

Whether it was skill or just good fortune, we have always been very pleased with our selection.

Mathematica has been a very friendly company and responsive to user suggestions. Their enhancements to the product and announcements of new companion products keep pace with our learning curve. But the most impressive feature of RAMIS is the devotion of its users. I've found very few people dissatisfied with the product. A survey in a recent Datamation supports our feelings. In this survey, it was ranked number one.

Our first application to be developed using RAMIS was our reduction in force in 1978. We were "requested" to have personnel and "RIF" information available for on-line inquiry. At this point we were very low on the learning curve and "panic" was the word of the day.

Fortunately, RAMIS was user friendly and forgiving and we were able to satisfy some very interesting inquiries with RAMIS nonprocedural language.

Summary reports of impacts by organization and skills affected could be developed in a few statements. For example, see Figures 5 and 6.

FIGURE 5

```
TABLE
FILE PE-ACTIVE
COUNT ENTRIES AS '' AND ROW-TOTAL AND COLUMN-TOTAL
ACROSS DIR
BY SKI
IF DUTY EQ D
END
```

FIGURE 6

RP0808: NUMBER OF RECORDS IN TABLE= 3736 LINES= 7
Page 1

SKI \ DIR	000	100	200	300	400	500	600	700	800	900	TOTAL
1	2	0	147	0	0	0	0	0	0	0	149
2	0	3	16	0	3	3	0	1	1	4	31
3	122	0	85	21	3	16	54	236	33	41	611
5	28	65	182	10	44	27	30	45	53	52	536
6	16	60	327	5	90	20	21	29	103	39	710
7	77	8	20	79	194	153	212	398	208	348	1697
9	2	0	0	0	0	0	0	0	0	0	2
TOTAL	247	136	777	115	334	219	317	709	398	484	3736

After the RIF was completed we still had an interested user in our Personnel Department, so we soon had the requirement to teach the reporting capability to Personnel Specialists. This soon extended to training and locator systems and to the development of a skills inventory we call Human Resources Inventory System.

We now have systems -- or at least the ability to report -- as an on-line inquiry from an interesting number of systems across all areas.

For example, we have access to information in the following systems:

* Capital assets
* Disposal of Property
* Accounting
  * Budget
  * Travel
  * Status
* Manpower
  * Plans
  * Actuals
  * Plans/Actuals
  * Available

* Procurement
* Project Management
* Reports Distribution
* Supplies
* Personnel
* Training

RAMIS is a general purpose DBMS with appeal to both technical and non-technical users. Efficient hierarchies can be developed providing data independence, multiple views of the data, and complex networks. All the power of the nonprocedural language can then be used by nontechnical people like business specialists for reporting or ad-hoc queries.

To give you an idea of the power and flexibility of the system I would like to show you a few examples, using Figures 7-11.

First is a system I developed that starts with creating a data set from the volume table of contents of our disk packs and matching the data set names to the system catalog.

FIGURE 7

```
MODEL FOR RAMIS FILE :: DBA
PRODUCED ON 05/17/82 AT 8.44.22

LEVEL    FACTOR                                      FORMAT     SYNONYM
-----    ------                                      -------    -------
  1 D      0      :::::::::::::::::::::::::
                  :   1 VOL-SER           :          A 6        VS
                  :::::::::::::::::::::::::
                              :
                              :
                              V
  2 S      0      :::::::::::::::::::::::::
                  :   2 DSN               :          A 32       DSN
                  :   3 DSN               :          A 13       DSNS
                  :   4 TSOID             :          A 5        TSO
                  :   5 SYSTEM-ID         :          A 2        SID
                  :   6 FREQ              :          A 1        F
                  :   7 DSORG             :          A 2        DO
                  :   8 RECFM             :          A 2        RF
                  :   9 ALLOC             :          A 3        AL
                  :  10 EXT               :          A 2        EXT
                  :  11 SECURE            :          A 1        SEC
                  :  12 LAST-USED         :          A 5        LU
                  :  13 USE-COUNT         :          A 5        UC
                  :  14 CRT-DATE          :          A 5        CD
                  :  15 LRECL             :          A 4        LR
                  :  16 BLKSIZE           :          A 5        BLK
                  :  17 PDSA              :          A 2        PA
                  :  18 PDSU              :          A 2        PU
                  :  19 TYPE              :          A 1        TYPE
                  :  20 TASK-NO           :          T 6        TN
                  :  21 TRACKA            :          ( 6        TA
                  :  22 TRACKU            :          ) 6        TU
                  :::::::::::::::::::::::::
```

FIGURE 8

```
MODEL FOR RAMIS FILE :: DBA
PRODUCED ON 05/17/82 AT 8.44.22

LEVEL      FACTOR                                       FORMAT      SYNONYM
-----      ------                                       -------     -------
  1 D        0        ::::::::::::::::::::::::
                      :   1 VOL-SER            :          A 6          VS
                      ::::::::::::::::::::::::
                                 :
                                 :
                                 V
  2 S        0        ::::::::::::::::::::::::
                      :   2 DSN                :          A 32         DSN
                      :   3 DSN                :          A 13         DSNS
                      :   4 TSOID              :          A 5          TSO
                      :   5 SYSTEM-ID          :          A 2          SID
                      :   6 FREQ               :          A 1          F
                      :   7 DSORG              :          A 2          DO
                      :   8 RECFM              :          A 2          RF
                      :   9 ALLOC              :          A 3          AL
                      :  10 EXT                :          A 2          EXT
                      :  11 SECURE             :          A 1          SEC
                      :  12 LAST-USED          :          A 5          LU
                      :  13 USE-COUNT          :          A 5          UC
                      :  14 CRT-DATE           :          A 5          CD
                      :  15 LRECL              :          A 4          LR
                      :  16 BLKSIZE            :          A 5          BLK
                      :  17 PDSA               :          A 2          PA
                      :  18 PDSU               :          A 2          PU
                      :  19 TYPE               :          A 1          TYPE
                      :  20 TASK-NO            :          I 6          TN
                      :  21 TRACKA             :          I 6          TA
                      :  22 TRACKU             :          I 6          TU
                      ::::::::::::::::::::::::


                      ASSOCIATED FILE IS
                         TSOIDS
                      ::::::::::::::::::::::::
                      :  23 FIRST-NAME         :          A 10         FIN
                      :  24 MI                 :          A 1          MI
                      :  25 LAST NAME          :          A 14         LN
                      :  26 GROUP              :          A 1          GRP
                      :  27 ORG                :          A 5          ORG
                      :  28 DIR                :          A 1          DIR
                      :  29 APPLICATION        :          A 30         APP
                      :  30 SPACE              :          A 2          SP
                      ::::::::::::::::::::::::
```

FIGURE 9

```
MODEL FOR RAMIS FILE :: DBA
PRODUCED ON 05/17/82 AT 8.44.22

LEVEL    FACTOR                                     FORMAT       SYNONYM
-----    ------                                     -------      -------
  1 D      0      :::::::::::::::::::::::::
                  :   1 VOL-SER          :            A 6          VS
                  :::::::::::::::::::::::::
                             :
                             :
                             V
  2 S      0      :::::::::::::::::::::::::
                  :   2 DSN              :            A 32         DSN
                  :   3 DSN              :            A 13         DSNS
                  :   4 TSOID            :            A 5          TSO
                  :   5 SYSTEM-ID        :            A 2          SID
                  :   6 FREQ             :            A 1          F
                  :   7 DSORG            :            A 2          DO
                  :   8 RECFM            :            A 2          RF
                  :   9 ALLOC            :            A 3          AL
                  :  10 EXT              :            A 2          EXT
                  :  11 SECURE           :            A 1          SEC
                  :  12 LAST-USED        :            A 5          LU
                  :  13 USE-COUNT        :            A 5          UC
                  :  14 CRT-DATE         :            A 5          CD
                  :  15 LRECL            :            A 4          LR
                  :  16 BLKSIZE          :            A 5          BLK
                  :  17 PDSA             :            A 2          PA
                  :  18 PDSU             :            A 2          PU
                  :  19 TYPE             :            A 1          TYPE
                  :  20 TASK-NO          :            I 6          TN
                  :  21 TRACKA           :            I 6          TA
                  :  22 TRACKU           :            I 6          TU
                  :::::::::::::::::::::::::

                  ASSOCIATED FILE IS
                     TSOIDS
                  :::::::::::::::::::::::::
                  :  23 FIRST-NAME       :            A 10         FIN
                  :  24 MI               :            A 1          MI
                  :  25 LAST NAME        :            A 14         LN
                  :  26 GROUP            :            A 1          GRP
                  :  27 ORG              :            A 5          ORG
                  :  28 DIR              :            A 1          DIR
                  :  29 APPLICATION      :            A 30         APP
                  :  30 SPACE            :            A 2          SP
                  :::::::::::::::::::::::::

                  ASSOCIATED FILE IS
                  SYSTEM-ID
                  :::::::::::::::::::::::::
                  :  31 SYS-NO           :            A 4          SNO
                  :  32 MAINT-NO         :            A 6          MN
                  :  33 SYS-NAME         :            A 28         SN
                  :  34 USER             :            A 32         U
                  :  35 ANALYST          :            A 2          AN
```

FIGURE 10

MODEL FOR RAMIS FILE :: DBA
PRODUCED ON 05/17/82 AT 8.44.22

LEVEL	FACTOR	FIELD	FORMAT	SYNONYM
1 D	0	1 VOL-SER	A 6	VS
2 S	0	2 DSN	A 32	DSN
		3 DSN	A 13	DSNS
		4 TSOID	A 5	TSO
		5 SYSTEM-ID	A 2	SID
		6 FREQ	A 1	F
		7 DSORG	A 2	DO
		8 RECFM	A 2	RF
		9 ALLOC	A 3	AL
		10 EXT	A 2	EXT
		11 SECURE	A 1	SEC
		12 LAST-USED	A 5	LU
		13 USE-COUNT	A 5	UC
		14 CRT-DATE	A 5	CD
		15 LRECL	A 4	LR
		16 BLKSIZE	A 5	BLK
		17 PDSA	A 2	PA
		18 PDSU	A 2	PU
		19 TYPE	A 1	TYPE
		20 TASK-NO	I 6	TN
		21 TRACKA	I 6	TA
		22 TRACKU	I 6	TU

ASSOCIATED FILE IS
TSOIDS

FIELD	FORMAT	SYNONYM
23 FIRST-NAME	A 10	FIN
24 MI	A 1	MI
25 LAST NAME	A 14	LN
26 GROUP	A 1	GRP
27 ORG	A 5	ORG
28 DIR	A 1	DIR
29 APPLICATION	A 30	APP
30 SPACE	A 2	SP

ASSOCIATED FILE IS
SYSTEM-ID

FIELD	FORMAT	SYNONYM
31 SYS-NO	A 4	SNO
32 MAINT-NO	A 6	MN
33 SYS-NAME	A 28	SN
34 USER	A 32	U
35 ANALYST	A 2	AN

FIELD	FORMAT	SYNONYM
39 TASK TITLE 1	A 25	TITLE 1
40 TASK TITLE 2	A 25	TITLE 2
41 TASK TYPE	I 1	TP
42 TASK COMPLEX	A 2	COMPLEX
43 PRIORITY	A 1	PRTY
44 TASK ORIGIN	I 1	ORIGIN
45 MSO ANALYST	A 18	ANAL
46 CTR EST DATE	A 6	CECO
47 CTR EST HRS	I 5	APHR
48 PROG ASSIGN	A 18	PROGA
49 ACT DATE	A 6	APHR
50 ACT PROG HRS	I 5	APHR
51 EVAL TOTAL	I 3	ET
52 APPROV DATE	A 6	APD
53 RECEIVED	A 6	RECEIV

FIGURE 11

```
MODEL FOR RAMIS FILE :: DBA
PRODUCED ON 05/17/82 AT 8.44.22

LEVEL      FACTOR                                   FORMAT   SYNONYM
-----      ------                                   -------  -------
 1 D        0       ::::::::::::::::::::::::
                    :   1 VOL-SER          :         A 6        VS
                    ::::::::::::::::::::::::
                              :
                              :
                              V
 2 S        0       ::::::::::::::::::::::::
                    :   2 DSN              :         A 32       DSN
                    :   3 DSN              :         A 13       DSNS
                    :   4 TSOID            :         A 5        TSO
                    :   5 SYSTEM-ID        :         A 2        SID
                    :   6 FREQ             :         A 1        F
                    :   7 DSORG            :         A 2        DO
                    :   8 RECFM            :         A 2        RF
                    :   9 ALLOC            :         A 3        AL
                    :  10 EXT              :         A 2        EXT
                    :  11 SECURE           :         A 1        SEC
                    :  12 LAST-USED        :         A 5        LU
                    :  13 USE-COUNT        :         A 5        UC
                    :  14 CRT-DATE         :         A 5        CD
                    :  15 LRECL            :         A 4        LR
                    :  16 BLKSIZE          :         A 5        BLK
                    :  17 PDSA             :         A 2        PA
                    :  18 PDSU             :         A 2        PU
                    :  19 TYPE             :         A 1        TYPE
                    :  20 TASK-NO          :         I 6        TN
                    :  21 TRACKA           :         I 6        TA
                    :  22 TRACKU           :         I 6        TU
                    ::::::::::::::::::::::::

                    ASSOCIATED FILE IS
                       TSOIDS
                    ::::::::::::::::::::::::
                    :  23 FIRST-NAME       :         A 10       FIN
                    :  24 MI               :         A 1        MI
                    :  25 LAST NAME        :         A 14       LN
                    :  26 GROUP            :         A 1        GRP
                    :  27 ORG              :         A 5        ORG
                    :  28 DIR              :         A 1        DIR
                    :  29 APPLICATION      :         A 30       APP
                    :  30 SPACE            :         A 2        SP
                    ::::::::::::::::::::::::
```

```
ASSOCIATED FILE IS
SYSTEM-ID
:::::::::::::::::::::::::
:  31 SYS-NO              :        A 4        SNO
:  32 MAINT-NO            :        A 6        MN
:  33 SYS-NAME            :        A 28       SN
:  34 USER                :        A 32       U
      ANALYST             :        A 2        AN
:  39 TASK TITLE 1        :        A 25       TITLE 1
:  40 TASK TITLE 2        :        A 25       TITLE 2
:  41 TASK TYPE           :        I 1        TP
:  42 TASK COMPLEX        :        A 2        COMPLEX
:  43 PRIORITY            :        A 1        PRTY
:  44 TASK ORIGIN         :        I 1        ORIGIN
:  45 MSO ANALYST         :        A 18       ANAL
:  46 CTR EST DATE        :        A 6        CECO
:  47 CTR EST HRS         :        I 5        APHR
:  48 PROG ASSIGN         :        A 18       PROGA
:  49 ACT DATE            :        A 6        APHR
:  50 ACT PROG HRS        :        I 5        APHR
:  51 EVAL TOTAL          :        I 3        ET
:  52 APPROV DATE         :        A 6        APD
:  53 RECEIVED            :        A 6        RECEIV
::::::::::::::::::::

ASSOCIATED FILE IS
   JCLCROSS
::::::::::::::::::::
:  55 LDSN                :        A 44       LDSN
:  56 PROCLIB             :        A 10       PLIB
:  57 PROCEDURE           :        A 8        PROC
:  58 STEPNAME            :        A 8        STEP
:  59 DDNAME              :        A 8        DD
:  60 PROGRAM             :        A 8        PGM
:  61 DISPOSITION         :        A 16       DISP
:  62 UNIT                :        A 10       VS
:  63 FLG                 :        A 1        F1
::::::::::::::::::::::::
:  35
::::::::::::::::::::

ASSOCIATED FILE IS
   TMW52001
::::::::::::::::::::
:  36 SYSTEM NO           :        I 4        STM
:  37 TASK NO             :        I 6        TKNO
:  38 MSO DUE DATE        :        A 6        MDD
```

Within minutes I can list all ISAM data sets. This could be very useful if we were interested in looking at IAM (by innovative data processing) - a replacement product for ISAM and VSAM.

I could find the most often used data set - possibly to improve channel usage.

The least used - as candidates for migration to tapes. There are no restrictions on the nonprocedural language. Any field can be a control field or a break (subtotal) field. By 1980 we had over 100 TSO users and TSO disk space became a problem to control. In Figure 8 we have added another file to the system to show the network capabilities of RAMIS.

We maintained our TSO ID file on a separate RAMIS Data Base. The file is updated via another RAMIS feature called PROMPT.

PROMPT would "prompt" you through the data dictionary and you could fill in the information.

With the TSOID field in the host file (DBA) I could network into the TSOID file (associated file) and pick the information. To the nonprocedural language it is as though I simply added elements to the file.

The TSOID file is in no way affected by the network. It is still maintained independently. Any number of files can network to the TSOID file.

Currently RAMIS can network up to 25 different files and can have any number of "VIEWS" of the main or host file. As shown in Figure 9, the System-ID file adds another file to the network. It simply shows how to use a file to eliminate redundant data. RAMIS has a masking feature and by requiring certain standards in naming conventions we were able to network into our contractor file and track active tasks. Obviously inactive tasks could be identified and their data sets deleted. This is shown in Figure 10.

Figure 11 shows another valuable feature. The JCLCROSS file contains all our production JCL, and by defining the network in a certain way I can find all of the matching records in the associated file network. For example, which programs and which systems are referenced by a particular data set.

This became an invaluable tool during system modifications. Which programs read the Personnel master can easily be answered.

RAMIS FEATURES

Figure 12 shows features of the RAMIS system which include:

System Features

* OS, VS, TSO, VS370, DOS/VS, VM/CMS

* Batch and Interactive

* Environment Controls

* Buffers (network pointer strategy, library data base)

Data Base Features

* Large data bases and files

* Reporting

* catalogued or ad-hoc

* RAMIS/Non-RAMIS files

* tabular reports

* graphs

* summary and/or detail

* masking

* selectors (if)

* ascending, descending sort sequence

* extensive computation capability

* column calculations

Programmer Interface

* COBOL, FORTRAN, PL1, Assembler, Complete Access/Update control with Read Only Options

Run Executive

* powerful environment control
* prompting for non ADP users
* easily modified
* interactive procedures
* invoke other procedures
* system information available
* testing facilities

Financial Planning

* Report Models

SCAN

* browsing capabilities
* update capabilities
* delete one or many records based on key

Automated Interface

* APL
* ADABAS
* TOTAL
* PL1
* IDMS
* IMS

Recent Announcements

Graphics

Relate

Full Screen Manager

* Statistical Interfaces
* Table lookup

FIGURE 12

RAMIS FEATURES

SYSTEMS FEATURES

ENVIRONMENT CONTROLS

DATA BASE FEATURES

RECORDS MANAGEMENT

REPORTING

PROGRAMMERS INTERFACE

RUN EXECUTIVE

FINANCIAL PLANNING

SCAN

AUTOMATIC INTERFACE

RECENT ANNOUNCEMENTS

Figure 13 gives examples of records management using networks:

* 24 different data bases may be concatenated for reporting

* Hierarchical file structure

* Separate files easily networked together and transparent to the user

* Automatic pointer maintenance

* Directory

* Alpha, binary, packed, single & double precision data types

* exponential

* non-procedural language interface

* concurrent access for reporting

* security
  scrambled passwords, data base, file, record
  data item access control

* Redefinition of element descriptions

* Field name or synonym

* Data Dictionary

FIGURE 13

```
USE
DATABASE
CODEPRC
END
ERASE
FILE DBA
END
REVISE
DIRECT
INCLUDE
ERRORSUM
KEY VS. DSN
JUSTIFIED
FORM 13,x-13,32,X1,5,X1,2,X1,1,2,2,3,2,
FORM 1,5,X4,5,X2,5,6,X3,6,4,4,4,
FORM X2,4,X5,2,X11,1
ORDER DSNS,DSN,TSO,SID,F,DO,RF,AL,EXT
ORDER TU,PA,PU,TYPE
READ DBA
FILE DBA
END
```

Figure 14 describes the system's capacity for Records Management including:

* verification
* addition
* change
* delete
* editing.
  accumulation or replacement update: one transaction updating multiple records

Figure 15 is an example of the ability to generate new fields and values including:

* generated fields
* control of search argument
* any field can be a key Field
* extensive log capabilities
* multiple transaction types
* audit trails
* mock update.

FIGURE 14

```
COMPUTE
FILE MNPR
ACCEPT = IF TC EQ 'D' THEN 'YES' ELSE 'TT  ':
3T = 'D';
END
REVISE
COMPUTE
INCLUDE 59
KEY ORG, UPN7,SKI, SKD
UPDATE BT
FORM 3,7,1,1,7,x105,1
ORDER ORG, UPN7,TBT, SK1, SKD, TC
ERRORSUM
READ MNPR
END
RAMREORG MNPR,MSGL=0, IFL = IF BT NE D
```

FIGURE 15

```
                    CATALOGS DBADEFINE
DEFINE
FILE DBA
CATEGORY/A9 = DECODE TYPE(M,MATCH,V,'VTOC ONLY',C,'CTLG ONLY',
  ELSE OTHER);
ATU/I6 = IF DO EQ 'DA' OR DO EQ 'IS' THEN TA ELSE TU;
LUX1/A2 = EDIT(LU,'99');
LUX1A/I4 = EDIT(LUX1);
LUX2/A3 = EDIT(LU,'$$999');
LUX2A/I4 = EDIT(LUX2);
LUYR/I6 = &YMD;
COST/F7.2 = TA * .25;
FREQUENCY/A12 = DECODE F(D,DAILY,B,BIWEEKLY,M,MONTHLY,Q,QUARTERLY,
W,WEEKLY,T,TABLES,' ',' ',
    S,'SEMI ANNUAL',A,ANNUAL,R,'AS REQUIRED', ELSE 'NOT STANDARD');
SCRATCH/A1 = IF TYPE EQ 'V' THEN 'X' ELSE
IF ((AGE GE 100) AND (F NE 'A' AND F NE 'S' AND F NE 'R')) THEN 'X'
ELSE ' ';
IEHPROGM/A80 = ' SCRATCH  '^'DSNAME='^DSN^',VOL=3330='^VS;
UNCATLG/A80  = ' UNCATLG  '^'DSNAME='^DSN;
FLAG/A1 = IF TYPE EQ 'V' THEN 'A' ELSE
IF ((TU EQ 0) AND (DSORG NE 'DA' AND  DSORG NE 'IS')) THEN 'B' ELSE
IF AGE GE 99 THEN 'C' ELSE ' ';
REASON/A10 = DECODE FLAG(A,'NOT CTLGED',B,'NO SPACE',C,'AGE',' ',' ');
DOUBT/I6 = IF FLAG NE ' ' THEN TA ELSE 0;
NAME4/A4 = EDIT(DSN,'9999');
NS/A1 = IF NAME4 EQ 'BDB.' OR NAME4 EQ 'SYS1' OR NAME4 EQ 'REST'
OR NAME4 EQ 'RAMI' THEN ' ' ELSE
 IF SID EQ ' ' AND
TSOID EQ ' ' THEN 'X' ELSE
IF FREQUENCY EQ 'NOT STANDARD' THEN 'X' ELSE '.';
ANALYST-NAME/A15 = DECODE AN(02,ARMSTRONG,03,ROMANO,04,SCHIAVONE,
05,BOYER,09,EVANS,10,FRIES,11,FARRALL,14,HEAD,
23,RIDGEWAY,24,DICAMILLO,72,NOVACO,93,MIDDLETON,98,PRC,99,UNASSIGNED,
ELSE UNASSIGNED);
TSOPREFIX/A2 = EDIT(TSOID,'99');
END
END CATALOGS
END OF DATA
```

On the negative side, we have experienced the pains while Mathematica was growing. We often felt that we were a Beta Test site. We made extensive use of the RAMIS "hotlines" feature. They were very responsive to our (and their) problems.

Some of these were brought up at a recent round table, much like cincom's KNOCKABOUT.

* Improved error diagnostics
* Consistent syntax throughout RAMIS
* Concurrent Usage of the same file
* Enhance database recovery
* Additional data structures (array, inversions)
* Design and performance aids
* Encryption
* Read only scan
* Multiple reports from a single pass of the database
* Modeling

The ensuing systems design was developed by Arthur Young and Company. The programming effort is by Electronic Data Systems. All edit and update programs are written in COBOL using the Procedural Language Interface. Reports are produced on-line and in batch using the non-procedural language. Cataloged procedures are available to prompt the non ADP user for most applications.

SYSTEM OVERVIEW

This section of the report is intended to provide a high-level view of BAS in the context of which the discussion of specific processing modules and concepts in succeeding volumes may be interpreted.

The system overview provides three high-level views of BAS:

- system modules - in which the modules of BAS are identified, and their interrelationships.
- functions and processes - in which the major program run units and processing cycles (daily, monthly, annually) are identified.
- data base organization - in which the major files of BAS are identified and their contents described.

The size and scope of BAS are such that only the combination of these views can generate an understanding of the organization and underlying concepts of BAS.

A. BAS SYSTEM MODULES

The majority of the material in this report is oriented toward individual modules of BAS. Therefore, the first overview of BAS will consider the modules of BAS, their relationship and purpose.

The BAS is a batch-processing system, enhanced by availability of data online using CRT terminals and the availability of a non-procedural language which enables non-programmers to write programs to interrogate BAS data. Most BAS data is stored in a central, integrated data base which will be available online. Exhibit I-1 lists the modules of the BAS. They are divided into five major groups, by major processing functions:

- Funds Processing
- Travel
- Property Accounting
- Miscellaneous Modules
- Interfaces.

1. Funds Processing

The funds processing modules are the heart of the BAS. Most of the programs and procedures to be implemented will be in support of Funds Processing. Funds Processing is divided into these modules:

- Administrative File Maintenance - which creates and maintains JON, 506 and 504 non-fiscal data in the BAS
- Major Funds Adjustments - processes the addition or withdrawal of 504, 506 or JON allocation funding
- Reprogramming - handles zero balance transfers of available funds between JONs under the same UPN
- Commitments - reserves funds at the JON level and reduces available 504, 506 and JON authority
- Obligations - records the fact that NASA is now legally obligated to the use of funds previously reserved

EXHIBIT I-1

BAS MODULES

1. Funds Processing

   . Administrative File Maintenance

   . Major Funds Adjustments

   . Reprogramming

   . Commitments

   . Obligations

   . Disbursements

   . Accruals

   . General Ledger

   . Chargebacks

   . Reimbursables

   . Mon-End Closing

   . Year-End Closing

   . Funds Availability

   . FACS

   . Miscellaneous Reports

   . Miscellaneous Queries

   . SACN

   . File Update

2. Travel Module

3. Property Accounting Module

   . Fiscal control of NASA-owned Capital Assets Under GSFC responsibility

   . Keeping of information for reporting to NASA HQ on capital assets under GSFC responsibility

4. System Function Modules

   . Table Maintenance

   . Batch Processing

   . Error Recycle

   . Document Data Base

   . Audit and Control Modules

5. Interfaces

   . Labor Interface

   . Supply Interface

   . Procurement Interface

   . Budget Interface

- Disbursements - handles payment from obligated funds when services or materials are received. The disbursement module will also schedule payments and produce an automated 1166

- Accruals - Accrual entries entered by Financial Analysts or generated automatically by the system will be processed for month-end closing, and reversed out the following month

- General Ledger - will handle posting of fiscal activity to the General Ledger. Most entries will be generated automatically by other funds processing Modules and posted by the General Ledger Module.

- Chargebacks - will handle special processing allocated with recording of cost against carrier accounts and the distribution of these costs to valid UPNs on an actual use or allocation basis, as appropriate.

- Reimbursables - will set up JONs for recording of Costs incurred in fulfilling reimbursable orders. Cash advances will be processed, and customer billings automatically generated. In addition, R&PM activities will be burdened according to applicable policies

- Month-end closing - will handle the production of special reports for GSFC, HQ and other agencies (i.e., SF224) at month-end, as well as generation of the current FACS reports and files for HQ use.

- Year-end closing - will handle internal and external requirements associated with the year-end closing

- Funds availability - checks for the availability of funding at the JON, 506, 504 and Reimbursable Order levels prior to acceptance of BAS fiscal transactions

- FACS - prepares GSFC internal reports as well as reports required by NASA HQ on a monthly basis

- Miscellaneous Reports - identifies reports produced by the system which are not identified with a particular processing module

- Miscellaneous Queries - identifies structures or pre-programmed queries which are available to aid the user in interrogating the BAS data base

- SACN - provides the capability to identify and track funding for computer systems by system number, and direct travel funding by organization

- File Update - defines the logic for updating of BAS files, particularly the JON file, for each transaction code.

2. Travel Module

The travel module will be used to provide comprehensive control of Fund Source 2 resources. Information will be recorded and tracked by UPN. Object Class and Organization Detail data will be maintained by traveller and individual trips. The travel module will also handle processing of certain funds which are related to Fund Source 1 (i.e., Change of Station) and Fund Source 3. Travel obligations and disbursements and funds availability checking will be handled by interfaces to the fiscal modules.

3. Property Accounting Module

The property accounting module has two objectives:

- Fiscal control of NASA-owned Capital Assets under GSFC responsibility

- Keeping of information for reporting to NASA HQ on capital assets under GSFC responsibility.

The module also will calculate depreciation using straight line methods based upon parameters (initial cost, useful life) supplied at the time an asset is recorded. The Property Accounting module will also provide fiscal transactions to update General Ledger accounts which relate to accounting for property.

4. System Function Modules

These modules support capabilities which are used by multiple functions and modules within BAS.

- Table Maintenance - the table maintenance module creates and maintains tables of values needed for data edit and validation. This module also creates and maintains the vendor file and the reimbursables customer file.

- Batch Processing - this module accepts and balances batches of input data received as input streams keyed by MSO or automatically prepared by other GSFC systems.

- Error Recycle - this module provides the ability to store transactions which are determined to be in error and to apply corrections against these transactions.

- Document Data Base - defines the organization and logic for the storage of document-level data supporting fiscal transactions, including the ability to store and access this data not only by JON, but by contract or purchase order number.

- Audit and Control Modules - the audit and control module is intended to provide for periodic self-audit of the data base and production of run-to-run control reports. In addition, certain General Ledger and fiscal balancing routines are provided.

Self audit will be accomplished by providing a means of reconciling processing done against fiscal files on a daily basis. To accomplish this, summary data will be maintained within the system. Transaction activity for each day will be added to the summary data generated as a result of previous day activity. The balances developed will be compared against actual system JON, 504, 506 and General Ledger totals. Moreover, certain General Ledger account and fiscal data totals will be compared and balanced.

A complete transaction log will be maintained to support fund and General Ledger balances, but also to enable reprocessing of transactions against a dump of the data base if problems are detected in the data base.

5. Interfaces

Some of the systems now in use at GSFC are not to be replaced by BAS. However, these modules interact heavily with BAS. Special interface modules will be developed to allow these systems to communicate with BAS and to access the BAS data base files.

- Labor Interface - the Labor interface will support coding of time card data and travel orders using the LACN concept, and also to post financial information into the BAS fiscal files. Time card data will be stored in the BAS document data base.

- Supply Interface - The Supply interface will support recording of fiscal data for supply carrier accounts.

- Procurement Interface - the Procurement interface can be used to generate Contract, Purchase Order and other data which will be needed by BAS. Small purchases data will be received from the AMIS system.

- Budget Interface - the Budget interface will be used to synchronize data between the Budget system and BAS, provide a means of exchanging data on Reprogrammings and other actions having budget impact, and to make data such as the cost and obligation plans available to BAS.

B. BAS FUNCTIONS AND PROCESSING

A second way of surveying the BAS is by considering the processing cycles which make up the BAS. The processing cycles may be broadly categorized as:

- Daily

- Monthly

- Annually

- As needed/ad hoc.

Exhibit I-2 shows the major processing cycles and identifies specific processing associated with each cycle.

1. Daily Processing

The daily processing cycle is subdivided into four major components:

- Input/Edit - all transactions, regardless of type, are

BAS Process Overview - Daily Processing

EDIT

- Merge Recycle and Memo Files with Edit Input
- Edit Transactions
- Send Errors to Recycle File and Generate Error List
- Create Daily Input Files and Controls

→

PROCESS

- Perform Specific Processing as Needed for
  - Administrative File Maintenance
  - Major Funds Adjustments
  - Disbursements
  - Obligations
  - Commitments
  - Accruals
  - Reimbursables
  - Other Fiscal Processing
  - Travel
  - Property Accounting

→

UPDATE

- Check Funds Availability for Fiscal Actions
- Recycle Transactions where Funds are not Available
- Update Fiscal Records
  - JON
  - 506
  - 504
- Update G/L
- Update Document Data
- Make Payments (1166)

→

REPORT

- Transaction Status
  - Errors
  - Accepted Transactions
  - File Changes
- Funding Status
  - 25002
  - Fund Violations
- 1166 Payment Schedule

BAS Process Overview - Monthly Processing

EDIT

- Edit Input Transactions
- Send Errors to Recycle File and Generate Error List
- Create Monthly Input Files and Controls

→

PROCESS

- Distribute Chargebacks
- Bill Reimbursables
- Process/Generate Accruals
- Perform Month-End Closing Processing

→

REPORT

- External Reports
  - FACS
  - SF224
  - MMR (partial automation)
  - Other NASA HQ Requirements
- GSFC Reports
  - Accruals
  - Trial Balances
  - Chargeback Status
  - Billing/Aged Receivables

BAS Processing Overview - Annual Processing

CLOSE OUT CURRENT YEAR

- GSFC Reporting
- NASA HQ Reporting
- Other NASA Site Reporting
- Other Agency Reporting

→

INITIALIZE FOR NEW FISCAL YEAR

- Set Up New Budget Data, Cost, Commitment & Obligation Plans
- Establish New Chargeback Percentages
- Reset Fiscal Accounts as Needed
- Set Up New JONs and Reference Data

processed through a "front-end" edit module. Inputs to the edit module are a batch transaction file, prepared in MSO by key-taping of input documents and correction forms, a memo transaction file which consists of "unofficial" transactions including those collected during the day, and an Error Recycle File which consists of transactions in error awaiting correction. The edit module will validate all transactions. Valid transactions will be collected in a transaction file and passed on to the process module. Transactions which are in error are added to the error recycle file, and simultaneously entered on an error report, along with a listing of the errors detected in the transaction. Errors which cycled through from the Recycle file and which were not corrected in this processing cycle are aged and flagged for special attention.

Processing - Transactions which have passed successfully through the edit are routed to appropriate processing modules. For example, travel transactions will be passed to the travel processing module. The modules which process these transactions correspond to a large degree to the modules identified in the previous section.

Funds Availability/Update - Before the fiscal data base can be updated and balances modified, fiscal records must be examined to determine whether funds are available to support the action requested. That is, if a request to commit $10,000 is being processed, we must first verify at the JON, 504, and 506 levels that $10,000 is available. If funds are not available, the transaction is written to the Recycle file. It will automatically be reentered into the system during the next daily run.

If funds are available, the fiscal data base is updated. That is:

- availability balances are adjusted as needed

- commitment, obligation, or disbursement actions are taken

- JON fields are updated

- General Ledger updating transactions are generated

• Reporting - Several generic report types are produced as a result of daily processing:

  - transaction lists, transaction error lists

  - status of funds by JON, UPN, action taken, organization, etc.

  - balance and control reports for JON and General Ledger files

  - reference file changes and alterations.

2. Monthly Processing

The monthly processing cycle is divided into three major components:

• Input/edit - is identical to daily cycle input/edit processing.

• Processing - Monthly processing involves several specialized activities which are required to be performed prior to month-end closing. These include:

- distribution of chargeback carrier accounts (FS9) to UPN JONs

- billing of Reimbursable Orders charges to customers

- input of manual accruals and generation of automatic accruals for scheduled disbursements

- processing of journal voucher adjusting entries to the General Ledger

- closing of the books and entry of reconciliation entries

- assembling of FACS input and update of the FACS data base

- preparation of FACS tapes and reports for HQ.

. <u>Reporting</u> - Month-end reporting involves a good deal of reporting external to GSFC, including FACS, and the SF224. Many budget-related reports also are produced at month-end, including the PMR, MMR, and Reprogramming Activity report. HQ oriented reports include the Financial Highlight Report, Preliminary Accrued Cost Report, and Financial Status of Programs.

3. <u>Annual Processing</u>

Annual processing involves the closing of the books for the current year, and the opening of the books for the next fiscal year.

. <u>Year-end closing</u> - is accomplished in accordance with FMM and involves the production of many special reports for NASA

HQ and other external sources, as well as GSFC internal reports.

- Beginning of year - involves the establishing of fiscal records for the new fiscal year. At this time, budget information is "rolled over" from the POP data in the budget files to establish new JON allocations.

4. Other Processing

The previous discussion of the major processing cycle, while not all-inclusive, gives an idea of the types of activity associated with each cycle. Certain other reporting requirements exist which are on other cycles -- quarterly, semi-annually, etc. However, the availability of information is not limited to that which is generated in the course of scheduled production. The use of data base technology and the availability of CRT terminals makes information available as needed to a user of the system. The ability to formulate tailored queries using the RAMIS non-procedural language makes ad hoc reporting requirements attainable without the use of programmers.

Some system reports may also be produced as needed, rather than on schedule. Reports which should be on demand include:

- ACN/JON Cross Reference
- Listings of System Tables
- Trial Balances and Detailed Analysis of Accounts
- JON or LACN Reference Data Printouts.

C. BAS DATA BASE

The third summary review of BAS is the BAS data base. The BAS processing is built around the use of a central, integrated data base which is available for normal processing and update as well as query by FMD and other personnel using terminals. Exhibit I-3 represents pictorally the components of the BAS data base and the interrelationships which exist between certain data sets. This section of the report briefly describes each file within the data base. The data base and its individual elements are described in detail in Section XII of this report.

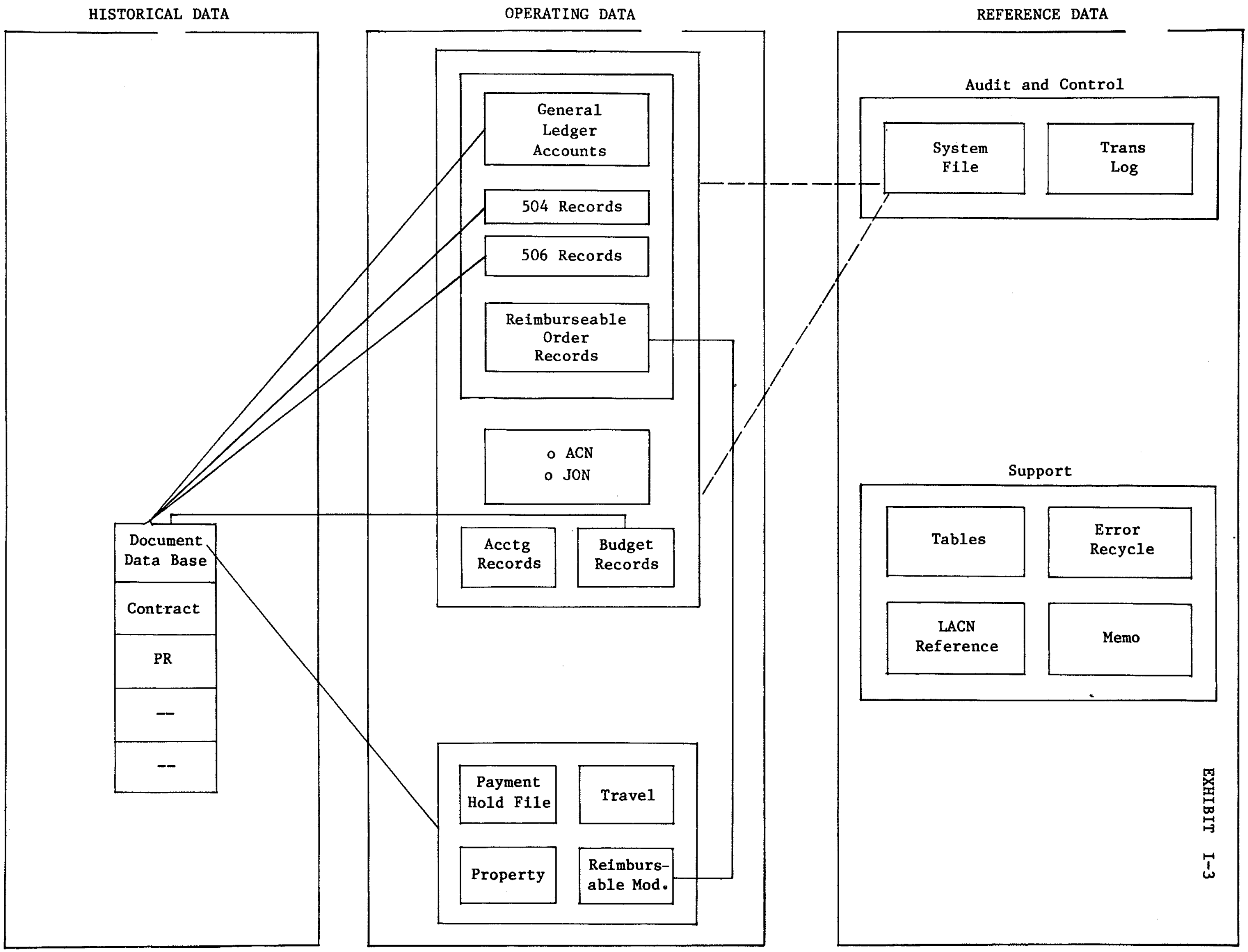
1. Fiscal Files

The fiscal files are the nucleus of BAS. In the fiscal files, information is maintained at the JON, 504 and 506 levels as to total funding and funds availability.

a. JON File

The JON file maintains detailed information as to funds availability and running totals of Commitments, Obligations, Disbursements and Accruals. In addition, each JON has an associated reference record, which contains the data elements which uniquely identify a JON:

- Organization
- UPN
- Function Code
- Method of Authorization
- Reimbursable Code


HISTORICAL DATA
OPERATING DATA
REFERENCE DATA
Document Data Base
Contract
PR
--
--
General Ledger Accounts
504 Records
506 Records
Reimburseable Order Records
o ACN
o JON
Acctg Records
Budget Records
Payment Hold File
Travel
Property
Reimburs-able Mod.
Audit and Control
System File
Trans Log
Support
Tables
Error Recycle
LACN Reference
Memo
EXHIBIT I-3

- Internal Use

and additional elements which supply additional DESCRIPTIVE INFORMATION ABOUT EACH JON, such as:

- Program Letter
- IMS Allocation Percentage
- Subauthorization Issuer Code
- Fund Source.

The JON data structures are described in the report Basic Accounting System: Proposed JON Coding Structure, issued in May 1979.

Each JON record has an associated Accounting Classification Number (ACN), which is used as a "shorthand" number to reference a specific JON and its associated data.

b. <u>506 File</u>

A 506 record is generated to maintain the status of Resource Warrant Authorities. Both the total authorization and the current funds available are maintained for audit/control purposes and funds availability checking. 506 records are present for each direct funding, reimbursable order or subauthorization received.

c. <u>504 File</u>

A 504 record is generated to maintain the status of Allotment Authorizations. Both the total authorization and the current funds available are maintained for audit/control purposes and funds availability checks. 504 records are present for each appropriation and include both direct and reimbursable funding.

d. General Ledger

The BAS data base will incorporate the GSFC chart of accounts and maintain current balances for each account. The General Ledger file will be supported by BAS detail transaction data. General Ledger transactions will not usually be coded by hand, since each fiscal transaction will automatically generate the necessary General Ledger entries -- a major benefit, since some transactions could involve up to six ledger entries. A journal voucher direct entry capability will be available for adjustment entries.

e. FACS File

A summary level file containing inception-to-date data will be available to meet FACS reporting requirements. The existing FACS file will be converted to accommodate the new BAS coding requirements and subsequently will be maintained by BAS.

f. LACN File

A Labor Accounting Classification Number (LACN) file will be maintained which will contain reference information for LACNs. This file replaces the LCR and 02 tables of the current GSFC fiscal system and will be used to validate Labor and Travel input data, and provide additional functions to the Labor and Manpower systems, such as convertion of WBS UPNs to AWCs UPNs and prorations of labor charges.

g. Reimbursable Order File

The Reimbursable Order file will be used as a reference file to support unique data requirements of Reimbursable processing. The Reimbursable Order file is unique in that it operates as a "high level" record coordinating several JON records, which can include multiple fund sources.

h. Reimbursables Customer Billing File

This file supports the processing of cash advances, preparation of bills, and the recording of payments against Reimbursable Orders.

i. Payables File

To support the cash management function of the Disbursements Module, payments which are scheduled for future issuance of a check will be stored by release data on the Payables file. On the release date, the Payables file records for that date will be used to produce an automated 1166 and then purged from the Payables file.

2. System Files

The system files support processing requirements of many applications.

a. Tables File

Each table required by BAS, i.e., tables of

- Object Classes

* Program Letters
* Vendor Information (for Disbursements)
* A Customer Information (for Reimbursables)
* Function Codes

will be maintained and accessible through the BAS data base. Generally, each table will contain not only valid codes for a particular data element, but also a description of the meaning of the code. Where possible, to facilitate ease of use by people, the description will be used on output reports rather than (or sometimes in addition to) the coded value.

Thus, for example, the Object Class value 2142 would appear in the Object Class file. The Travel Module would make use of this value to edit Object Class data, and could also use the description to replace the coded value 2142 with the more descriptive Rental of Passenger--Carrying Vehicles - Government in any output reports.

b. Document File

The Document file will provide a machine-readable version of all document transactions input to BAS. The document file will be used for many purposes, including historical reporting, development of ad hoc reports at a level of detail not available directly elsewhere, editing and processng of Obligation or Disbursement transaction, and as detail to support the numbers in the fiscal files and the General Ledger.

c. Error Recycle File

All data entering the BAS is passed through a comprehensive edit program. Transactions which fail the edit are retained by the system in the Error Recycle file. At the time the error transaction enters the Recycle file, a report of error transacting is generated which will explain the nature of the error(s) in the transaction. Transactions are removed from the recycle file only by the reentry of corrections to the fields in error on the recycled transaction, or by an explicit delete request. An uncorrected error will cycle through the daily edit process until positive action is taken.

d. Memo File

A potential future expansion of the BAS will permit certain transactions to be entered into BAS as they are generated during the day using video terminals. These transactions would be stored on the "memo" file, and retained for possible processing that evening during the manual batch runs, or at a later date.

3. System Audit and Control Files

These files are used to provide audit trails for all fiscal transactions, and also could be used to assist in the reconstruction of BAS files in the event of a loss of BAS data base integrity.

a. <u>System File</u>

The system file contains "rollups" or summary totals for the fiscal files. Daily activity against both the General Ledger and JON files can be compared and balanced against the system file summary data. Moreover, cross-checks between the JON and General Ledger are also provided. The system file is unique in that it is written twice to disk, on different disk file packs, to provide an adidtional level of security and control. The system file is also not under the control of RAMIS, so that failure of the data base software will not impact the integrity of the System File.

b. <u>Transaction Log</u>

All fiscal transactions are time-stamped and recorded in order as received to the log file. In the event of system failure, the log tape could be used in conjunction with a data base dump/restore file (see below) in order to rebuild the BAS data base from a prior point in time. Transaction log files will be produced in duplicate, with one copy to be retained offsite.

c. <u>Data Base Dump/Restore File</u>

Periodically, the contents of the data base will be copied onto magnetic tape. This tape will be used to reconstruct the data base according to the information on the tape in the event of system failure.

4. <u>Travel Data Base</u>

The travel data base is organized to facilitate information gathering and reporting in the dimensions of 506 Authority and organizational allocations, benefitting project, and object class. The information itself is tied to the person making the trip, the trip itself, and the fiscal information about thet trip. Key information about documents having to do with a specific trip (car rental agreements, advances, etc.) will be stored with the trip, and cross-reference indices will be provided to the paper document files for further back-up.

In addition to the ability to track and report information about travel from the time the orders are signed until all advances and invoices have been cleared, the data base has been designed in such a way as to provide information about possible trips prior to the obligation stage. This "memo" facility will provide an additional degree of control over travel prior to actual obligation of funds.

5. <u>Property File</u>

The property data includes two sets of records:

- Fixed Asset Master Status records for each unit of property tracked by BAS, and a Current Year Transaction History to support the property master records. The Master Status records include those data elements required to show the current status of a tagged item.

- Master Status data elements are updated through Fixed Asset transactions.

All Fixed Asset transactions which affected a given property record during the current year have generated a record stored in the Current Year Transaction History. These transaction history records are available to facilitate queries and provide an audit trail.

Other property related data on the Basic Accounting System Data Base can be found in the document and table files.

RIMS - RESOURCE INFORMATION MANAGEMENT SYSTEM

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

J. Symes
Lyndon B. Johnson
Space Center

RIMS TOPICS

PURPOSE

CHARACTERISTICS

FUNCTIONAL CAPABILITIES

USER INTERFACE

USER COMMUNITY

RIMS PURPOSE

PROVIDE A SIMPLE INTERACTIVE DMS TOOL TO ALLOW USERS TO BUILD, MODIFY, AND MAINTAIN DATA MANAGEMENT APPLICATIONS

SUPPORT FLAT (SINGLE KEY) FILES

USE FULL SCREEN I/O

MINIMIZE PROGRAMMER SUPPORT REQUIRED TO DEVELOP/MAINTAIN SMALL DATA BASE APPLICATIONS

PROVIDE A DMS TOOL TO ASSIST IN BRINGING THE UNITED INFORMATION SERVICES (UIS) BUDGET SYSTEM WORK INHOUSE

RIMS CHARACTERISTICS

HARDWARE

- UNIVAC 1108, 1100/81
- TERMINALS ARE U200 COMPATIBLE
- REMOTE PRINTERS

SOFTWARE

- MULTI-USER
- TEMPLATE DRIVEN (FULL SCREEN I/O)
- "INDEX SEQUENTIAL" TYPE FILE HANDLER
- DYNAMIC USE OF SORT AND OTHER MEMORY MANAGEMENT
- AUTOMATED CONTINGENCY PROCESSING OF ERRORS
- TIME OUT AVOIDANCE (30 MINUYTE GRACEFUL TERMINATION)
- AUTOMATED LOGGING OF USAGE DATA
- WRITTEN IN FORTRAN 5 and ASSEMBLER

THE RIMS DEVELOPMENT AND SUBSEQUENT PRODUCTION HARDWARE IS THE UNIVAC 1100 USING EXECUTIVE OPERATING SYSTEM LEVEL 36. THE TERMINALS USED AT JSC ARE UNISCOPE 200 COMPATIBLE TERMINALS MANUFACTURED BY MEGADATA CORPORATION. THE TERMINALS HAVE ATTACHED PRINTERS. IN ADDITION TO THE UNIVAC 1100 HIGH SPEED PRINTERS, SOME USERS HAVE A 600 LINE PER MINUTE PRINTER IN THEIR WORK AREA.

THE RIMS SOFTWARE IS A MULTI-USER SYSTEM UTILIZING AN "IN-HOUSE" DEVELOPED TERMINAL HANDLER FOR FULL SCREEN I/O, AND AN "INDEX SEQUENTIAL" TYPE FILE HANDLER. BECAUSE OF PROGRAM SIZE CONSTRAINTS IMPOSED BY THE CURRENT PRODUCTION HARDWARE (UNIVAC 1108), THE SORT CAPABILITY AND FILE HANDLER UTILIZE MEMORY MANAGEMENT TECHNIQUES TO EXPAND AND CONTRACT THE PROGRAM SIZE.

OTHER FEATURES OF RIMS ARE AUTOMATED CONTINGENCY PROCESSING OF ERRORS, TIME OUT AVOIDANCE, AND AUTOMATED LOGGING OF USAGE DATA.

RIMS Functional Capabilities

USER DEFINED/MAINTAINED ENTITIES

- DATA BASE DEFINITION
- SINGLE RECORD UPDATE/RETRIEVE TEMPLATES
- REPORT FORMATS
- DATA SELECTION CRITERIA
- COMPUTATIONAL ELEMENTS
- PROCEDURES
- ACCESS SECURITY

SYSTEM FUNCTIONS

- REPORT GENERATION
- QUERY
- PRINTED OUTPUT ROUTING
- MULTI FILE
  - APPEND
  - MERGE (IN DEVELOPMENT)

SYSTEM INFORMATION

- LISTS OF CURRENT USER ENTITIES
- ABBREVIATED USER DOCUMENTATION
- DATA FILE STATISTICS

THE RIMS FUNCTIONAL CAPABILITIES AVAILABLE TO THE USER FALL INTO THREE MAIN CATEGORIES:

(1) USER DEFINED AND MAINTAINED ENTITIES
(2) RIMS SYSTEM FUNCTIONS
(3) RIMS SYSTEM INFORMATION

(1) USER DEFINED AND MAINTAINED ENTITIES

SINCE RIMS IS ORIENTED TOWARD THE NON-PROGRAMMER END USER, HE/SHE MUST BE ABLE TO DEFINE AND CHANGE HIS/HER DATA RECORD DEFINITION. THE RECORD DEFINITION FUNCTION REQUIRES THE USER TO SPECIFY THE ELEMENT NAME (1-12 CHARACTERS), WHETHER OR NOT THE ELEMENT IS PART OF THE KEY, ELEMENT TYPE (ALPHA, ALPHANUMERIC, INTEGER, DECIMAL, DATE, ETC.) AND THE NUMBER OF CHARACTERS IN EACH ELEMENT.

THE TEMPLATES USED TO UPDATE AND RETRIEVE SINGLE RECORDS ARE ALSO USER DEFINED. THESE TEMPLATES CONTAIN THE ELEMENT NAMES (AS DEFINED IN THE DATA BASE DEFINITION) AND THE DESIRED OUTPUT FORMAT.

TO SATISFY THE REPORTING REQUIREMENTS, THE USER CAN CREATE HIS/HER INDIVIDUAL REPORT FORMATS. THE GENERAL FORMAT IS FROM 0-5 LINES (LINE IS 1 TO 132 CHARACTERS WIDE) IN THE TITLE SECTION, 1 TO 10 LINES IN THE HEADING SECTION, AND FROM 1 TO 5 LINES IN THE DATA FORMAT SECTION.

TO FACILITATE DATA SELECTION, THE ABILITY TO SAVE REPETITIVE SELECTION PARAMETERS (DATA ELEMENT NAMES, VALUES, SORT SPECIFICATION, PAGE BREAK REQUESTS) HAS BEEN INCLUDED. THESE SELECTION CRITERIA ARE USED TO GENERATE REPORTS, FOR GLOBAL MODIFICATIONS TO DATA BASES, AND TO APPEND VARIOUS FILES TOGETHER.

THE USER HAS THE ABILITY TO DEFINE AND STORE ARITHMETIC RELATIONSHIPS BETWEEN DATA ELEMENTS AND CONSTANTS. THESE RELATIONSHIPS CAN BE USED AS SELECTION CRITERIA AND AS OUTPUT FIELDS IN RETRIEVAL AND REPORTING.

THE REQUIREMENT TO STRING TOGETHER VARIOUS FUNCTIONS SUCH AS REPORT GENERATION FOLLOWED BY ROUTING THE OUTPUT TO A REMOTE PRINTER IS SUPPORTED BY THE RIMS PROCEDURES FUNCTION. THIS ALLOWS REPETITIVE MULTISTEP PROCESSING TO BE SAVED, RECALLED, AND PERFORMED WITH A MINIMUM OF USER INTERACTION.

THE RIMS ACCESS SECURITY ADDRESSES THE ACCESS REQUIREMENTS FROM THE SYSTEM LEVEL, THROUGH ENTITY DEFINITIONS, DOWN TO THE DATA ELEMENT RETRIEVAL AND UPDATE CAPABILITIES.

(2) SYSTEMS FUNCTIONS

THE RIMS SYSTEM FUNCTIONS INCLUDE REPORT GENERATION, AD HOC INQUIRIES, ROUTING PRINTED OUTPUT, AND MULTI FILE APPEND AND MERGE.

THE REPORT GENERATION AND INQUIRY FUNCTIONS ALLOW THE USER TO REQUEST A SPECIFIC REPORT FORMAT (INQUIRIES GENERATE THEIR OWN FORMAT) AND RESTRICT THE AMOUNT OF DATA VIA SELECTION CRITERIA SPECIFICATIONS INCLUDING COMPLEX BOOLEAN RELATIONSHIPS. IN ADDITION, ANY SORT REQUESTS AND PAGE BREAK REQUESTS CAN BE SPECIFIED FOR THE CURRENT GENERATION.

THE RIMS PROVIDES FOR VARIOUS HARDCOPY DEVICE SELECTION. THESE ARE TERMINAL PRINTERS, HIGH SPEED PRINTERS, REMOTE PRINTERS, MICROFICHE, AND LIMITED XEROX PROCESSING. THE DEVICE SELECTION IS UNDER USER CONTROL.

THE RIMS PROVIDES A MECHANISM WHEREBY DATA FILES CAN BE APPENDED OR MERGED. THE APPEND OPERATES ON FILES HAVING THE SAME DATA BASE DEFINITION AND THE MERGE PROCESS RESULTS IN A THIRD FILE WITH ITS OWN DEFINITION.

(3) SYSTEM INFORMATION

LIMITED STATUS INFORMATION IS AVAILABLE TO THE USER. EACH DEFINITION FUNCTION WILL PROVIDE A LIST OF CURRENT ENTITIES. ALSO AVAILABLE IS ABBREVIATED USER DOCUMENTATION ACCESSABLE ONLINE. THE RIMS PROVIDES SOME DATA FILE STATISTICS SUCH AS NUMBER OF RECORDS, AND FILE HANDLER INFORMATION OF USE IN DBA FUNCTIONS.

RIMS USER INTERFACE

SAMPLE TEMPLATES (WITHOUT DATA)

SIGNON
USER TEMPLATE DEFINITION (2)
REPORT GENERATION

SAMPLE TEMPLATES (WITH DATA)

SYSTEM ENTRY - SIGNON

USER DEFINED AND MAINTAINED ENTITIES

DATA BASE DEFINITION
USER TEMPLATE DEFINITION
REPORT FORMAT DEFINITION (3)
SELECTION CRITERIA DEFINITION
COMPUTATIONAL ELEMENT DEFINITION

SYSTEM FUNCTIONS

REPORT GENERATION
PRINTED OUTPUT ROUTING
FILE APPEND

SYSTEM INFORMATION

LIST USER ENTITIES AND SYSTEM DOCUMENTATION
DATA BASE STATISTICS

RIMS USER INTERFACE

SAMPLE TEMPLATES

(WITHOUT DATA)

RIMS USER INTERFACE - SIGNON

```
______          rims version (6h) signon                 signon

             rims domain ______
            rims user id ______
           rims password ______

initial data-set base (optional) - domain/file-definition/file to be
                                   accessed initially
                    file ______
         file definition ______
                  domain ______ (enter only if different than rims domain)

                     666666      hh        hh
                   66      66    hh        hh
                  66             hh        hh
                  66 666666      hhhhhhhhhhh
                  666     66     hh        hh
                  66        66   hh        hh
                   66      66    hh        hh
                     666666      hh        hh

COMPLETE 05/21/82 08:28:24, 0.6800 SECS, 0.0328 SUPS
```

RIMS USER INTERFACE - USER TEMPLATE DEFINITION (1)

_____ A/C ___ FILE _____ FILE-DEFN _____ DOMAIN _____ SELECT _____ NEWTEM

RESERVED
PROCESSING COMPLETE
ENTER NEW TEMPLATE

RIMS USER INTERFACE - USER TEMPLATE DEFINITION (2)

```
_____ a/c ___ file _____ file-defn _____ domain _____ select _____ drm$st

                    -- key --

      date: ______     hrs: __    min: __    sec:__

                   --- data ---

      ltm: ____.__       wtm: ____.__       xacnt: ___,___

                   --- calc's ---

 avs-load: ____.__  avs-work: ____.__  avs-resp: ____.__  resp-tm: _____.__
                                        udt: __/__/__  __:__:__
```

COMPLETE 05/21/82 08:26:22, 0.1910 SECS, 0.1172 SUPS

RIMS USER INTERFACE - REPORT GENERATION

```
______       a/c: ___     sel-id: ______     temp/perm: _                report
                            file: ______     defn: ______     domain: ______
 report generation:  report id: ______     summary only: _
     item name        OP  value                  mask      sort      breaks    totals
 a.  ___________      __  __________________     __        _/_       _/_       _
 b.  ___________      __  __________________     __        _/_       _/_       _
 c.  ___________      __  __________________     __        _/_       _/_       _
 d.  ___________      __  __________________     __        _/_       _/_       _
 e.  ___________      __  __________________     __        _/_       _/_       _
 f.  ___________      __  __________________     __        _/_       _/_       _
 g.  ___________      __  __________________     __        _/_       _/_       _
 h.  ___________      __  __________________     __        _/_       _/_       _
 i.  ___________      __  __________________     __        _/_       _/_       _
 j.  ___________      __  __________________     __        _/_       _/_       _
 k.  ___________      __  __________________     __        _/_       _/_       _
 l.  ___________      __  __________________     __        _/_       _/_       _
 m.  ___________      __  __________________     __        _/_       _/_       _
conditional relationships:   and = '&', or = '+', not = '-', exclusive or = '#'
________________________________________________________________________________
```

COMPLETE 05/21/82 08:25:30, 1.1830 SECS, 0.0588 SUPS

<u>RIMS USER INTERFACE</u>

SAMPLE TEMPLATES

(WITH DATA)

RIMS USER INTERFACE

SYSTEM ENTRY

```
                 rims version (6) signon                          signon

              rims domain FSSNDN
             rims user id FSSNDN
            rims password FSSNDN

initial data-set base (optional - domain/file-definition/file to be
                                  accessed initially)

                     file ______
          file definition ______
                   domain ______  (enter only if different than rims domain)

                           666666
                         66       66
                        66
                        66 666666
                        666       66
                        66         66
                         66       66
                           666666

COMPLETE 03/15/82 08:21:43, 0.6950 SECS, 0.1216 SUPS
```

THIS TEMPLATE IS USED TO SIGN ONTO:

1. A SPECIFIED DOMAIN WHEN NO FILE AND FILE DEFINITION ARE GIVEN.

2. A SPECIFIED APPLICATION WHEN A FILE AND A FILE DEFINITION ARE GIVEN.

RIMS USER INTERFACE

USER DEFINED AND MAINTAINED ENTITIES

RIMS USER INTERFACE - DATA BASE DEFINITION

data-item definition

data-item name	#	key	type/sign	width	data-item name	#	key	type/sign	width
LAST-NAME	10	Y	A /	20	FIRST-NAME	20	Y	A /	15
MID-INITIAL	30	Y	/	1	UPDATE-DT	40		U /	12
AREA CODE	50		N /	3	PHONE-#	60		N /	7
BIRTHDAY	70		D /	6	STREET	80		X /	20
APT-#	90		X /	5	STATE	100		A /	2
CITY	110		A /	20	ZIP-CODE	120		N /	5
#-KIDS	130		N /	2	SALARY	140		D2/	8
COMMENTS			/	30				/	

COMPLETE 03/15/82 08:57:31, 6.6710 SECS, 0.8598 SUPS
ALL EXISTING DATA-ITEMS HAVE BEEN DISPLAYED - TRANSMIT WHEN READY TO GO ON

THIS TEMPLATE IS USED TO DEFINE THE CHARACTERISTICS OR ALL DATA ITEMS INCLUDED IN EACH RECORD OF THE DATA FILE.

RIMS USER INTERFACE - USER TEMPLATE DEFINITION (1)

______ A/C ___ FILE _____ FILE-DEFN _____ DOMAIN _____ SELECT _____ TESTR6

RESERVED
PROCESSING COMPLETE
ENTER NEW TEMPLATE

THIS TEMPLATE IS USED TO DESCRIBE THE INPUT DATA LAYOUT FOR THE TEMPLATE (TESTR6) REFERENCED IN EITHER AN "INS" OR A MOD.

TEMPLATE CHARACTERISTICS

- ALL DATA IS FREE FORM.
- ALL KEY ELEMENTS MUST BE ENTERED.
- ELEMENT NAMES AND FIELD SIZES MUST AGREE WITH DBDEF.
- UNDERSCORES (_____) ARE USED TO DISPLAY FIELD SIZES.

RIMS USER INTERFACE - USER TEMPLATE DEFINITION (2)

```
______ a/c ___ file ______ file-defn ______ domain ______ select ______ testr6

first-name: _______________  mid-initial: _  last-name: ____________________

area-code: ___   phone-#: ___-____

address:--                                    personal info --

street: ____________________                  birthday: __/__/__   #-kids: __

city:     ____________________  apt-#: _____  salary: ___,___.__

              state: __   zip-code: _____

comments: _______________
          _______________
                                              update-dt: __/__/__   __:__:__
```

COMPLETE 03/15/82 09:03:21, 3.2570 SECS, 0.0408 SUPS
TRANSMIT=ACCEPT, PF3=START-OVER, PF4=CANCEL (AS IF CURRENT OPERATION NOT BEGUN)

A COMPLETED FORM OF THE PREVIOUS TEMPLATE WITHOUT INPUT DATA.

RIMS USER INTERFACE - REPORT FORMAT DEFINITION (1)

```
______    a/c: INS    report-id: TESTR6    defn: ______    domain: ______    rptdef

          a/c: ins - inserts (creates) a new report format
               mod - modify title (default) or format line # _
               ret - retrieve display (default) or title _ or format line # _
               del - deletes report
               dup - duplicate report format to new report-id: ______

                          report title input area
                  (title will be centered at generation time)
OUR FIRST______________________________________________________________
_____________________________________________
REPORT_________________________________________________________________
_____________________________________________
_______________________________________________________________________
_____________________________________________
_______________________________________________________________________
_____________________________________________
_______________________________________________________________________
_____________________________________________

COMPLETE 03/15/82 09:32:21, 0.1400 SECS, 0.1054 SUPS
```

TEMPLATE IS USED TO IDENTIFY THE NAME OF A REPORT (REPORT-ID), THE REQUIRED ACTION (A/C) AND THE REPORT TITLE. IT IS THE FIRST TEMPLATE USED DURING THE COURSE OF DEFINING (INS) A NEW REPORT.

IT CAN ALSO BE USED TO MODIFY (MOD) OR RETRIEVE (RET) A SPECIFIED PART OF A REPORT, OR DELETE (DEL) OR DUPLICATE (DUP) AN ENTIRE REPORT.

PAGES C-19 THRU C-30 DEMONSTRATE THE PARTS OF THE REPORT BEING DEFINED, A PAGED OUTPUT OF THE REPORT LAYOUT, A DATA SELECTION FOR THE REPORT, REPORT GENERATION USING THAT SELECTION, AND FINALLY A PAGED OUTPUT AND A PRINTED COPY OF THE ACTUAL REPORT.

RIMS USER INTERFACE - REPORT DEFINITION (2)

```
______   line a/c: INS   report-id: TESTR6   line #: 1   (a/c = ret,ins,mod,del,end)

   item name     tot     item name     tot     item name     tot     item name     tot
a. LAST-NAME___   _   b. FIRST-NAME__   _   c. AREA-CODE___   _   d. PHONE-#______   _
e. ____________   _   f. ____________   _   g. ____________   _   h. ____________   _
i. ____________   _   j. ____________   _   k. ____________   _   l. ____________   _
m. ____________   _   n. ____________   _   o. ____________   _   p. ____________   _
q. ____________   _   r. ____________   _   s. ____________   _   t. ____________   _
u. ____________   _   v. ____________   _   w. ____________   _   x. ____________   _
y. ____________   _   z. ____________   _

heading line(s):   2         3         4         5         6         7         8
_____LAST-NAME_________FIRST-NAME_____AREA-CODE PHONE-#_________________________
_____________________________________________________
________________________________________________________________________________
_____________________________________________________

format line:       2         3         4         5         6         7         8
AAAAAAAAAAAAAAAAAAAA, BBBBBBBBBBBBBBB____(CCC)_DDD-DDDD_________________________
_____________________________________________________

COMPLETE 03/15/82 09:33:36, 3.0750 SECS, 0.2670 SUPS
U315: TITLE INSERTED, CONTINUE
```

THIS IS THE SECOND OF THE TWO TEMPLATES REQUIRED TO COMPLETELY SPECIFY A REPORT FORMAT. IT IDENTIFIES:

- LINE ACTION CODE.
- REPORT IDENTIFICATION.
- THE LINE OF THE REPORT BEING ACTED UPON BY THE LINE A/C.
- DATA ITEM APPEARING ON THE LINE.
- HEADINGS FOR THE LINE.
- FORMAT OF DATA FOR THE LINE.

THIS TEMPLATE IS USED REPEATEDLY UNTIL ALL LINES OF THE REPORT HAVE BEEN DEFINED. LINE A/C "END" IS USED TO SIGNIFY THAT THE REPORT IS COMPLETE. THIS IS LINE 1 OF THE REPORT IDENTIFIED AS TESTR6.

```
______          ***  paged-output display template ***                  pages

                               OUR FIRST
                                REPORT
     LAST-NAME                 FIRST-NAME     AREA-CODE PHONE-#
#-KIDS       SALARY                     COMMENTS
AAAAAAAAAAAAAAAAAAAA, BBBBBBBBBBBBBBB     (CCC) DDD-DDDD
   AA  $BB,BBB,BBB.BB     CCCCCCCCCCCCCCCCCCCCC




COMPLETE 03/15/82 09:50:15, 5.4650 SECS, 1.5208 SUPS   PAGE #1 OF 1
INS REPORT DEFINITION COMPLETE
```

THIS TEMPLATE IS USED TO LOOK AT THE MOST RECENTLY GENERATED OUTPUT IN THE PAGE FILE.

THIS TEMPLATE AND CONTENTS ARE AUTOMATICALLY DISPLAYED (PAGE 1) UPON COMPLETION OF A REPORT FORMAT. (NOTE: TITLE AND TWO FORMAT LINES.)

```
______        a/c: INS       sel-id:  TESTR6     temp/perm: P                   seldef
                               file:  TESTR6     defn: ______      domain: ______
    report generation:  report id:  TESTR6    summary only: _
       item name      or   value                     mask     sort     breaks     totals
    a. LAST-NAME____  __   ___________________        _       1/_       _/_         _
    b. SALARY_______  GT   25000.00___________        _       2/D       _/_         _
    c. SALARY_______  LT   200000.00__________        _       _/_       _/_         _
    d. _____________  __   ___________________        _       _/_       _/_         _
    e. _____________  __   ___________________        _       _/_       _/_         _
    f. _____________  __   ___________________        _       _/_       _/_         _
    g. _____________  __   ___________________        _       _/_       _/_         _
    h. _____________  __   ___________________        _       _/_       _/_         _
    i. _____________  __   ___________________        _       _/_       _/_         _
    j. _____________  __   ___________________        _       _/_       _/_         _
    k. _____________  __   ___________________        _       _/_       _/_         _
    l. _____________  __   ___________________        _       _/_       _/_         _
    m. _____________  __   ___________________        _       _/_       _/_         _

conditional relationships:    and = '&', or = '+', not = '-', exclusive or = '#'
________________________________________________________________________________

COMPLETE 03/15/82 09:58:01, 6.6230 SECS, 0.8448 SUPS        PAGE 1 OF 1
SELECT DEFINITION COMPLETE
```

THIS TEMPLATE IS USED TO PREDEFINE (CAN) A DATA SELECTION FOR A REPORT (REPORT-ID MUST BE GIVEN) OR FOR APPENDING A FILE.

FIELD SPECIFICATIONS

- o A/C...........ACTIONS CODES
  - INS....DEFINES A DATA SELECTION.
  - RET....RETRIEVE A PREDEFINED DATA SELECTION.
  - MOD....MODIFY A PREDEFINED DATA SELECTION.
  - DEL....DELETE A PREDEFINED DATA SELECTION.
- o SEL-ID........NAME OF DATA SELECTION BEING ACTED UPON.
- o FILE, DEFN....LOCATION OF SEL-ID AND DOMAIN
- o REPORT-ID.....IDENTIFIES REPORT TO WHICH DATA SELECTION APPLIES.
- o DATA ITEMS....NAMES, OPERATORS, VALUES, MASK SORT, BREAK AND TOTAL SPECIFICATIONS.

RIMS USER INTERFACE - COMPUTATIONAL ELEMENT DEFINITION

```
______     a/c:INS      defn: TESTR6       domain: ______               cmpdef

        comp-id: MNTH-SALARY_     type: c2     sign: _     width: 10

   item name             item name              item name              item name
a. SALARY______       b. ____________        c. ____________        d. ____________
e. ____________       f. ____________        g. ____________        h. ____________
i. ____________       j. ____________        k. ____________        l. ____________

valid symbols:  a-z, add = '+'m subtract = '-', divide = '/', multiply = '*',+ '
                exponent = '**', absolute value = '[...]'

computational expression:
A/12______________________________________________________________________________

COMPLETE 03/15/82 10:32:02, 1.4460 SECS, 0.2864 SUPS     PAGE 1 OF 1
COMP DEFINITION COMPLETE
```

THIS TEMPLATE IS USED TO DEFINE A COMPUTATIONAL DATA ITEM. CURRENTLY, DATA ITEMS CAN ONLY BE USED IN OUTPUT REPORTS. THE ELEMENT (COMP-ID), THE NUMBER OF DECIMAL PLACES IF REQUIRED (TYPE), THE SIGN AND FIELD WIDTH MUST BE GIVEN.

THE DATA ITEMS (FROM DBDEF) TO APPEAR IN THE COMPUTATIONAL EXPRESSION MUST BE LISTED. THE COMPUTATIONAL EXPRESSION USING THE "ALPHA" CHARACTERS ADJACENT TO THE ITEM NAMES AND VALID SYMBOLS MUST BE FORMULATED.

IN THE ABOVE EXAMPLE, SALARY IS USED AS AN ANNUAL AMOUNT: THUS, A/12 (SALARY ÷ 12) PROVIDES A MONTHLY SALARY (MNTH-SALARY) FOR EVERY SALARY IN THE DATA FILE TESTR6.

PAGES C-43 THRU C-46 DEMONSTRATE THE FORMULATION OF A COMPUTATIONAL ELEMENT AND ITS USE IN AN OUTPUT REPORT.

RIMS USER INTERFACE

SYSTEM FUNCTIONS

RIMS USER INTERFACE - REPORT GENERATION

```
______        a/c: RET       sel-id:  TESTR6     temp/perm: P                  report
                               file:  TESTR6     defn: ______      domain: ______
   report generation:  report id:  TESTR6   summary only: _
      item name       or   value                     mask    sort     breaks    totals
   a. LAST-NAME___                                    _       1/_      _/_        _
   b. SALARY______    GT   25000.00__________         _       2/D      _/_        _
   c. SALARY______    LT   200000.00_________         _       _/_      _/_        _
   d. ____________    __   __________________         _       _/_      _/_        _
   e. ____________    __   __________________         _       _/_      _/_        _
   f. ____________    __   __________________         _       _/_      _/_        _
   g. ____________    __   __________________         _       _/_      _/_        _
   h. ____________    __   __________________         _       _/_      _/_        _
   i. ____________    __   __________________         _       _/_      _/_        _
   j. ____________    __   __________________         _       _/_      _/_        _
   k. ____________    __   __________________         _       _/_      _/_        _
   l. ____________    __   __________________         _       _/_      _/_        _
   m. ____________    __   __________________         _       _/_      _/_        _
conditional relationships:    and = '&', or = '+', not = '-', exclusive or = '#'
```

COMPLETE 03/15/82 13:45:33, 1.6250 SECS, 0.1458 SUPS

THIS TEMPLATE IS USED TO GENERATE (RPT) A REPORT OR RETRIEVE (RET) A PREDEFINED REPORT DATA SELECTION.

- A/C...........ACTIONS CODE (RET OR RPT)
- SEL-ID........NAME OF PREDEFINED DATA SELECTION.
- OPTIONAL FIELDS
- FILE, DEFN, AND DOMAIN....LOCATION OF REPORT BEING GENERATED.
- REPORT-ID.....NAME OF REPORT BEING GENERATED IF "SEL-ID" NOT USED.
- DATA ITEMS....NAMES, OPERATORS, VALUES, MASK, SORT, BREAKS, AND TOTAL SPECIFICATIONS ON REPORT BEING GENERATED IF "SEL-ID" NOT USED.

RIMS USER INTERFACE - PRINTED OUTPUT ROUTING

print request template

print devide:

```
 Y terminal printer         # of copies: 1  * id: _____   # of copies: ___
__ building 12 printers                     * id: _____   # of copies: ___
__ remote printer   (printer id: ______)    * id: _____   # of copies: ___
__ microfiche     24X:__   42x: __   48x:__ * id: _____   # of copies: ___
__ xerox       form:______   duplex: __
```

print controls:

```
___ line spacing
___ number of lines per page (standard is 66)
___ starting page number (default = 1)
___ ending page number (default = last page)
___ suppress print of selection criteria
```

* id = a 6 character output identifier of the form nnnxxx where nnn is a box number, and xxx are alpha characters.

COMPLETE 03/15/82 10:01:27, 0.5270 SECS, 0.1048 SUPS PAGE 1 OF 1

THIS TEMPLATE IS USED TO DIRECT OUTPUT TO AN EXTERNAL DEVICE(S) ONCE A REPORT HAS BEEN GENERATED. THIS EXAMPLE DIRECTS THE REPORT WHICH APPEARS ON THE PREVIOUS PAGE TO THE TERMINAL PRINTER.

RIMS USER INTERFACE - FILE APPEND

______ file append module append

append records selected according to APTEST (criteria in 'from' file-defn)

from: file TESTR6 in file-defn ______ in domain ______
to: file TEST62 in file-defn ______ in domain ______

records with duplicate keys are to be disposed of as follows:

not append the duplicate record from the 'from' file? Y or,
replace the record in the 'to' file with the record in the 'from' file? _ or,
the entire 'append' action will fail if any duplicate keys are encountered.

COMPLETE 03/15/82 11:11:34, 0.0350 SECS, 0.0442 SUPS PAGE 1 OF 1

THIS MODULE IS USED TO EXTRACT RECORDS BASED ON A SELECTION CRITERIA (APTEST) FROM A FILE IDENTIFIED AS THE "FROM FILE" (TESTR6) AND APPEND THEM TO A FILE DESIGNATED AS THE "TO FILE" (TEST62)
NOTE: DISPOSAL OF RECORDS WITH DUPLICATE KEYS MUST BE INDICATED.

RIMS USER INTERFACE

SYSTEM INFORMATION

RIMS USER INTERFACE - LIST USER ENTITIES AND SYSTEM DOCUMENTATION

```
_____            rims help module                                    (help)

 the rims help module is designed to supply application and rims system user
 information.  to request any information, put a 'y' in the appropriate space.
     file: TESTR6    * defn: TESTR6     domain: ______

   Y  template id's       Y  report id's and titles        Y  procedures
   Y  select criteria     Y  computational def's           Y  data item list

 rims system information (documentation):

   _  signon      _  data base definition (dbdef)
   _  security definition (secdef)
   _  single record transaction template definition (srtdef)
   _  single record transaction usage (s.r.t.)
   _  report generation (report)   _  report definition (rptdef)
   _  selection criteria definition (seldef)
   _  hard copy print request (print)
   _  procedure execution (proc)   _  procedure definition (prcdef)
   _  append (append)   _  computation definition (cmpdef)
* note: defn/domain are input if different than ones currently accessed

COMPLETE 03/15/82 11:34:10, 2.4650 SECS, 0.1110 SUPS     PAGE 1 OF 1
```

ALL OUTPUT GOES TO THE PAGE FILE. THE INFORMATION REQUESTED ABOVE REQUIRES 7 PAGES OF OUTPUT. THOSE PAGES FOLLOW.

RIMS USER INTERFACE - DATA BASE STATISTICS

______	file: TESTR6	file-defn: ______	domain: ______	dbstat

data base statistics

date and time of file creation __/__/__ __:__:__

record counts

original	current	inserts	modifies	deletes
_____	____0	____0	____0	____0

file characteristics

# data blocks	block size (words)	key size (words)	record size (words)	no. index blocks
____	____	__	____	___

COMPLETE 03/15/82 10:09:33, 1.9720 SECS, 0.0242 SUPS PAGE 1 OF 1
INS:STEP 2 OF 2:ENTER 'TID' FOR NEXT STEP OR DATA IF NXT STEP HAS SAME 'TID'

AFTER THE DATA HAS BEEN INPUT FOR THE CURRENT STEP (DBSTAT), THE ABOVE UNDERSCORED STATEMENT WILL APPEAR. AT THAT TIME ENTER TEMPLATE ID FOR NEXT STEP (REPORT)

RIMS USER COMMUNITY

PERSONNEL

PROCUREMENT

FACILITIES ENGINEERING

BUDGET

SUPPLY INVENTORY

TERMINAL INSTALLATION/MAINTENANCE, WORK REQUESTS

PERFORMANCE (SYSTEM/APPLICATION) DATA

<u>RIMS USER COMMUNITY</u>

(1) PERSONNEL - THE JSC PERSONNEL OFFICE USES THE RIMS TO:

SUPPORT PERSONNEL MANAGEMENT INFORMATION SYSTEM (PMIS) REPORTING AND AD HOC INQUIRIES

PROJECT FUTURE MAN HOUR USAGE PER YEAR BASED ON THE NUMBER OF EMPLOYEES

SUPPORT MERIT PAY REPORTING AND INQUIRY

PROVIDE STATUS AND TRACKING JOB APPLICATIONS AND OFFERS

PROVIDE STATUS AND TRACKING OF COOPERATIVE EDUCATION STUDENTS, UPWARD MOBILITY PERSONNEL, INTERNS, AND SUPERVISORY TRAINING PLANS

(2) PROCUREMENT - THE JSC PROCUREMENT OPERATIONS OFFICE USES RIMS TO:

REPORT STATUS OF PURCHASE REQUESTS/CONTRACTS ON MONTHLY BASIS FOR THE BUYING BRANCHES

TRACK AWARD FEE CONTRACT INFORMATION AND DATES

RETRIEVE NASA PROCUREMENT REGULATIONS/CLAUSES AND DETERMINE IF THEY ARE REQUIRED, APPLICABLE, OR OPTIONAL FOR CERTAIN CONTRACTS

TRACK DUE DATES ON MAJOR PROCUREMENTS

(3) ENGINEERING - THE FACILITIES ENGINEERING AND PLANT ENGINEERING DIVISION USES RIMS TO:

PROVIDE STATUS AND TRACKING OF WORK REQUESTS FROM RECEIPT THROUGH DESIGN

PROVIDE STATUS AND TRACKING OF CRITICAL SPARES

RIMS USER COMMUNITY

(4) BUDGET MANAGEMENT/PREPARATION - VARIOUS ORGANIZATIONS USE RIMS TO:

SUPPORT INTEGRATION OF DIRECTORATE/OFFICE POP SUBMISSIONS AND COMPARISON OF BUDGET GUIDELINES/MARKS AGAINST THOSE SUBMISSIONS FOR PROGRAM OFFICE AND CENTER MANAGEMENT REVIEWS PRIOR TO SUBMISSION TO HEADQUARTERS

MAINTAIN DIVISION AND BRANCH LEVEL BUDGETARY DATA TO SUPPORT THE JSC POP REQUIREMENTS

(5) SUPPLY INVENTORY - THE LOGISTICS DIVISION USES RIMS TO:

MANAGE THE REDISTRIBUTION OR DISPOSAL OF EXCESS MATERIAL AND CAPITAL EQUIPMENT UNDER JSC CONTROL

(6) TERMINAL INSTALLATION/MAINTENANCE/WORK REQUESTS - THE IDSD USES RIMS TO:

PROVIDE STATUS AND TRACKING OF THE IDSD TELECOMMUNICATIONS (DATA) SYSTEM CONFIGURATION

PROVIDE INVENTORY AND MAINTENANCE ACCOUNTING

PROVIDE STATUS AND TRACKING OF WORK REQUESTS FOR INSTALLATION/MOVE/REMOVAL OF TERMINAL STATIONS

(7) PERFORMANCE DATA - THE IDSD USES RIMS TO:

RESPOND TO MANAGEMENT QUESTIONS RELATED TO COMPUTER SYSTEM PERFORMANCE

OVERVIEW OF THE INTEGRATED PROGRAMS FOR AEROSPACE VEHICLE DESIGN (IPAD) PROJECT

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

S. L. Venneri
NASA Headquarters

SUMMARY

To respond to national needs for improved productivity in engineering design and manufacturing, a NASA supported joint industry/government project is underway denoted Integrated Programs for Aerospace-Vehicle Design (IPAD). The objective is to improve engineering productivity through better use of computer technology. It focuses on development of data base management technology and associated software for integrated company-wide management of engineering and manufacturing information. The project has been underway since 1976 under the guidance of an Industry Technical Advisory Board (ITAB) composed of representatives of major engineering and computer companies and in close collaboration with the Air Force Integrated Computer-Aided Manufacturing (ICAM) program. Results to date on the IPAD project include an in-depth documentation of a representative design process for a large engineering project, the definition and design of compûter-aided design software needed to support that process, and the release of prototype software to manage engineering information. This paper provides an overview of the IPAD project and summarizes progress to date and future plans.

INTRODUCTION

The national need for improved productivity has become increasingly apparent with recent statistics of zero or negative growth in gross national product. Significant improvements in aerospace productivity are believed possible through effective utilization of current and future CAD/CAM technology. The IPAD project goal is to increase U.S. aerospace industry productivity through application of computers for integrated company-wide management and control of engineering design data and manufacturing information.

In the early 1970's, NASA-funded feasibility studies showed that dramatic increases in engineering productivity were feasible through the automation of routine information handling tasks. These results, which were extensively reviewed by the aerospace and computer industry showed that such automation would directly decrease cost and flow time in the product design proces and would improve the competitive position of the U.S. aerospace industry. Based on these and other results, NASA began the IPAD project in 1976 to develop the appropriate technology and associated computer software. Work under the IPAD project is being done princially through a NASA prime contract to the Boeing Commercial Airplane Company supported by appropriate subcontracts and under the guidance of an Industry Technical Advisory Board (ITAB) composed of members of aerospace and computer companies. The ITAB concept provides an innovative and effective management approach for a joint industry/government high technology R&D

effort. Figure 1 illustrates the ITAB membership and delineates the major activities performed by this organization. NASA has found the ITAB concept to be very effective in reviewing ongoing work and developing long term goals for the IPAD project.

DESCRIPTION OF THE IPAD SYSTEM

The IPAD project conducted a detailed dissection of a reference design process using conventional takeoff and landing, SST, and hydrofoil vehicles as baselines. This work helped to clarify the designer's work environment, to identify ways to improve it,and to determine how best to use CAD/CAM to support it. Several ITAB companies, emulating this assessment, compared their design approach to the IPAD studies. For example, Lockheed-Georgia dissected the development cycle for a military transport aircraft comparing military program phasing with phasing used in IPAD studies, and Rockwell International compared its military aircraft systems development cycle with the IPAD studies. Such activities help a company define and implement an interated CAD/CAM capability.

Through joint industry/government efforts stimulated by the IPAD project, technology has now reached the stage where the basic requirements for an integrated CAD/CAM system are defined and its key software elements identified. The primary engineering interface operates through interactive terminals to select or control events and to define or display designs. The requirements and key software for such an integrated CAD/CAM software system (denoted "full IPAD") have been defined under the IPAD project and a preliminary design prepared with the following basic software components:

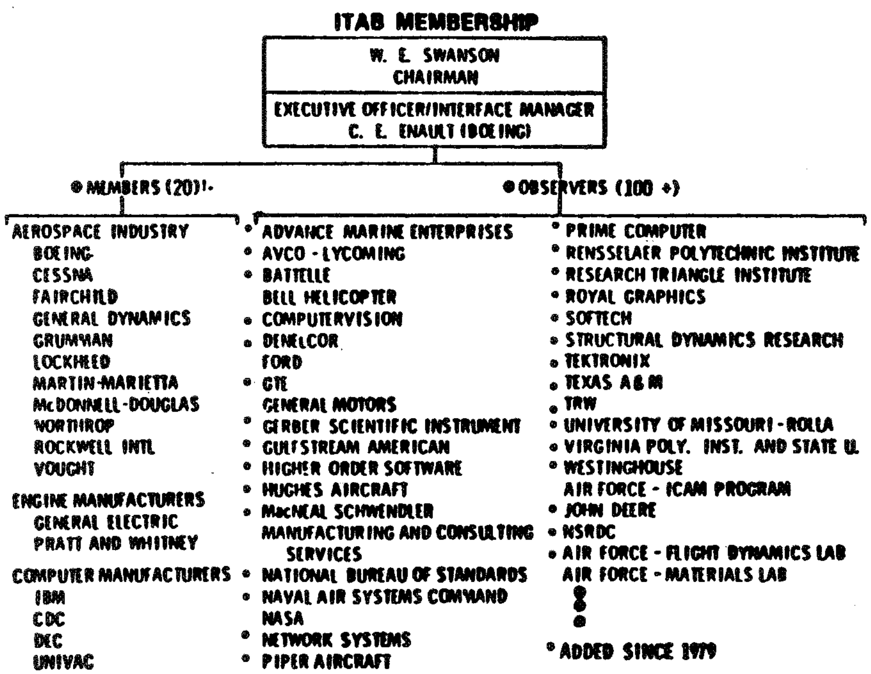
- Executive software (IPEX) to control user-directed processes through interactive interfaces with a large number of terminals in simultaneous use by engineering and management personnel and to provide communications among computer hardware within and outside the distributed computing system of IPAD.

- Data management software (IPIP) to provide a comprehensive, versatile ability to store, track, protect, and retrieve exceptionally large quantities of data maintained on many different devices.

- Geometry and graphics software to provide a wide range of capabilities for information display and geometry creation and manipulation, including design and drafting and interactive and display graphics.

- General utility software to give users an assortment of features to aid in using CAD/CAM, including user languages, tutorial aids, report generators, design/manufacturing indexing and routing facilities, error diagnostics and configuration-management aids.

IPAD INDUSTRY TECHNICAL ADVISORY BOARD (ITAB)



ITAB ACTIVITIES

GUIDE DEVELOPMENT TO MEET INDUSTRY NEEDS

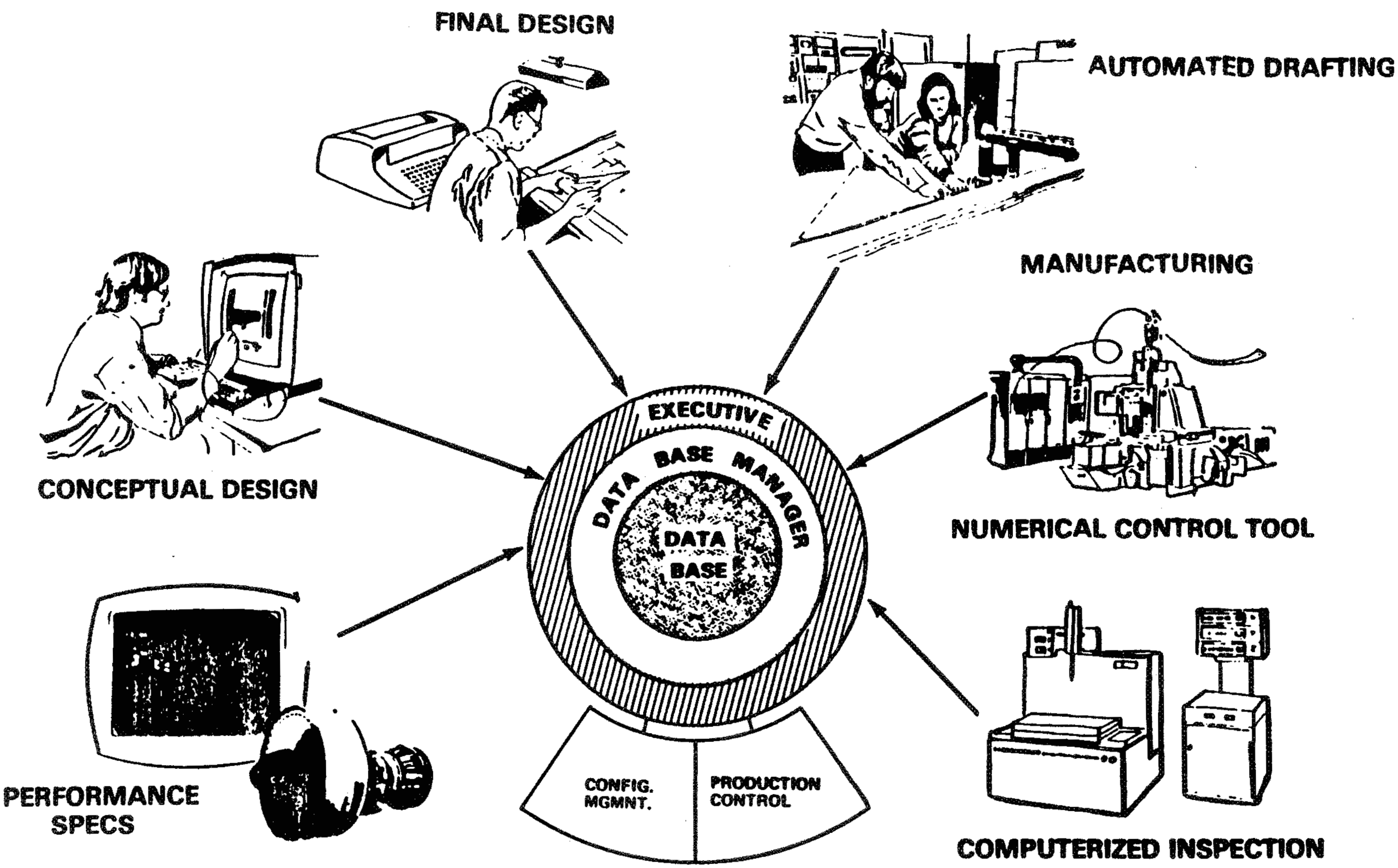
REVIEW/CRITIQUE ONGOING WORK

EVALUATE PROTOTYPE SOFTWARE

USE IPAD TECHNOLOGY AND PRODUCTS TO SPUR IN-HOUSE PLANNING AND DEVELOPMENT

Figure 1

PRODUCT INTEGRATED DESIGN (EXPANDED IPAD)
COMPUTER SCIENCES FOR IMPROVED PRODUCTIVITY



NASA

OAST NASA HQ RT80-4817(1)
REV. 5-22-81

Figure 2

Basically, such a Full IPAD system would be a general-purpose interactive computing system to support engineering design, with significant capability to manage and manipulate engineering data. It would support activities at all levels of design--conceptual, preliminary, and final--for a typical company mix of development projects, and would aid in the assembly and organization of design data for manufacturing. Figure 2 presents how the IPAD data base management system would be the common interface to the various disciplines in an aerospace company.

Much progress has been made in recent years toward the integrated CAD/CAM capability. Several minicomputers or mainframes exist or are evolving which can serve as the basis for an integrated CAD/CAM system. Numerous technical programs (e.g., NASTRAN) and design/drafting systems (e.g., AD-2000) exist or continue to be developed to execute engineering analysis and design tasks, and many of these programs have already been connected to improve CAD/CAM capabilities. Means for integrating manufacturing activities, together with development of a set of user utilities denoted General Utility Systems, are under development in the ICAM program. However, critical technology issues which remain for an integrated CAD/CAM system include data management, data communication, geometry, and company management support.

Lack of the right technology and software to manage engineering and scientific information has been a major stumbling block to development of an integrated CAD/CAM system. ITAB thus urged the IPAD project to focus on data management for design and manufacturing. The needs have been identified by investigating the data flow and user requirements of the reference design processes for several aerospace vehicles. The IPAD project also conducted a limited assessment of design/manufacturing interface requirements. The ICAM program now has underway a thorough investigation of data management requirements for manufacturing, with work closely coordinated with IPAD.

IPAD results to date indicate that a CAD/CAM data-management system must meet at least the following requirements:

- Accommodate many different views of data from a variety of users and computing storage devices.

- Allow many levels of data description to support a wide variety of engineering organizations and tasks.

- Permit easy changes in data definition as work progresses.

- Allow data to be distributed over networks of computers of various manufacture.

- Permit data definitions to be readily extended as needs arise.

- Store and manipulate geometry information.

- Embody adequate configuration-management features.

- Provide broad means to manage information describing stored data.

To meet these requirements IPAD is developing a multischema ("multiview") IPIP for a network of computers (Figure 3). Its software defines and manipulates information through three different types of format: logical (or user) schemas to organize information appropriate for each user, internal schemas to describe the way a specific machine stores information, and mapping schemas to connect the various logical and internal schemas.

The multischema approach permits an unlimited number of data formats for different applications and these formats can be readily changed as the need arises. Yet the data are stored only once on a specific computer in a distributed network.

This year the IPAD project will complete a prototype IPIP data management system and limited IPEX executive software for a CDC CYBER/NOS computer. Figure 4 presents the performance capability of the IPAD system for a test data base. Initial results were unacceptable for production level software but various enhancements have reduced data access response times to acceptable levels (1-3 seconds). An assessment of a DEC VAX implementation has been completed but no software has been developed.

The IPAD project has also developed a quick-response data-management subsystem denoted Relational Information Manager (RIM) to be used either as a stand-alone component or as an interactive interface to IPIP. Figures 5 and 6 illustrate the desired configuration for RIM and IPIP to support an integrated engineering analysis capability. The RIM data base system will be for single user, quick access, and interactive query capability, while the IPIP data management system will be for project level control of a complete library of data and be accessed with moderate response times. A network capability could be used to connect the two data base management systems together. RIM is already in use at numerous sites on CDC, DEC, and other computers and has proven to be a significant aid to managing data typical of engineering design studies. For example, it served as the key data management capability at NASA-Langley for a 600,000-word data base associated with a study of 8000 tiles on the Orbiter (Figure 7).

TYPICAL ARRANGEMENT OF APPLICATIONS AND SCHEMAS

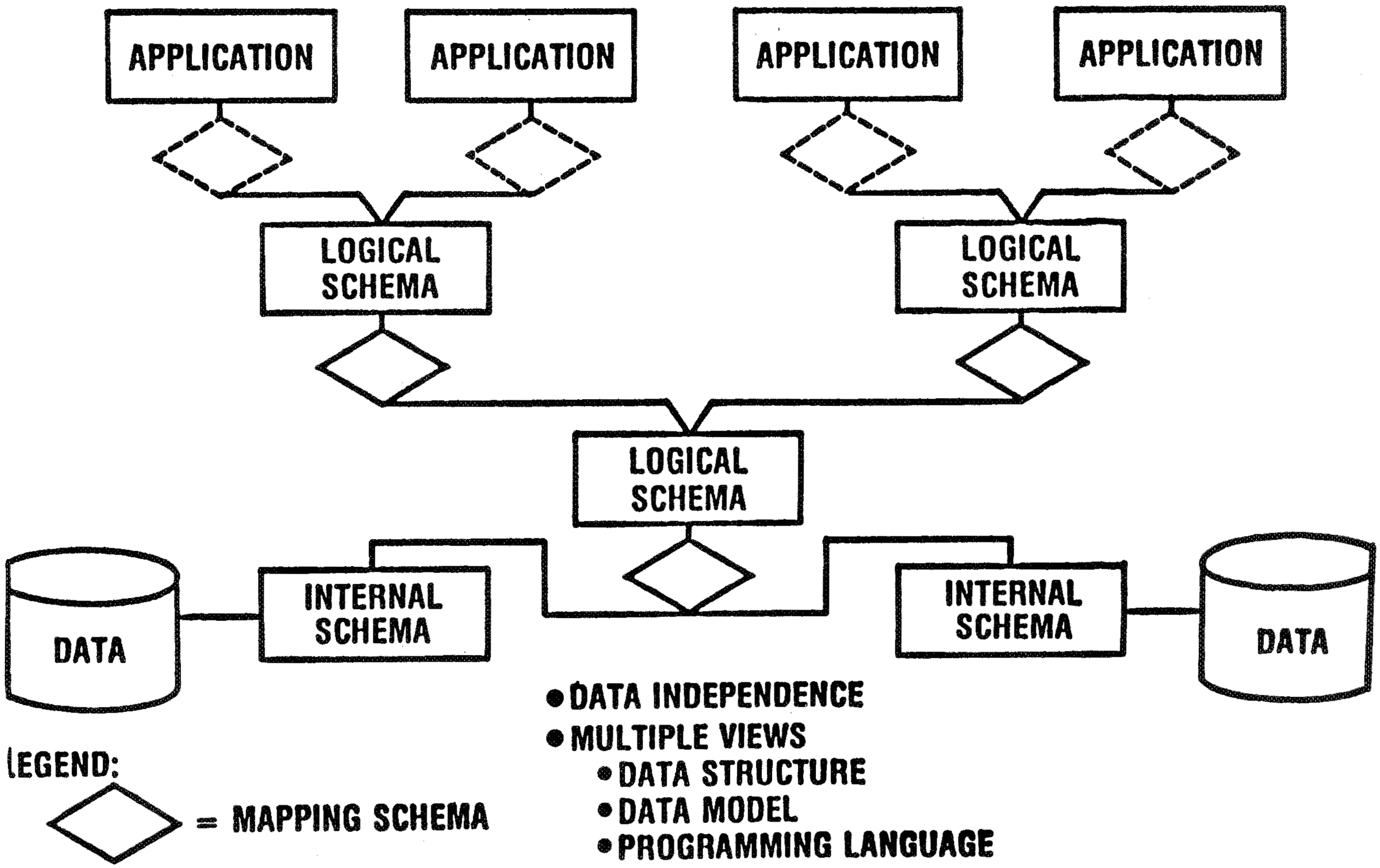
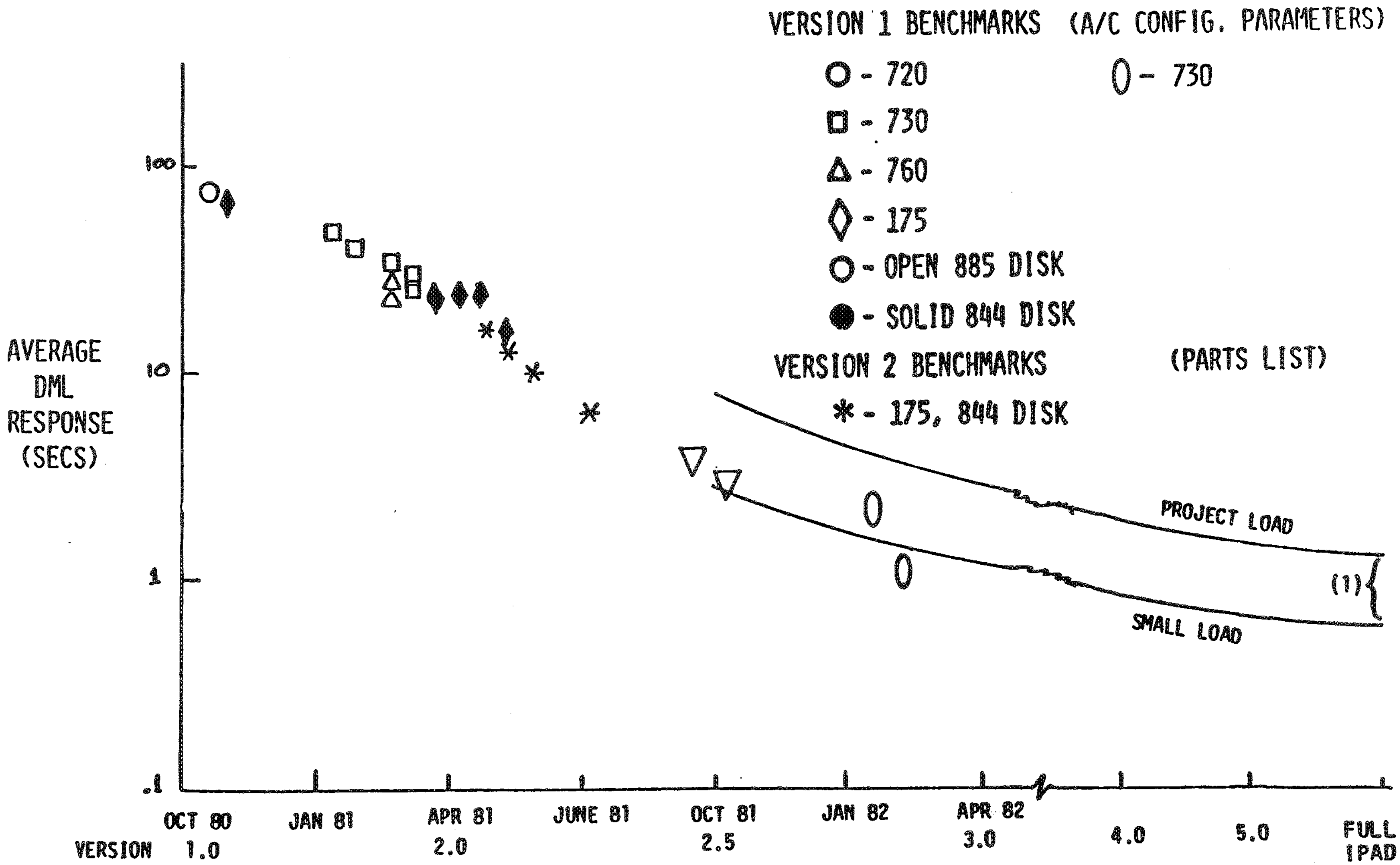


Figure 3

Figure 4

IPIP PERFORMANCE RESULTS/GOALS



(1) FULL IPAD REQT 106 (6/10/77)
(ACCEPTABLE HUMAN NEEDS)

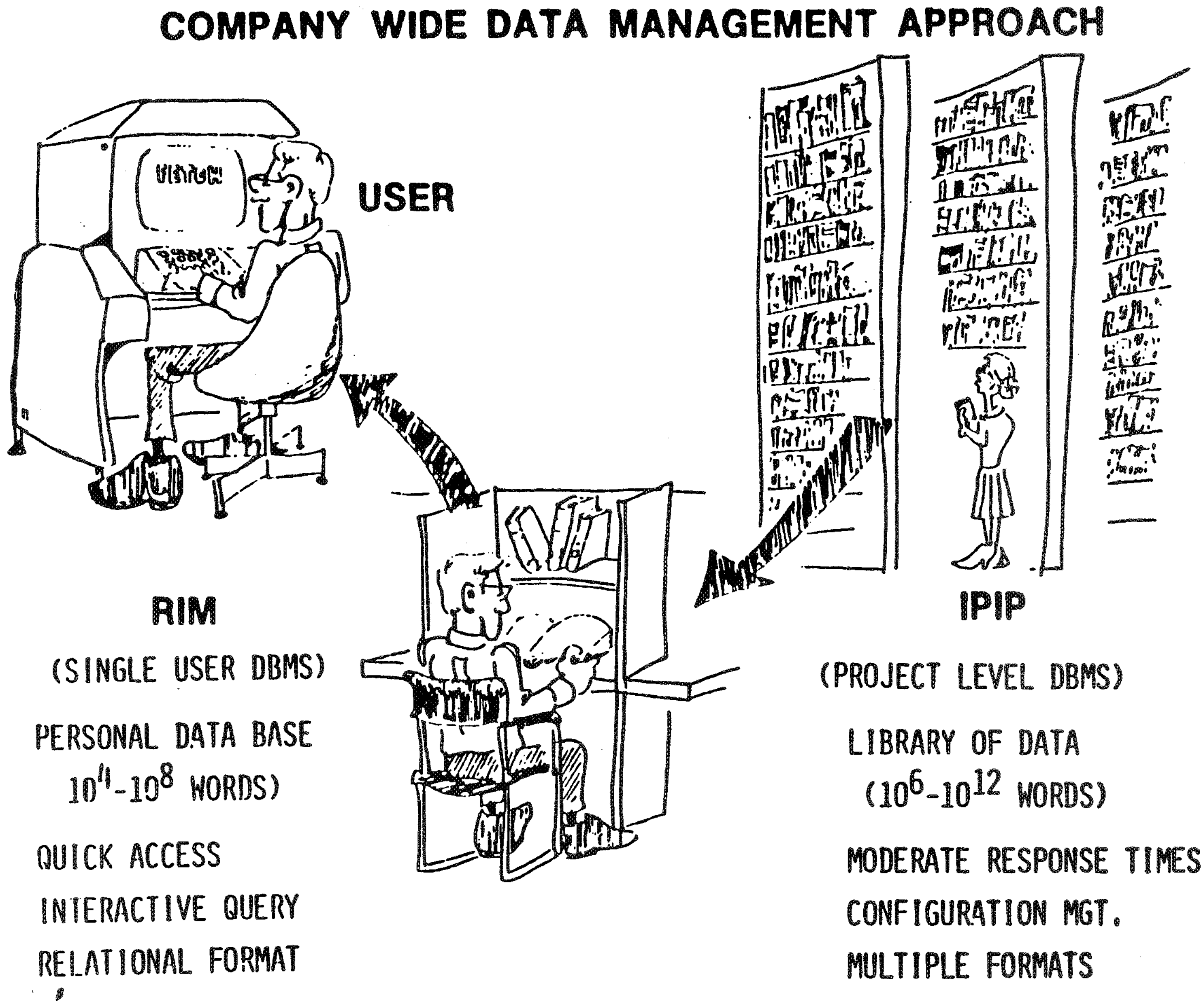

COMPANY WIDE DATA MANAGEMENT APPROACH
USER
RIM
(SINGLE USER DBMS)
PERSONAL DATA BASE
10^4-10^8 WORDS)
QUICK ACCESS
INTERACTIVE QUERY
RELATIONAL FORMAT
IPIP
(PROJECT LEVEL DBMS)
LIBRARY OF DATA
(10^6-10^{12} WORDS)
MODERATE RESPONSE TIMES
CONFIGURATION MGT.
MULTIPLE FORMATS

Figure 5

IPAD CONTRIBUTIONS TO
INTERDISCIPLINARY ANALYSIS AND SYNTHESIS SYSTEM

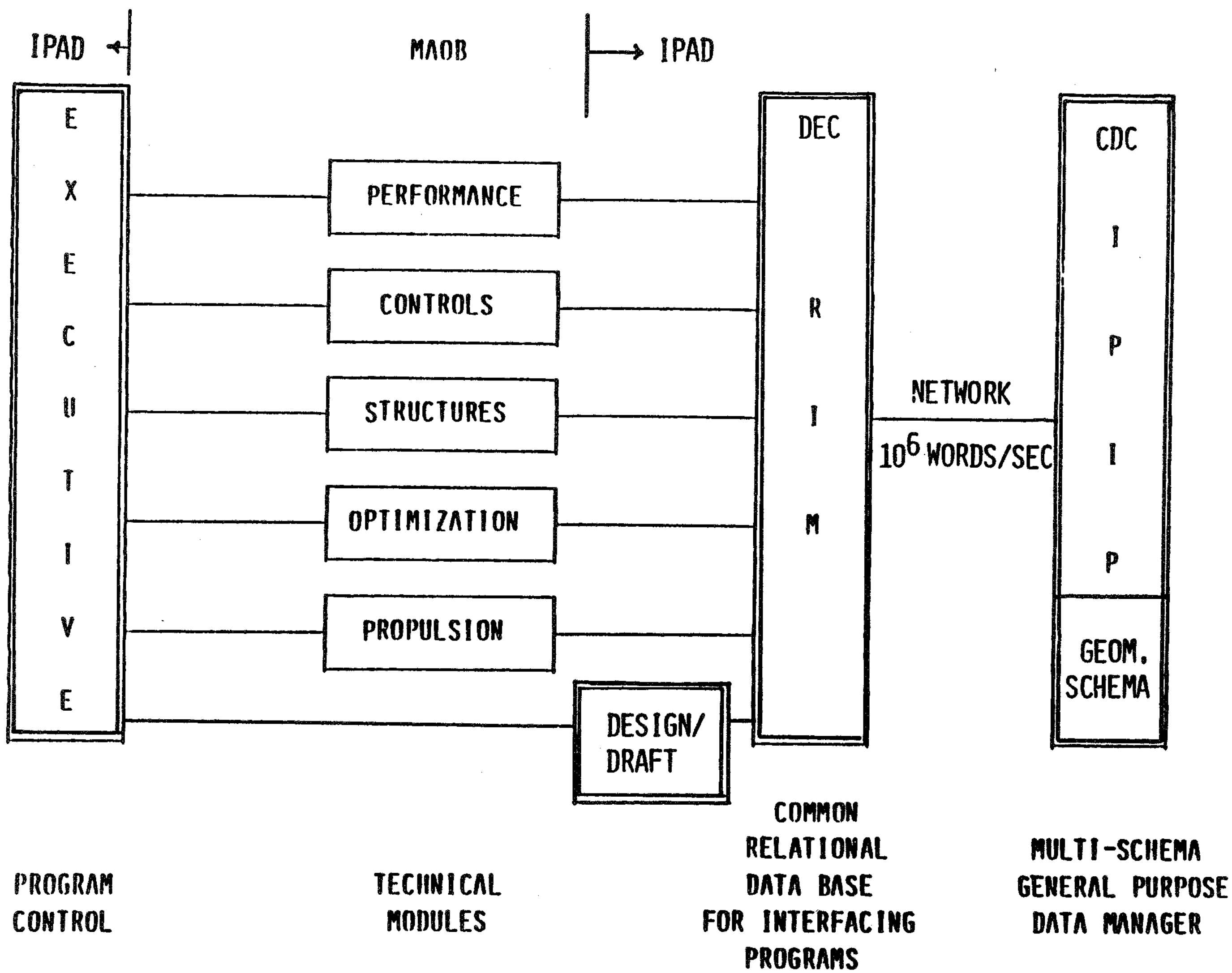
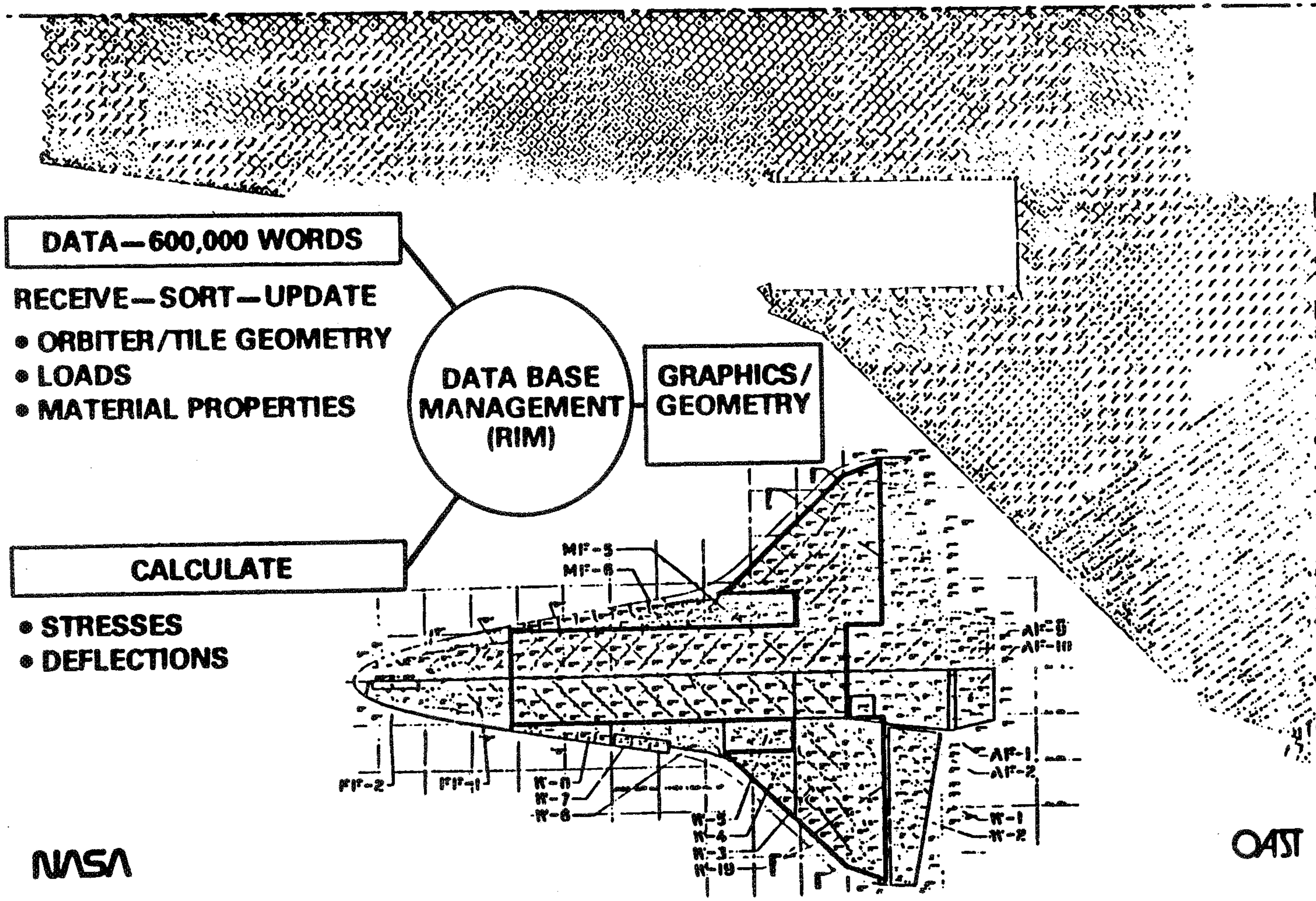


Figure 6

IPAD/RIM APPLICATION TO SHUTTLE TILE ANALYSIS



NASA OAST

Figure 7

FUTURE PLANS

Ability to distribute data over a network of different makes of computer represents a major CAD/CAM challenge. Distributed data management is considered by ITAB to be the next critically needed IPAD project thrust. The IPAD software design approach was carried out for such a computer network, and a high-speed network has been developed for a CDC/DEC complex. Data management implementation to date has been limited to single computers.

For 1982 and beyond, ITAB strongly recommends the IPAD project work on technology needed to extend data-management concepts to a unified system spanning design and manufacturing, contractors, various computers, and geographically dispersed sites. Such a distributed data-management capability appears feasible judging from the full IPAD system design and the IPAD software developments to date. It will require completing development of the IPEX executive software, as well as extending the IPIP data management software. The resulting system should offer substantial productivity improvements. Figure 8 illustrates this desired system for future IPAD distributed data management in both the engineering and manufacturing environment. The key technology areas are geometry capability and computer networking.

Geometry - the thread that runs throughout the design and manufacturing proces - represents a major link in development an integrated CAD/CAM system. Interactive 3-D design/drafting subsystems based on "wire frame" concepts are fairly well developed, and several alternatives exist to serve as a foundation for geometry descriptions to support design and drafting tasks and to provide connections to numerical control machines. The IPAD software will incorporate the capability to handle engineering geometry information and will encompass the range from points and curves to solid 3-D descriptions.

In summary, IPAD plans for future data management research are: (1) management of engineering geometry data; (2) design/manufacturing data management interfaces, and (3) distributed data base management.

FUTURE TECHNOLOGY CHALLENGE (1982→)

DISTRIBUTED DATA MANAGEMENT IN AN INTEGRATED DESIGN/MANUFACTURING ENVIRONMENT

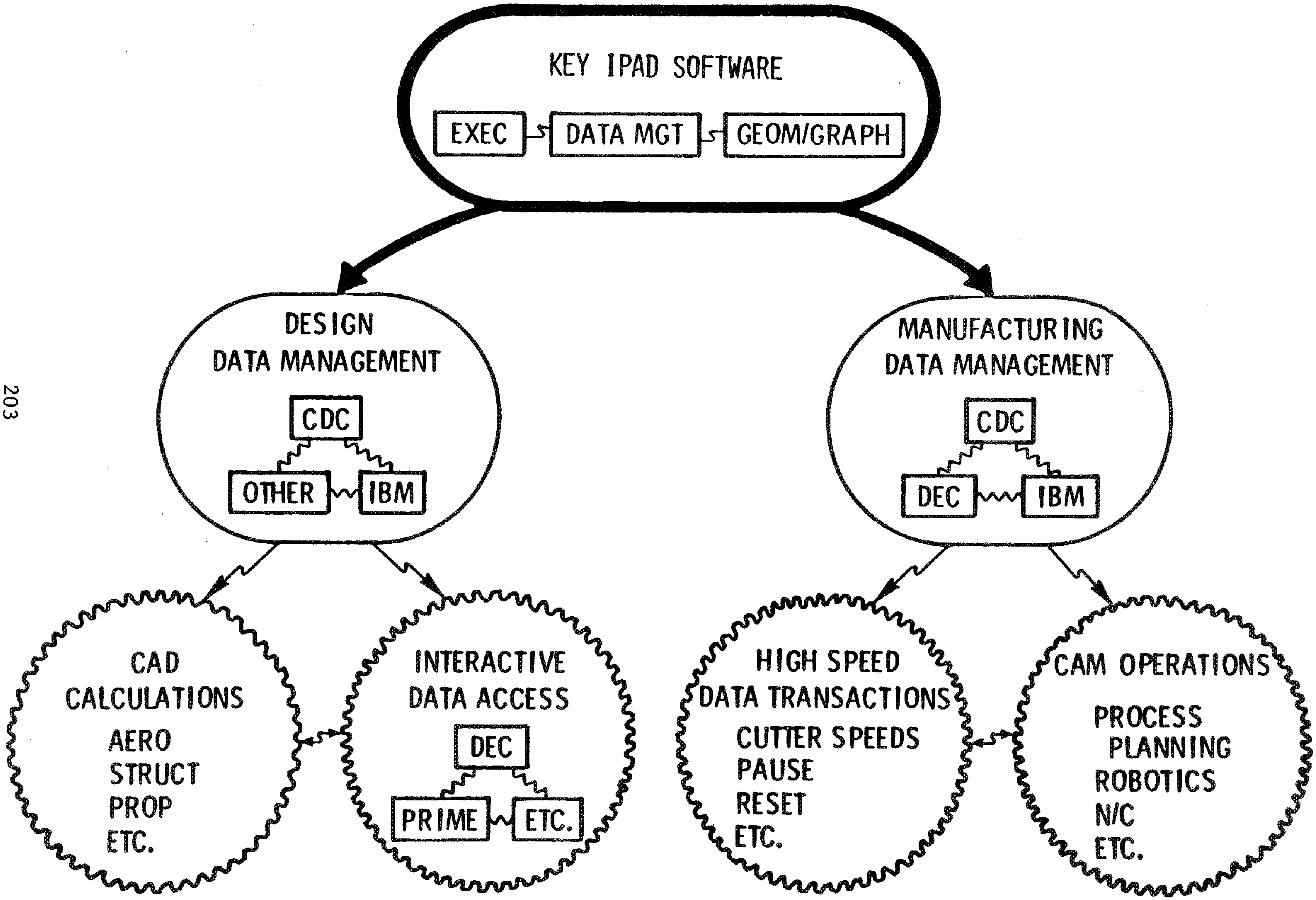


Figure 8

PLANNING THE FUTURE OF JPL's MANAGEMENT AND ADMINISTRATIVE SUPPORT SYSTEMS AROUND AN INTEGRATED DATABASE

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

M. M. Ebersole
Computing and Information
Services Office
Jet Propulsion Laboratory

ABSTRACT

JPL's management and administrative support systems have been developed piece-meal and without consistency in design approach over the past twenty years. These systems are now proving to be inadequate to support effective management of tasks and administration of the Laboratory. New approaches are needed. Modern database management technology has the potential for providing the foundation for more effective administrative tools for JPL managers and administrators.

Plans for upgrading JPL's management and administrative systems over a six year period evolving around the development of an integrated management and administrative data base are discussed.

I. Introduction

The Management and Administrative Support Systems (MASS) are the computer applications and data which support the management and administrative activities of the laboratory. For the most part these systems are comprised of software that is known as the Administrative Computing Services which operate on the IBM 3032 at Booth Computing Center at Caltech. The ACS systems have been developed incrementally at JPL over the last 20 years with low budget, limited management attention, no consistent organization of the ACS into systems, and little documentation. The lack of emphasis on administrative computing has resulted in the following problems:

(i) Each system was designed independently as the need arose. Therefore, many systems are interdependent so that a change in one affects many others.

(ii) There is substantial redundancy and overlap in processes and data.

(iii) Data access is poor, usually limited to once a month.

(iv) The responses and capability of the systems are not adequate for many small tasks/projects. This has fostered a proliferation of special software "add-ons."

(v) Frequency and granularity for planning and reporting are inadequate for today's needs (e.g. planning by quarter, reporting by month.)

Because of these deficiencies, a complete upgrade of JPL's MASS are urgently needed. JPL's Computing and Information Services Office (CISO) has undertaken the development of a plan which would upgrade the MASS over a period of six years. This foundation of the plan is the use of a generalized database management system which has the potential of overcoming the problems discussed above. The following describes the MASS plan with emphasis on the role of the database management system in providing the improved MASS capabilities.

II. Plan for MASS Development

1. Guidelines and Constraints for MASS Development

After a period of extensive review by JPL management and administrative system users, the following basic MASS development strategy has been endorsed and adopted:

1. The centrally located MASS will continue to operate in a IBM computing environment.

2. A commercially available generalized database management system (DBMS) will be purchased and installed on the IBM 3032 at JPL. This generalized DBMS will provide the capabilities to evolve a JPL Integrated Administrative Database and improved Administrative Systems in an orderly fashion.

3. Commercially available applications software will be used, with adaptation for JPL, whenever this approach is time and cost justified.

4. The first phase of MASS development activity will concentrate on acquiring and installing the generalized DBMS and improving financial planning capabilities including providing interactive planning.

5. There will be a steady evolution of new or significantly upgraded administrative systems over a six year period.

6. The planning and execution of all MASS development will be conducted in phases of 6 to 18 months with each phase redeveloping one major system or a small number of related systems.

2. MASS Design Methodology

The MASS Design methodology closely follows the approach suggested by D. S. Appleton and A. F. Cardenas.[1] The methodology suggests that getting the user involved interactively with a database while under development is important in developing flexible, responsive databases. The design of the MASS, consisting of both databases and applications, will start by developing the database "conceptual schema" which defines the items that can be stored in the database and the relationships between them. The development of the conceptual schema applys to the entire JPL MASS Integrated database. Next the data collection and input structure necessary for establishing and maintaining the database content will be established and finally the development of the MASS applications which use the database will be done.

3. MASS Development Phases

Mass development will be conducted in four phases over six years. It is expected these will be done in the sequence listed. However an evaluation process will be followed to prioritize development for maximum benefit at all stages.

1. Phase I (1982-1983) the "System for Resources Management (SRM)"
2. Phase II (1984-1985) the "financial feeders" phase
3. Phase III (1986) the "financial related systems" phase
4. Phase IV (1987) the "standalone systems" phase

Figure 1 shows the evolution of the four phases of development. At the top of the figure the evolution of the JPL Integrated Administrative Database, comprised of several smaller databases is shown. In the center of the figure, a block identified as "Generalized Database Management System" is shown.

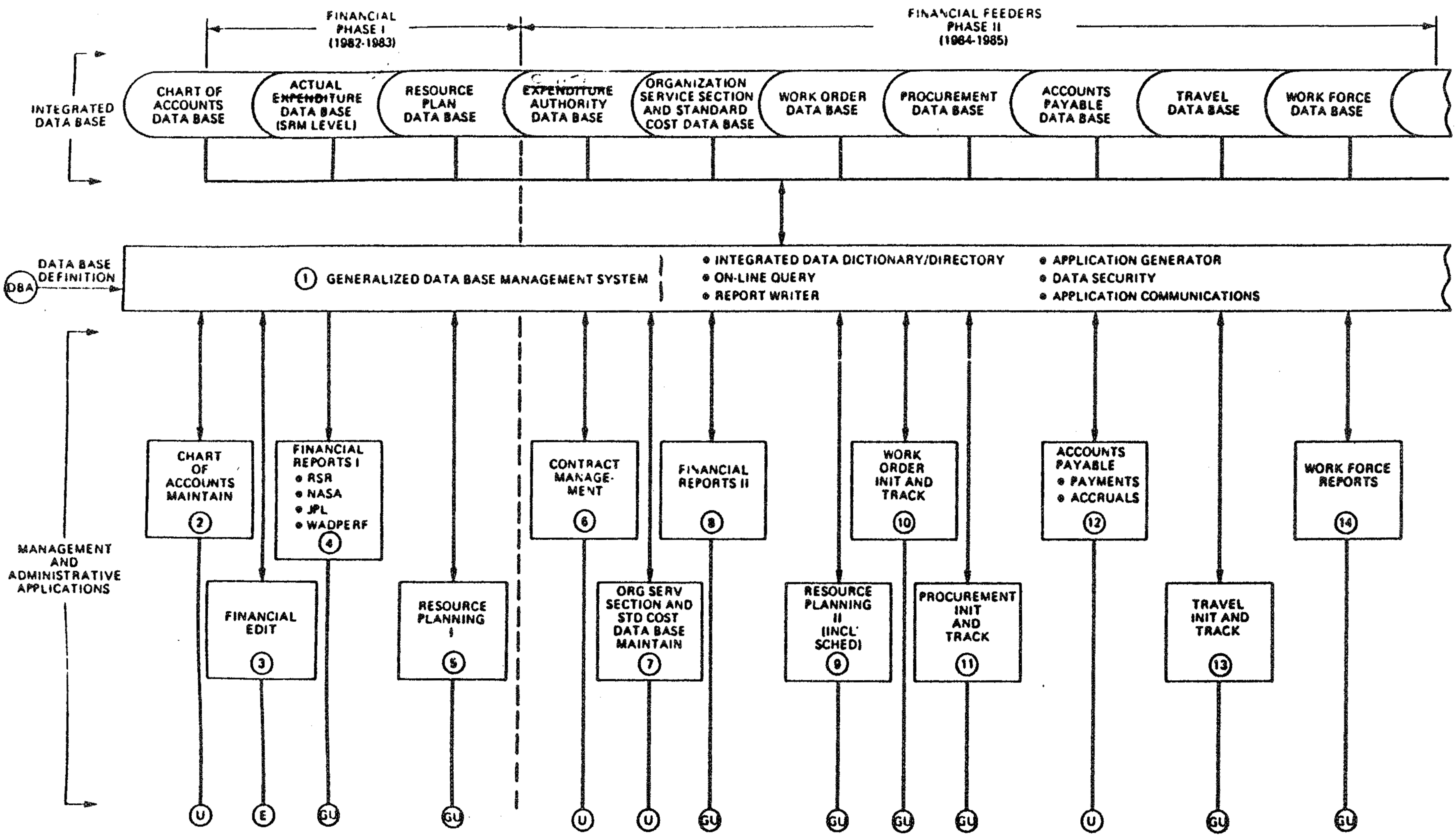
It is the capabilities of DBMS that provides the framework for virtually all aspects of MASS development from documenting programs and data for existing systems, to developing high-level and detailed database designs, to use in developing applications, and to use in an operational sense for accessing data.

Across the bottom of Figure 1 is shown several boxes, each of which represents a specific administrative application. The order in which the boxes appear represents the order in which the applications will be developed. By developed is meant that the system could be modified from an "existing" system, could be a purchased package, or could be newly developed.

The activities of each of the four phases is further described as follows:

1. Phase I - "SRM" Phase

MASS EVOLUTION

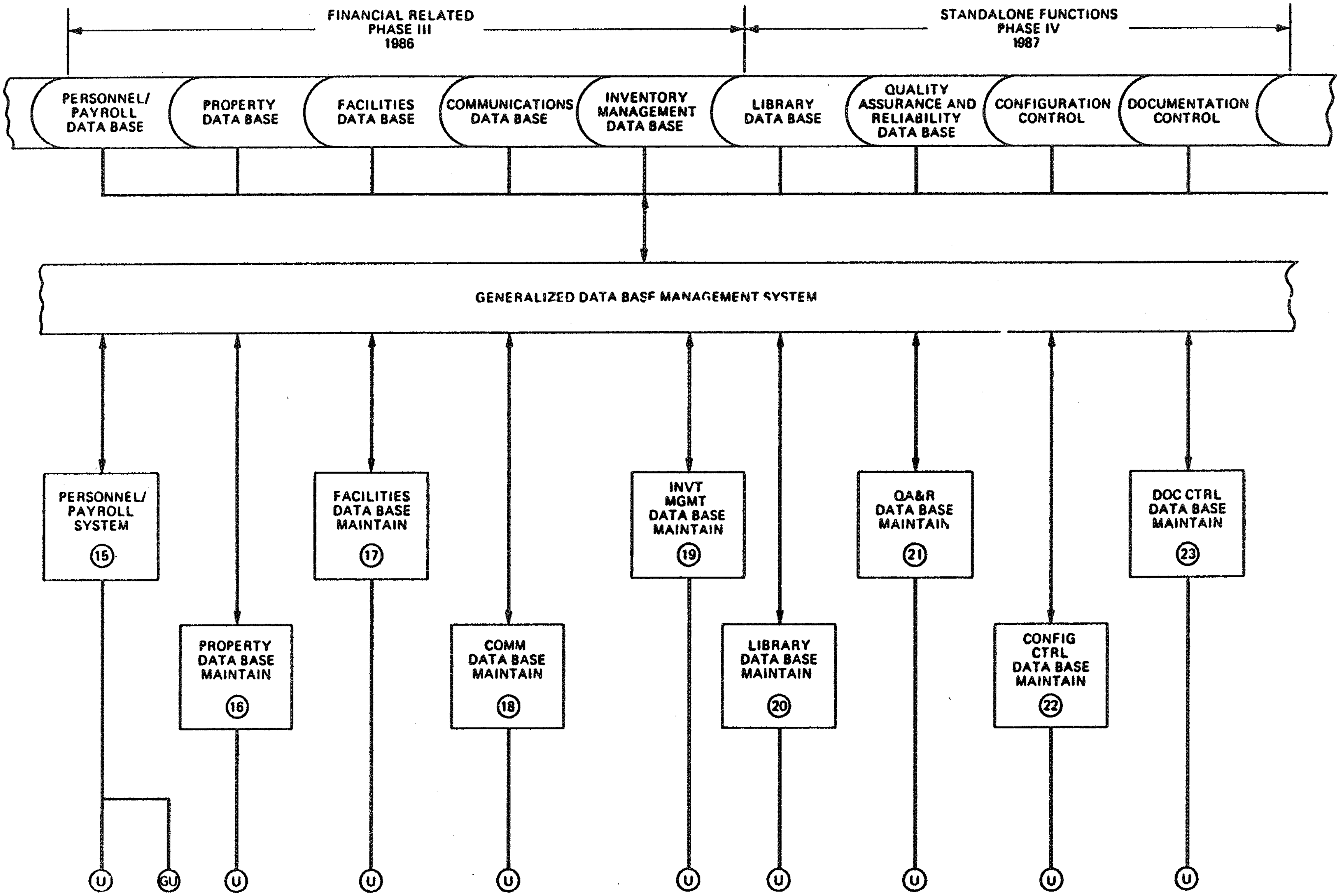


KEY:

- (DBA) - DATA BASE ADMINISTRATOR
- (U) - DATA BASE USER; DATA UNDER CONTROL OF SPECIFIC ADMINISTRATIVE ORGANIZATION
- (GU) - DATA BASE GENERAL USER; DATA UNDER CONTROL OF GENERAL USER OR PROJECT
- (E) - INTERFACE TO EXISTING FILES TO PROVIDE INTERFACE TO DATA BASE

Figure 1

MASS EVOLUTION (contd)



PAGE 2 OF 2

Figure 1

The first step in Phase I will be to acquire a generalized database management system (GDBMS). Key features of such a system will include an integrated data dictionary/directory, on-line query capability, application generation, and report writing capability.

In developing a software system using database management methodology, the database structure assumes an overriding importance. Because of this, the description of MASS as it will exist after Phase I begins here with a description of the database which will be implemented and used by 1 October 1983. This is not JPL's entire management and administrative database. Detailed financial data, inventory data, communications data, personnel data, library data and many other data types will remain to be added. However, all data categories will be included in the conceptual schema of the database to be finished by 1 January 1983. This will ensure that no major problems will be encountered in expanding the database after 1 October 1983.

The following describes the nucleus of a financial database. As shown in the Phase I portion of Figure 1, it consists of three parts:

1. A Chart of Accounts Database
2. An Actual Database
3. A Resource Plan Database

JPL uses two kinds of account numbers - its own, and NASA's. The chart of accounts database keeps track of both kinds of account numbers and of the relationships between them. The JPL account code forms the foundation of JPL's entire financial database structure. Any actual expenditure, work order, resource plan, or procurement is associated with a given JPL account number.

Although the databases described here will be kept in a centrally located and controlled area, capability will be provided to extract any subset of data and transmit it across the network to another computer.

The three databases of Phase I of Figure 1 are described one at a time below.

1. A Chart of Accounts Database

   A chart-of-accounts database is shown in Figure 2. This database models both NASA and JPL work breakdown structures to the individual account level. Each NASA account number is associated with one or more JPL account numbers and this relationship is modeled in the database as indicated by the arrow from box 5 to box 9. The database also identifies the project owner and the line organization owner of each JPL account code. Code sets (box 10) allow association of an arbitrary set of account numbers. An example of the use of a code set occurs in identifying all software related account numbers for a given project for use by a project software manager.

2. <u>An Actual Database</u>

This database, shown in Figure 3, contains summarized actuals associated with a given JPL account number. Actuals can be commitments, obligations, costs or expenditures when associated with the Financial system. Procurement and work order actuals are summed in here along with other actuals even though they will in subsequent phases also be maintained in their own databases. The actual database is essentially a form of "sales" database since it contains itemizations of deliveries against the "customers' orders" in the Contract Authority Database. When viewed in this context, commitments are a preliminary stage which result in obligations. Commitments are of two varieties - work orders and authorizations for procurement. There are more varieties of obligations each of which have sub-varieties. The relationships, though fairly numerous, can be structured to reflect the accepted accounting treatment for commitments, obligations, costs and expenditures. The relationships in this database are numerous and diverse so that working out the details of data structure will take considerable effort.

3. <u>A Resource Plan Database</u>

A resource plan database is shown in Figure 4. This database contains any number of resource plans for a particular account number but only one can be marked as approved. Scheduling information and task descriptions will both be included in this database in subsequent phases to facilitate resource planning. Neither of these things can be easily associated with a resource plan today.

Planning factors, including such things as burden rates and standard costs, are arbitrarily included in this database. They are used for resource planning and for updating of actuals. It is not completely clear how planning factors should be accessed, so they are shown as a stand-alone part of the database.

Applications to be Implemented by 1983

Modified, purchased, or newly developed applications programs (boxes 2 through 5 Figure 1) are described one at a time below.

1. <u>Chart of Accounts Maintenance</u>

The chart of accounts maintenance program will utilize the capabilities of the generalized database management system to maintain the chart of accounts. Databases update and retrieval will be done by terminal access. Standard reports will be generated weekly and monthly as is now done.

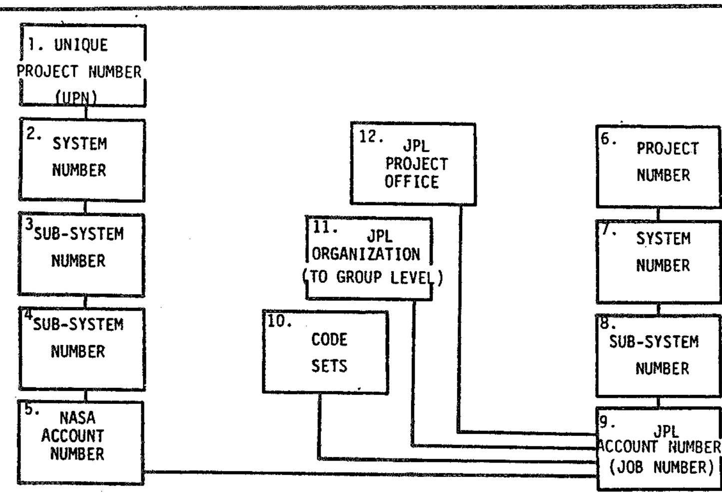


Figure 2
A CHART OF ACCOUNTS DATA BASE

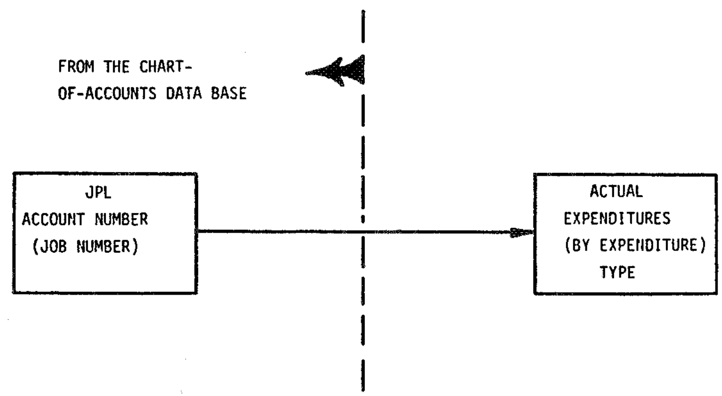


Figure 3
AN ACTUAL EXPENDITURES DATA BASE

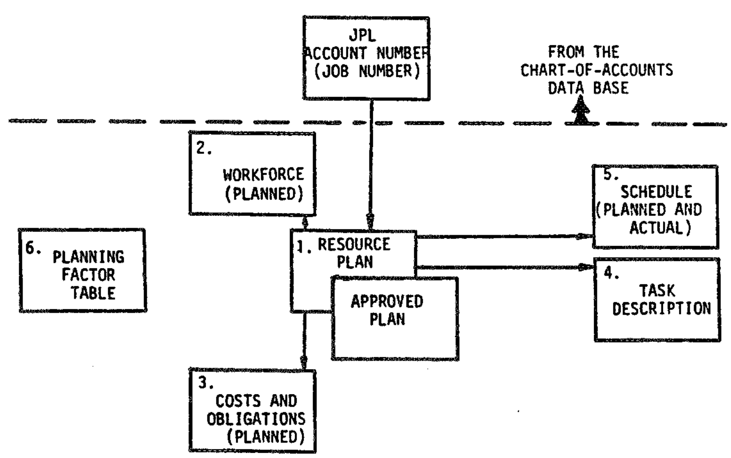


Figure 4
A RESOURCE PLAN DATA BASE

2. Financial Edit

All actual expenditure data will enter the database through the Financial Edit subsystem. This data flow is much like the present one and it may be that some actual code can be preserved from the present financial edit programs. Instead of producing data on ordinary tape and disk files as it does now, however, the new Financial Edit programs must update the actual database.

3. Financial Reports I

There are a large number of financial reports generated for JPL or NASA use. The RSR is one of them. The DBMS query language will be used to produce as many of these reports as possible. Procedural languages will be used only when report formats are too complex for a query language to handle. Reports will be produced either on the laser printer or directly on the screen of a user's terminal. Some standard reports will continue to be produced and distributed on a periodic basis, but others will be available only on a query basis.

4. Resource Planning I

Resource planning capabilities will be implemented partially in phase I and partially in phase II. In phase I on-line planning capabilities will be introduced including on-line resource planning, reporting summaries of plans and actuals, and on-line access to planning factors.

2. Subsequent Phases

Phase II "Financial Feeder" Phase (1984-1985)

Phase II will involve extending the capabilities of Financial Reporting and Resource Planning applications implemented in Phase I, but will primarily involve developing the database and applications for the "financial feeder" systems that feed detailed financial data into the summary level data of the systems developed in Phase I. In this phase the concept of user initiation and tracking of transactions such as work orders and procurements will be introduced.

The financial database will be extended to contain more detailed financial information. As shown in the Phase II portion of Figure 1, the following databases will be introduced into the integrated database in Phase II:

1. Contract (funds) authority database.
2. Organization services section and standard cost database
3. Work Order database.
4. Procurement database.
5. Accounts payable database.
6. Travel database.
7. Workforce database.

Data models for these seven databases will be developed when the conceptual design of this MASS integrated database is completed on 1 January 1983.

Phase III "Financial Related" Phase (1986)

The "financial related" systems are those which have only minor coupling to the financial systems. The need to introduce the data associated with these systems into the integrated database is less urgent than the financial dates because access to the data is required by a much narrower community of users. As shown in the Phase III portion of Figure 1, the following databases will be introduced into the integrated database in Phase III.

1. Personnel/Payroll database
2. Property database.
3. Facilities database
4. Communicationd database
5. Inventory management database

Data models for these five databases will be developed when the conceptual design of the MASS integrated database is completed in 1 January 1983.

Phase IV - "Standalone" System (1987)

The "standalone" systems are those which do not have interfaces with other systems. The following standalone databases will be introduced into the integrated database in Phase IV:

1. Library database
2. Quality Assurance and Reliability database.
3. Configuration Control database
4. Documentation Control database

Data models for these four databases of will be developed when the conceptual design of the MASS integratd database is completed in 1 January 1983.

3. Conclusions

This paper has presented a plan for an orderly upgrade of JPL's management and administrative systems with an integrated administrative database as the cornerstone. The success of this plan is contingent upon adherence to the design methodology and phases of development described. It is particularly important to:

1. Develop a rigorous process for selection of the generalized DBMS which provides the necessary tools for every phase of database involvement from conceptual design to operations.

2. Obtain user involvement during all phases of database development starting with developing requirements for applications and analyzing existing administrative processes to actual interaction with the data as early as possible to assess the suitability of the data and access methods.

3. Divide the development into short enough phases so that tangible results can be measured and the total effort is manageable.

1. D. S. Appleton and A. F. Cardenas, PDM 80 Prototype Development Methodology for Database, (Pasadena, Calif.: JPL, 1980)

DATA BASE MANAGEMENT SYSTEMS - A FEDERATED APPROACH

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

Robert W. Iverson
Jet Propulsion Laboratory

OUTLINE

MOTIVATION FOR THIS PRESENTATION

- NEED FOR A NEW APPROACH
  - GETTING FROM HERE TO THERE
  - INSTALLED CAPITAL INVESTMENT
  - RESOURCE LIMITATIONS
- EXPERIENCE AS A DEVELOPMENT MANAGER
  - COSTS AND TIME OF DEVELOPING AND CHANGING CENTRALIZED SYSTEMS
  - FRUSTRATIONS INTERFACING DISTRIBUTED SYSTEMS
- EXPERIENCE AS A USER
  - BENDING NEEDS TO AVAILABLE SYSTEMS
  - FINDING AND CORRELATING DISTRIBUTED DATA
  - MOVING DATA BETWEEN SYSTEMS

GOALS

- ALLOW DATA BASE MANAGEMENT SYSTEMS TO SUPPORT A WIDER VARIETY OF USERS
- MAKE DATA BASES ACCESSIBLE TO CLERICAL, ADMINISTRATIVE, AND MANAGEMENT PERSONNEL WITH LITTLE OR NO TRAINING IN COMPUTER PROGRAMMING
- BALANCE THE NEED FOR DECENTRALIZATION AND THE NEED FOR EFFECTIVE SHARING OF INFORMATION
  - RETAIN THE PROPER LEVEL OF RESPONSIBILITY FOR DATA BASES
- PROVIDE A UNIFIED AND COMPREHENSIVE SET OF SOFTWARE TOOLS FOR CREATING, MANIPULATING, MAINTAINING, AND MANAGING DATA BASES

CURRENT STATE OF DATA BASES

- MANY EXISTING DATA BASES USE 60's TECHNOLOGY
  - DATA IS OWNED BY THEIR PROGRAMS AND IS DIFFICULT TO ACCESS AND MANAGE
  - PHYSICAL AND MACHINE-DEPENDENT FACTORS IMPEDE FUTURE CONVERSION AND MAINTENANCE OF EXISTING DATA BASE APPLICATIONS
  - MUCH OF THE USE IS MANPOWER-INTENSIVE
- DECENTRALIZATION OF DATA PROCESSING HAS ALLOWED THE UNCONTROLLED DEVELOPMENT OF AD HOC AND NON-INTERCOMMUNICATING DATA BASES
- A FRAMEWORK IS LACKING FOR EVOLVING THESE DATA BASES INTO AN INTEGRATED, EASY TO USE SYSTEM THAT MAXIMIZES BENEFITS FROM EXISTING INVESTMENTS

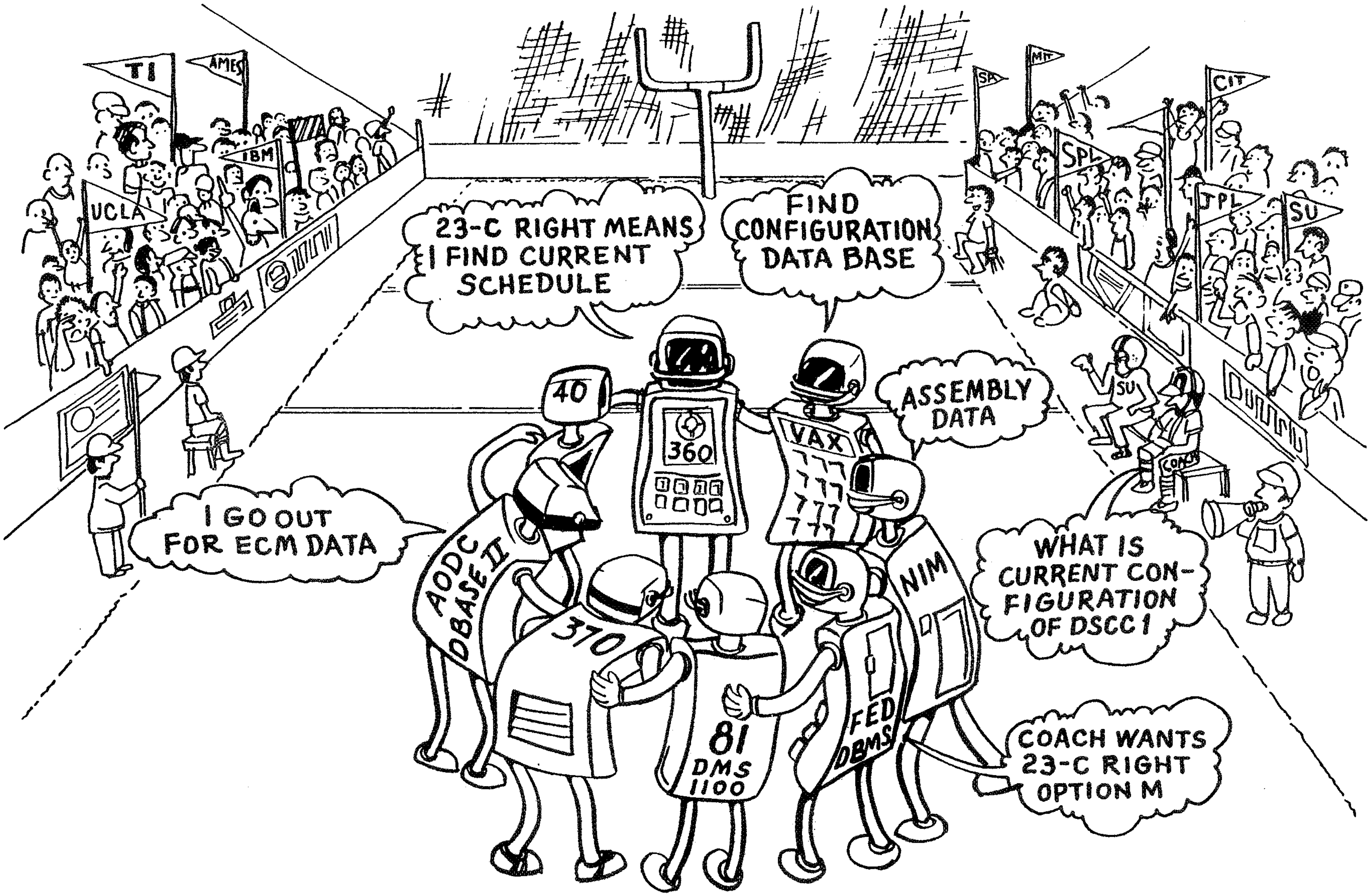
A FEDERATED APPROACH

CURRENT JPL DATA BASE SYSTEMS

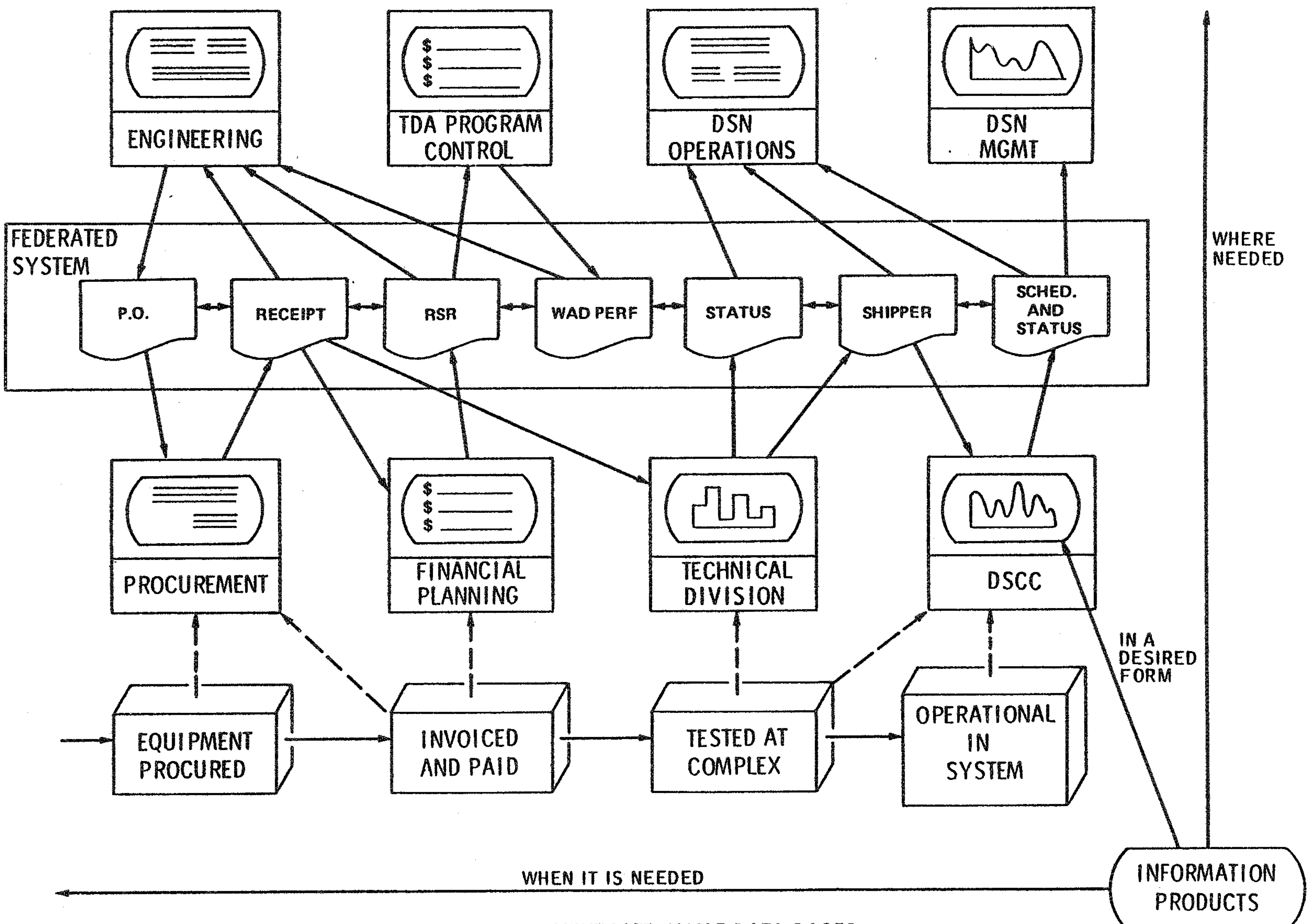
(NOT AN EXAUSTIVE LIST)

- UNIVAC 1100
  - DMS 1100
  - JPL DIS
- IBM 3032
  - SYSTEM 2000
  - MARK IV (FILE MANAGEMENT)
  - IMS AND IDMS UNDER CONSIDERATION
- DEC
  - VAX = RIM
  - VAX - INGRES (UNDER CONSIDERATION)
- MODCOMP
  - INFINITY

LONG-RANGE VISION

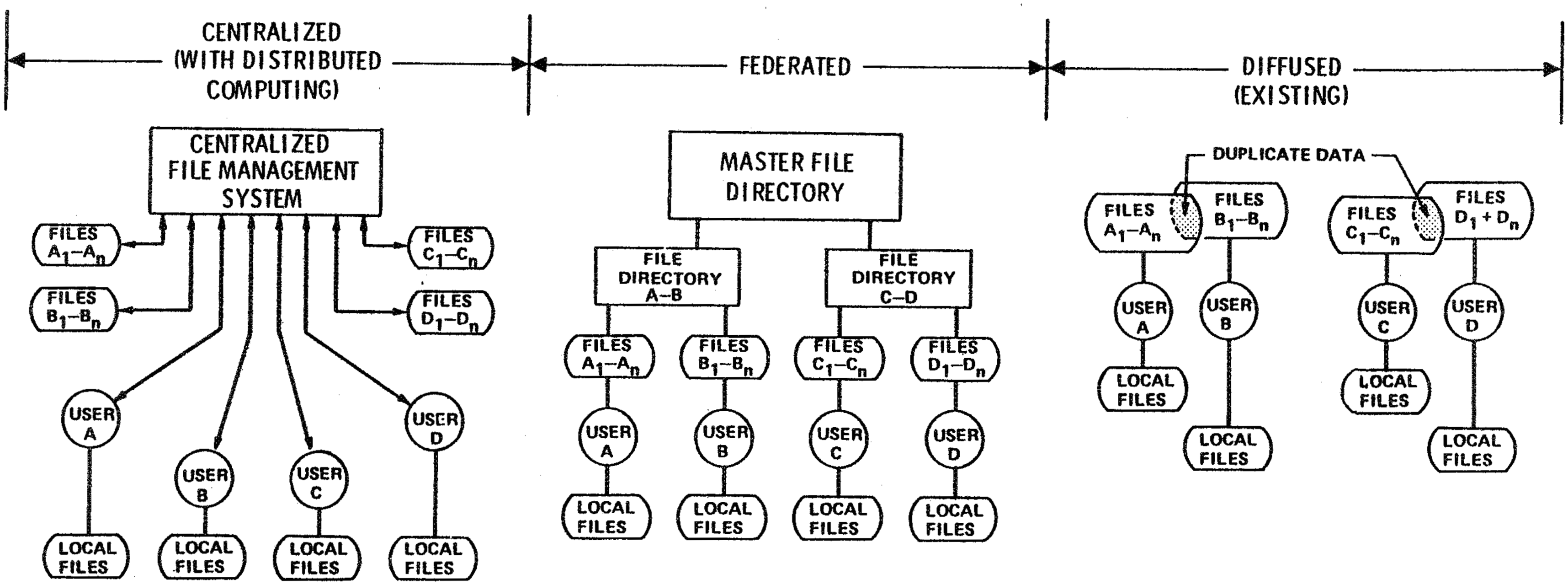


LONG-RANGE VISION



EXAMPLE OF EQUIPMENT LIFE-CYCLE DATA BASES

A FEDERATED APPROACH

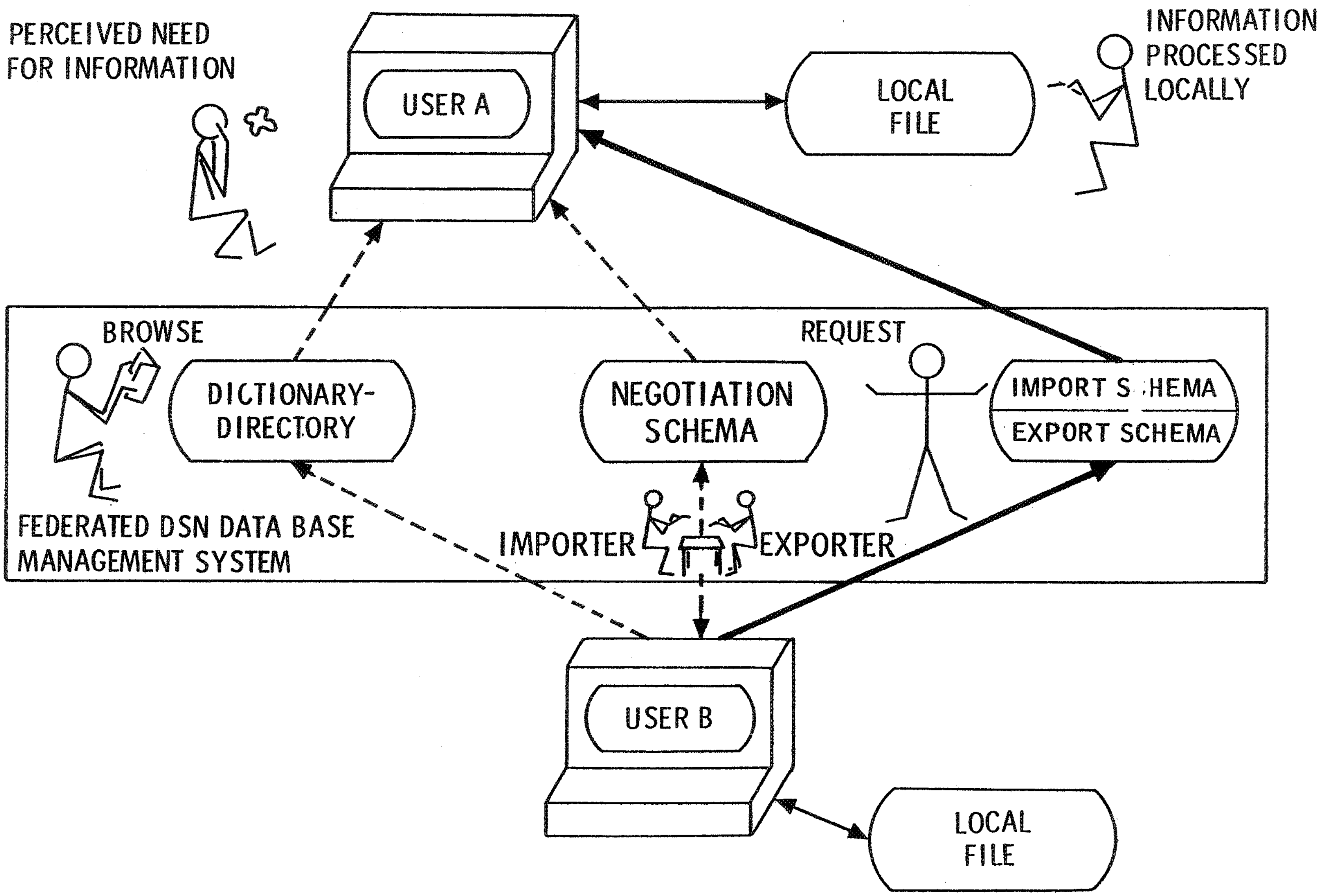


- CHARACTERIZATION OF FEDERATED DATA BASE
  - THE FEDERATED DATA BASE CONCEPT IS A SET OF PRINCIPLES AS OPPOSED TO A FORMALIZED OR RIGOROUS DESIGN METHODOLOGY
  - ALLOWS DATA BASE INDEPENDENCE AT THE LOWER LEVELS WHILE PROVIDING A ORGANIZATIONAL FRAMEWORK THAT INTEGRATES THESE DATA BASES INTO A SYSTEM
  - THE FEDERATED DATA BASE APPROACH USES A SET OF UNIFIED -
    - DATA DEFINITIONS
    - DATA FORMATS
    - ACCESS RULES

THE FEDERATED APPROACH PROVIDES A REALISTIC STRATEGY FOR EVOLVING THE CURRENT STATE OF DATA BASES INTO THE DESIRED FUTURE STATE

- THE FEDERATED APPROACH
  - IS CONSISTENT WITH PHYSICAL AND LOGICAL DISTRIBUTION OF EXISTING DATA BASES
  - RECOGNIZES THE NECESSITY TO PLACE DATA AND ACCESS TO IT AS NEAR AS POSSIBLE TO THE SOURCE OF OPERATIONAL REQUIREMENTS
- FEDERATION INITIALLY CAN BE IMPLEMENTED BY DOCUMENTATION AND OPERATIONAL PROCEEDURES RATHER THAN BY DATA BASE SOFTWARE
- DATA DICTIONARIES/DIRECTORIES CAN BE DEVELOPED AND USED ONLY WHERE AND WHEN REQUIRED
- DATA ACCESS CAN BE CONTROLLED BY THE OWNERS OF THE DATA - CAN KEEP FILES OUT OF THE FEDERATION

OPERATIONAL EXAMPLE



SUMMARY

- THE COSTS AND INFLEXIBILITY OF A GLOBAL, UNIVERSAL DATA BASE MANAGEMENT SYSTEM PRECLUDE THIS APPROACH IN MANY APPLICATIONS
- FULLY DISTRIBUTED DATA BASE MANAGEMENT SYSTEMS ARE NOT YET AVAILABLE AS A MATURE TECHNOLOGY
- THE REALITIES OF THE CURRENT AND NEAR FUTURE STATE OF THE ART, AND THE INVESTMENTS IN CURRENT DATA BASES REQUIRE A NEW APPROACH TO THE NEEDS OF THE USERS
- A FEDERATED APPROACH MAY HAVE USEFULL APPLICATIONS IN MANY, ENVIRONMENTS SUCH AS JPL's

SPACE TRANSPORTATION SYSTEM (STS)
INTEGRATED SCHEDULING AND REPORTING SYSTEM (SISRS)

PRESENTED TO THE

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE

JET PROPULSION LABORATORY
PASADENA, CALIFORNIA

MAY 26-27, 1982

Ken Martindale
Lyndon B. Johnson
Space Center

STS INTEGRATED REPORTING SYSTEM

OUTLINE

SYMPTOMS

- CONFLICTING DATES GIVEN FOR IDENTICAL STS MILESTONES
- VOLUMINOUS SCHEDULE STATUS DOCUMENTS ARE PUBLISHED MONTHLY
- PUBLISHED INFORMATION DOES NOT REFLECT WORKING SCHEDULES
- SCHEDULES DO NOT SUPPORT CURRENT FLIGHT MANIFEST

CAUSES

- MANY SCHEDULE STATUS DOCUMENTS COVER COMMON MILESTONES
- EACH DOCUMENT TRIES TO SATISFY A WIDE RANGE OF INFORMATION REQUIREMENTS
- PUBLISHING A DOCUMENT "AGES" STATUS AND SCHEDULE INFORMATION
- SCHEDULE UPDATES AND RELEASES ARE NOT SYNCHRONIZED
- FLIGHT PRODUCT SCHEDULES ARE NOT COORDINATED WITH THE FLIGHT MANIFESTING PROCESS

- STS PARTICIPANTS HAVE LIMITED VISIBILITY INTO WHOLE OF OPERATIONS
- OPERATIONS SCHEDULE INFORMATION IS MUSHROOMING BECAUSE OF LONG-LEAD INTEGRATION TIMES

SYSTEM REQUIREMENTS

- DATA ORGANIZATION
  - ADDRESSES FOUR INTERRELATED SCHEDULES FOR EACH FLIGHT
    - CARGO INTEGRATION
    - FLIGHT OPERATIONS
    - LAUNCH VEHICLE PREPARATION
    - FACILITY RECONFIGURATION
  - COLLECTS (NOT GENERATES) GEOGRAPHICALLY AND ORGANIZATIONALLY DISPERSED STS SCHEDULE DATA
- OPERATIONS INTEGRATION
  - VERIFIES AND TRACKS OPERATIONAL INTERFACES AMONG STS PARTICIPANTS (GSFC, HDQTS, JSC, KSC, MSFC, SD, USERS)
  - SCHEDULE DATA REFLECTS TOTAL INTEGRATION OF COMPLEX SET OF OPERATIONS
  - PROMOTES A COORDINATED STS PLANNING EFFORT THROUGH SCHEDULE INTEGRATION
- SIMPLIFICATION
  - COLLECTS, ORGANIZES, MECHANIZES MUCH OF REPORTING AND COMMUNICATION OF STS SCHEDULE DATA

REPRESENTATIVE SCHEDULE DATA
SISRS SCHEDULE MILESTONES
STS OPERATIONS

FLIGHT
- LAUNCH DATE
- ORBITER ASSIGNMENT
- CARGO DEFINITION
- MISSION DURATION

ANALYTICAL CARGO INTEGRATION
- PIP
- CIR

SPACECRAFT DEVELOPMENT
- PDR
- SAFETY REVIEW

INTEGRATION ANALYSES
- PIP ANNEXES
- LOADS ANALYSIS

UPPER STAGE READINESS
- ASE REFURBISHMENT
- PAM DELIVERY

STS CARRIER READINESS
- MODULE INTEGRATION
- HWD M & R

ADMINISTRATIVE CONTROL
- LSA
- OPTIONAL SERVICES
- REIMBURSEMENTS

FLIGHT OPERATIONS
- CONCEPTUAL FLIGHT PLAN
- INTEGRATED SIMS

MISSION DESIGN
- FLIGHT EVENT SEQUENCE
- CONSUMABLES ANALYSIS

CREW ACTIVITIES
- SMS RECONFIGURATION
- CAP

FLIGHT CONTROL
- I-LOAD
- FLIGHT RULES
- FDF TAPE

LAUNCH VEHICLE PREPARATION
- HARDWARE DELIVERIES
- MATING OPS

ORBITER TURNAROUND
- OIT
- CITE
- TPS

CARGO INTEGRATION
- S/C-SSUS MATING
- CITE

ET MANUFACTURE & DELIVERY
- BARGE OPS
- FABRICATION
- ASSEMBLY

SRB MANUFACTURE REFURBISHMENT & DELIVERY
- SRM SHIPMENTS
- ASSEMBLY

SSME REFURBISHMENT & DELIVERY
- SET SHIPMENTS
- REPLACEMENT CYCLES

FUEL TRANSPORT & LOGISTICS
- MANUFACTURE CAPABILITY
- SHIP DATES

FACILITIES RECONFIGURATION
- RECONFIGURATION
- ENHANCEMENT
- REFURBISHMENT

LAUNCH & LANDING FACILITY READINESS
- MLP
- RSS
- SPIF

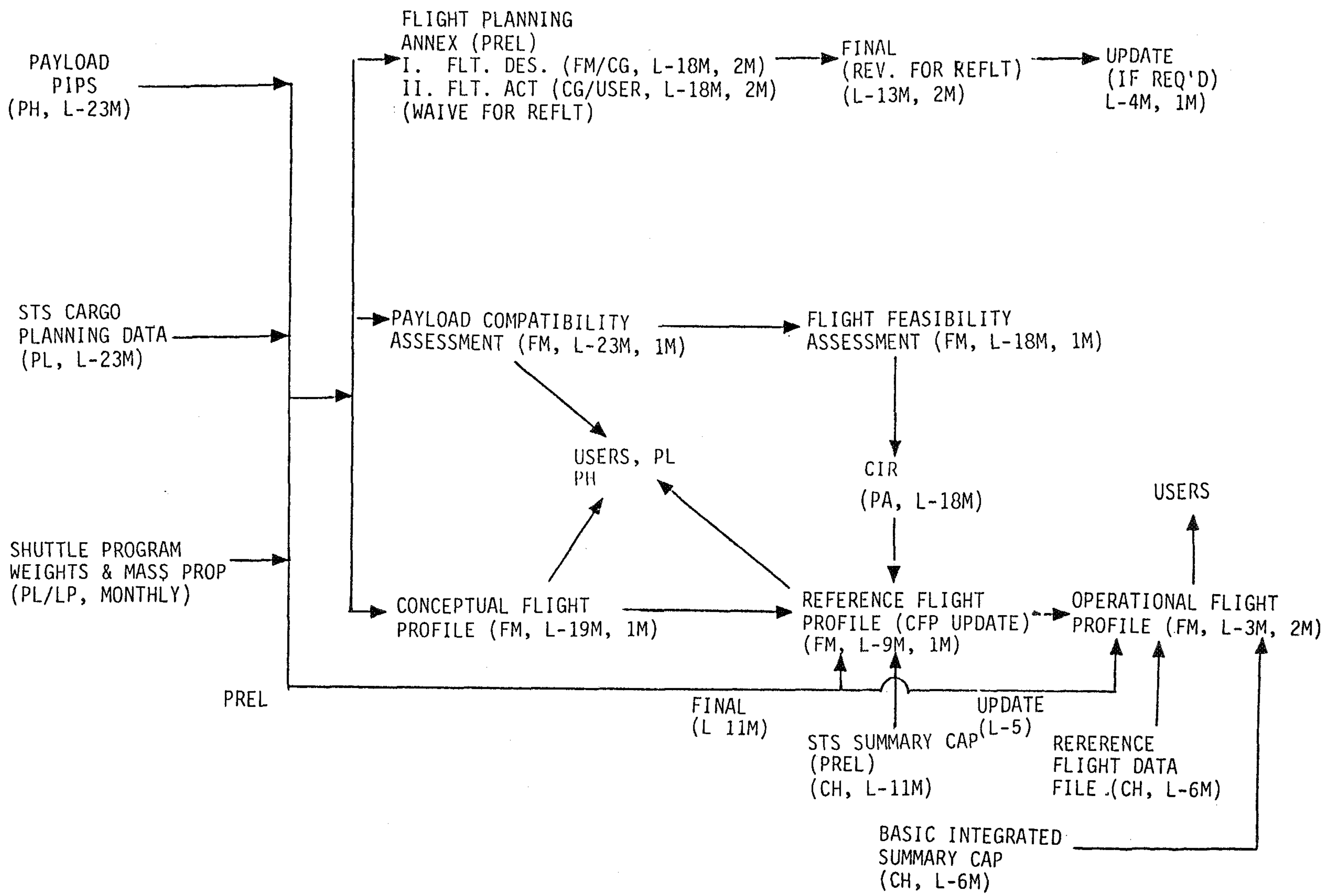
MISSION CONTROL FACILITIES
- MCC RECONFIGURATION
- AFSCF READY

TRACKING FACILITIES
- TDRSS
- STDN
- GPS

PAYLOAD CONTROL FACILITIES
- POCC
- COMSAT
- AFSCF

LOGISTICS CAPABILITY
- SPARES

FLIGHT DESIGN (MPAD) TYPICAL TEMPLATE

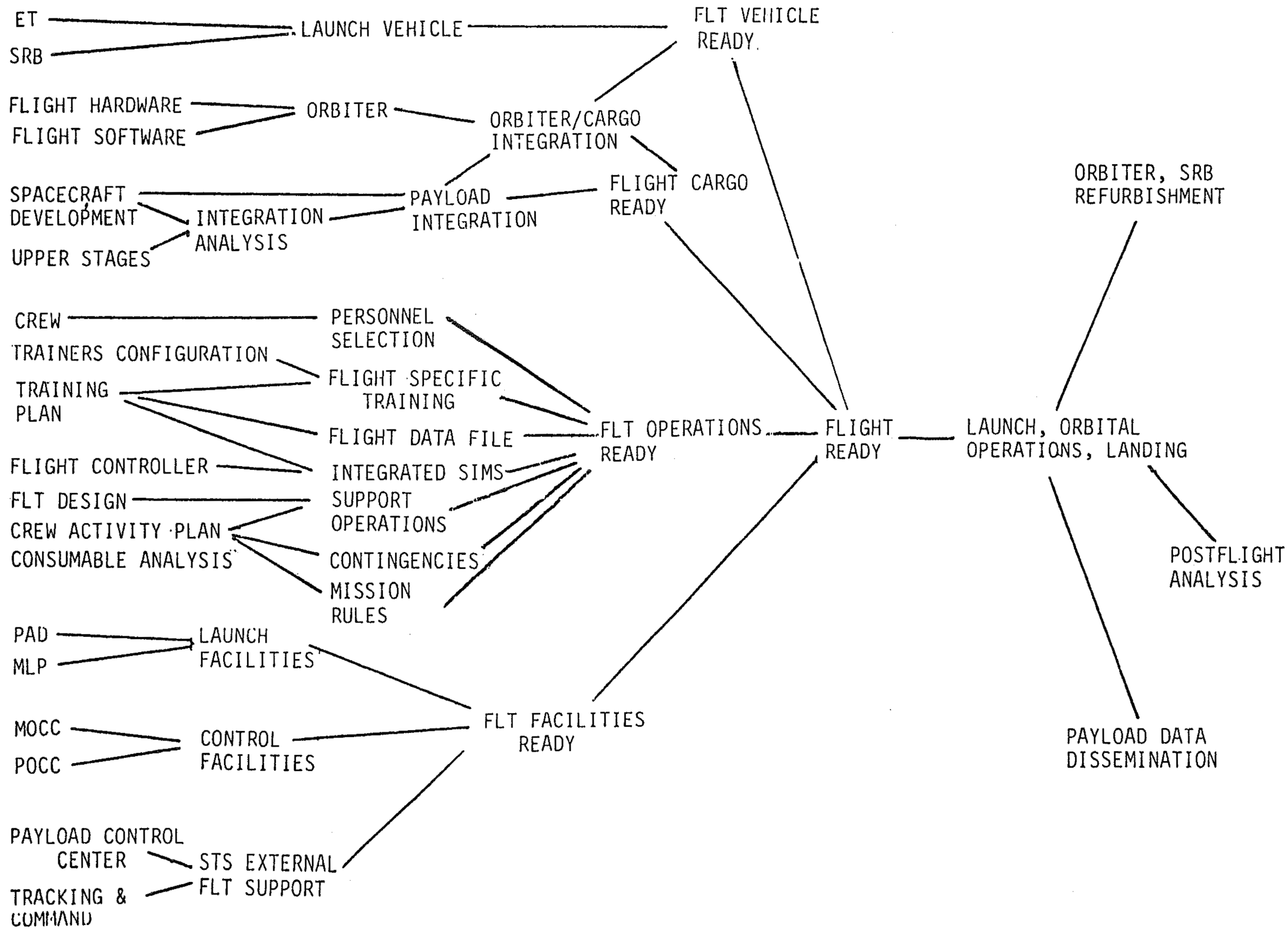


OPERATIONAL SISRS OBJECTIVES

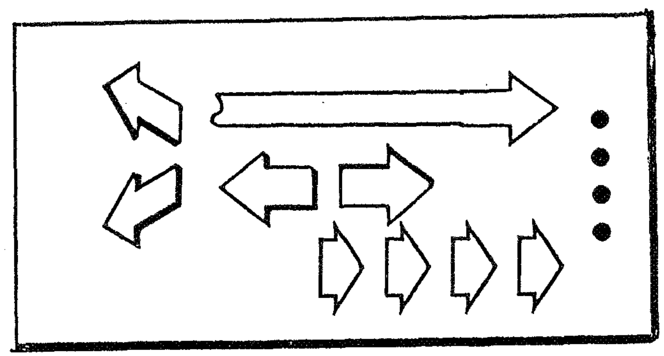
- SINGLE AUTHORITATIVE SOURCE OF STS OPERATIONS SCHEDULE INFORMATION
  - SUPPORT PLANNING, CONTROL AND OPERATIONS FUNCTIONS OF STS PARTICIPANTS (HDQTS, JSC, KSC, MSFC, SD/DOD, VAFB, STS USER)
  - FURNISH UNIFORM AND ACCURATE STS OPERATIONS SCHEDULES
  - PROVIDE DELIVERY SCHEDULES OF FLIGHT PRODUCTS
- DEMONSTRATE ABILITY OF STS TO MEET FLIGHT DATE COMMITMENT
  - CARGO INTEGRATION
  - FLIGHT OPERATIONS
  - LAUNCH VEHICLE PREPARATIONS
  - FACILITY RECONFIGURATION

STS OPERATIONS SCHEDULES

FLIGHT PERSPECTIVE



STANDARDIZED TEMPLATES



STANDARDIZED SCHEDULE TEMPLATES

DEFINITION:

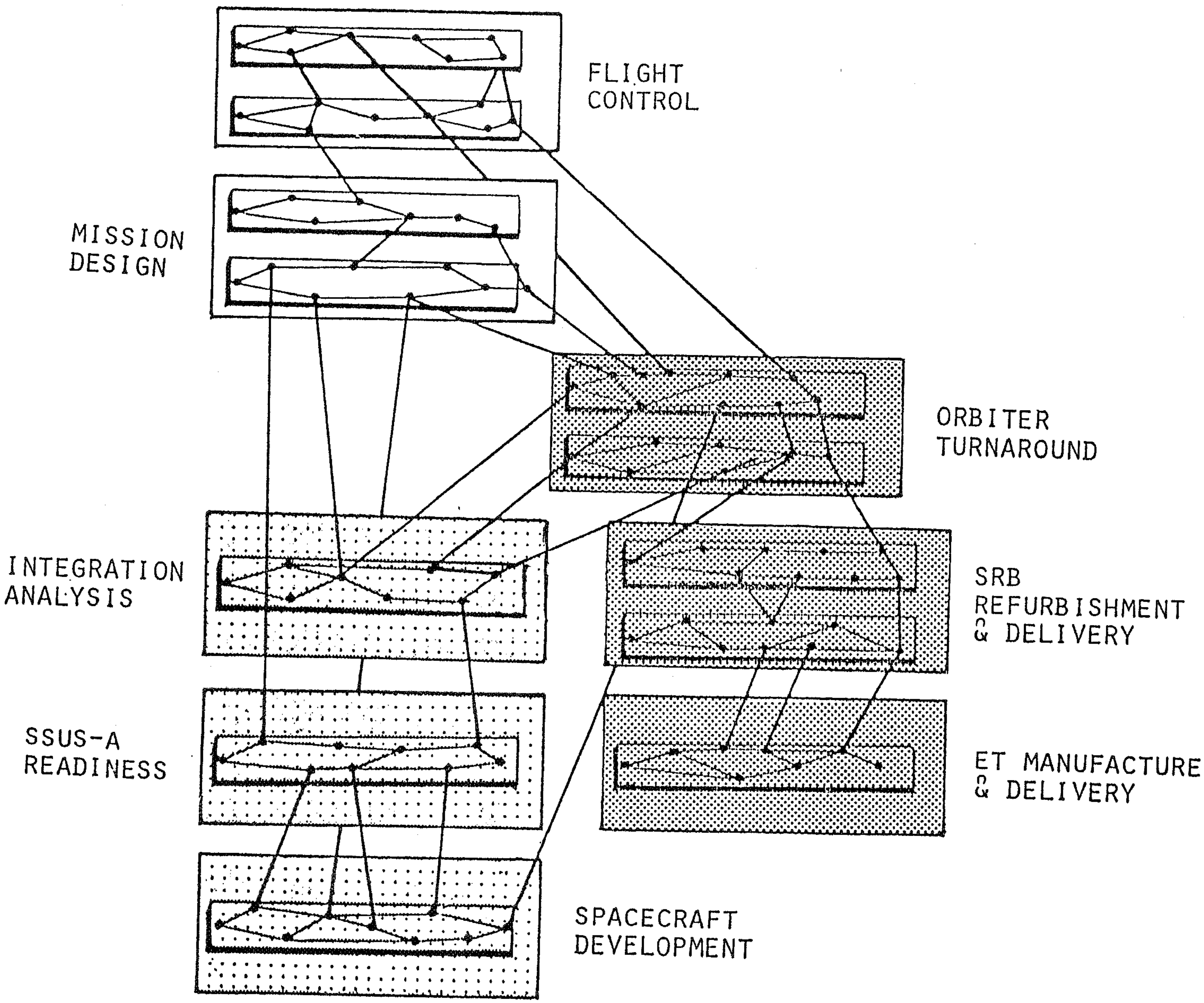
- A SET OF MILESTONES, INPUT REQUIREMENT DATES AND PRODUCT DELIVERIES ASSOCIATED WITH
- AN INTERRELATED NETWORK OF OPERATIONS CONDUCTED BY ONE ORGANIZATION (USUALLY) IN GENERATING ONE OR MORE PRODUCTS
- THE TEMPLATE MAY VARY DEPENDING ON THE PAYLOAD TYPE AND FIRST/FLIGHT REFLIGHT CATEGORY
- TIME IS REFERENCED TO LAUNCH AS "L-TIME" DATES

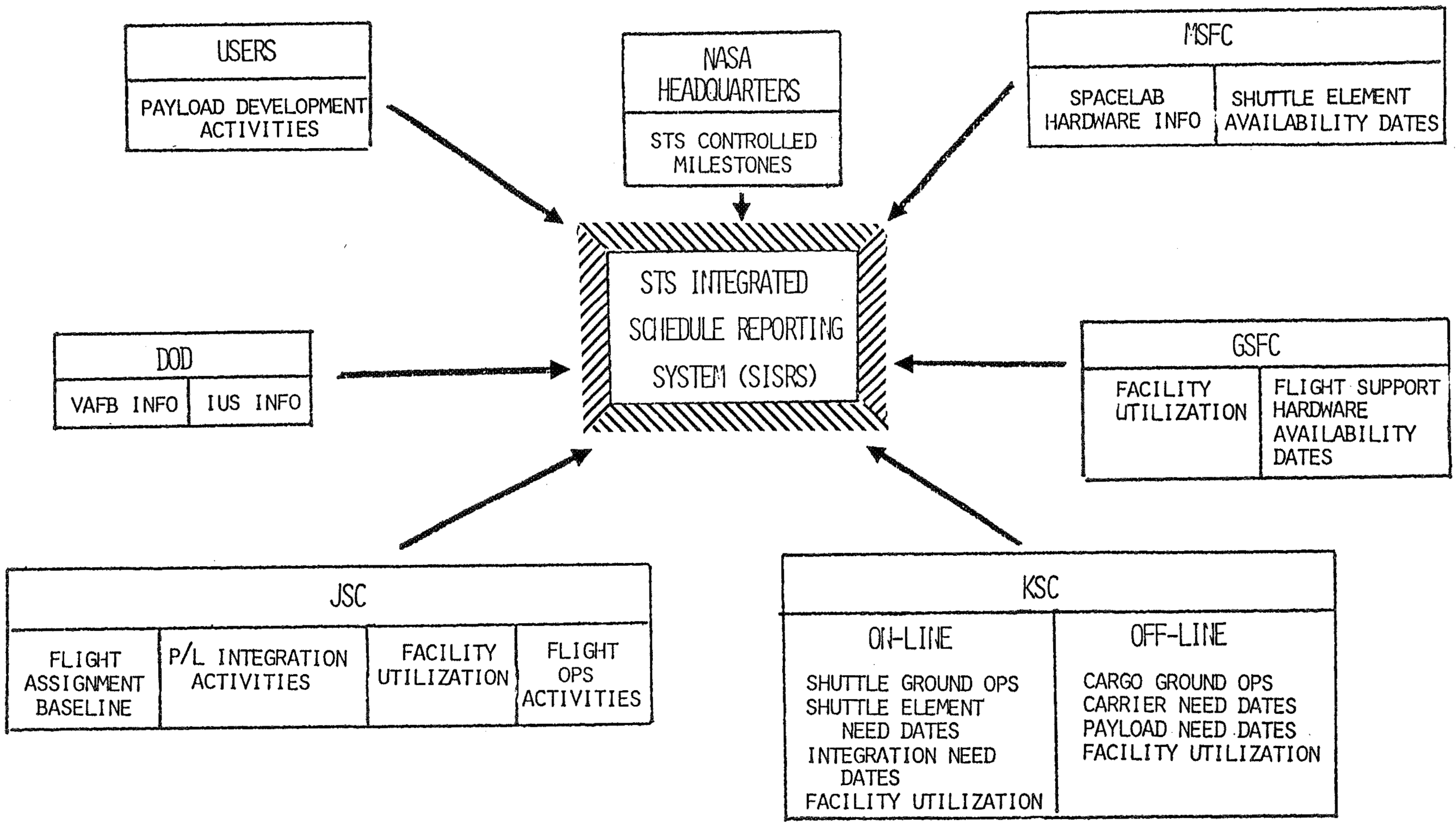
TEMPLATE BENEFITS:

- EXPLICIT APPRECIATION OF STANDARDIZATION CONCEPT
- "FIRST-OF-A-KIND" OR COMPLEX FLIGHTS MAY BE HANDLED WITH UNIQUE COMBINATION OF STANDARD TEMPLATES
- ALLOWS QUICK ADDITION OR REMOVAL OF COMPLEX INTERRELATED ACTIVITIES FROM SCHEDULE
- FACILITATES "RUN-OUT" OF SCHEDULE IMPACTS ("WHAT IF" ANALYSES)
- ALLOWS MINIMUM LOADING OF UNIQUE DATA FOR BRINGING A SCHEDULE ONLINE (SPECIFIC TAILORING IS CONSIDERED AN EXCEPTION)

- FACILITATES PROGRAM MANAGEMENT FUNCTIONS AT NASA CENTERS AND HEADQUARTERS AND STS USER ORGANIZATIONS
    - STS SCHEDULES REPORTING
        - MANAGEMENT DIAGNOSTICS
        - OPERATIONS PLANNING AND STATUSING
- RESOURCE AND WORKLOAD ANALYSIS FOR FLIGHT OPERATIONS SUPPORT BY CENTERS
- BUDGET OR MANPOWER REQUIREMENTS
- FLIGHT PRODUCT DELIVERY SCHEDULES

RELATIONSHIP AMONG SISRS TEMPLATES





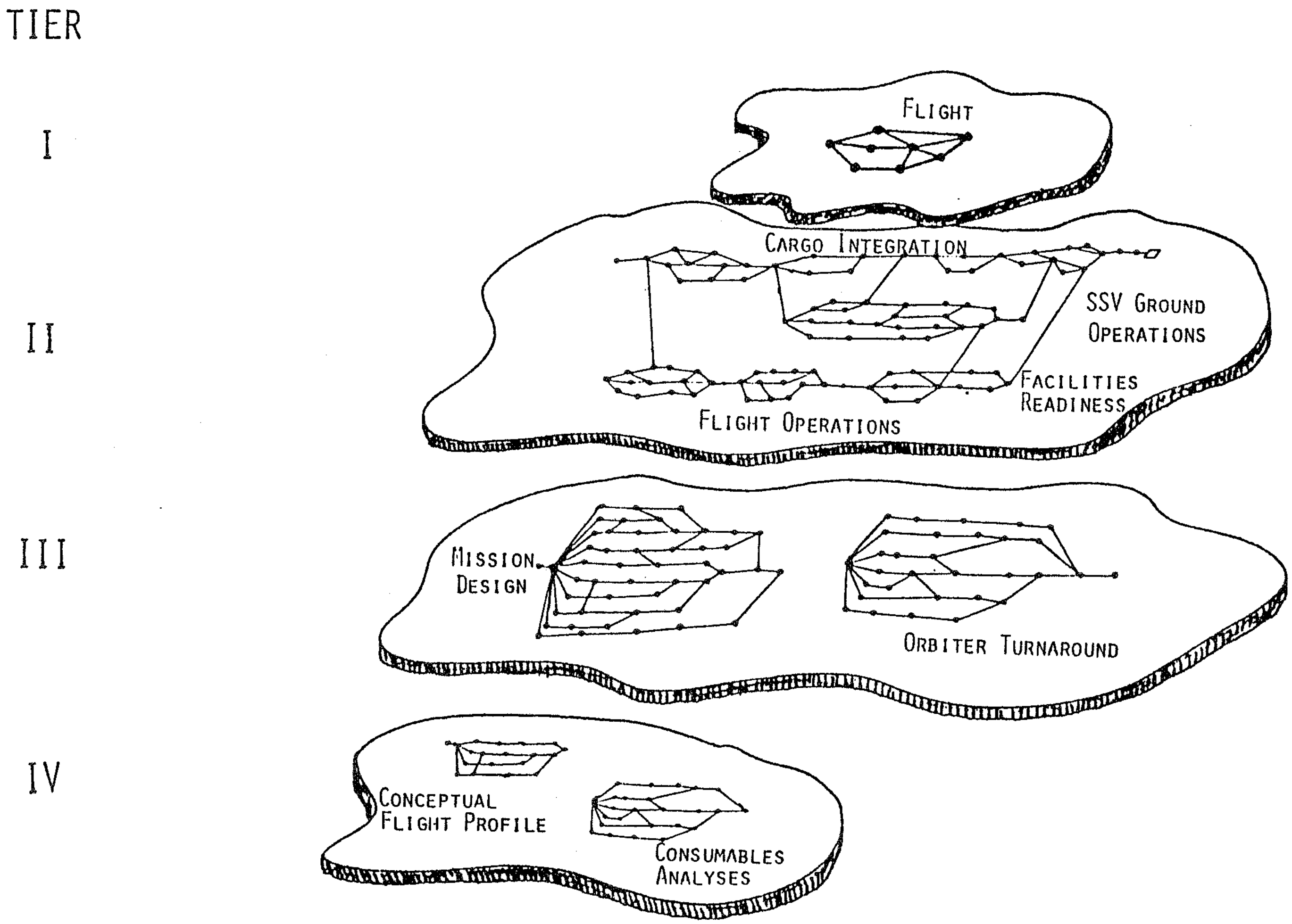
USERS
PAYLOAD DEVELOPMENT ACTIVITIES
NASA HEADQUARTERS
STS CONTROLLED MILESTONES
MSFC
SPACELAB HARDWARE INFO
SHUTTLE ELEMENT AVAILABILITY DATES
DOD
VAFB INFO
IUS INFO
STS INTEGRATED SCHEDULE REPORTING SYSTEM (SISRS)
GSFC
FACILITY UTILIZATION
FLIGHT SUPPORT HARDWARE AVAILABILITY DATES
JSC
FLIGHT ASSIGNMENT BASELINE
P/L INTEGRATION ACTIVITIES
FACILITY UTILIZATION
FLIGHT OPS ACTIVITIES
KSC
ON-LINE
SHUTTLE GROUND OPS
SHUTTLE ELEMENT NEED DATES
INTEGRATION NEED DATES
FACILITY UTILIZATION
OFF-LINE
CARGO GROUND OPS
CARRIER NEED DATES
PAYLOAD NEED DATES
FACILITY UTILIZATION

BASIC SYSTEM CHARACTERISTICS

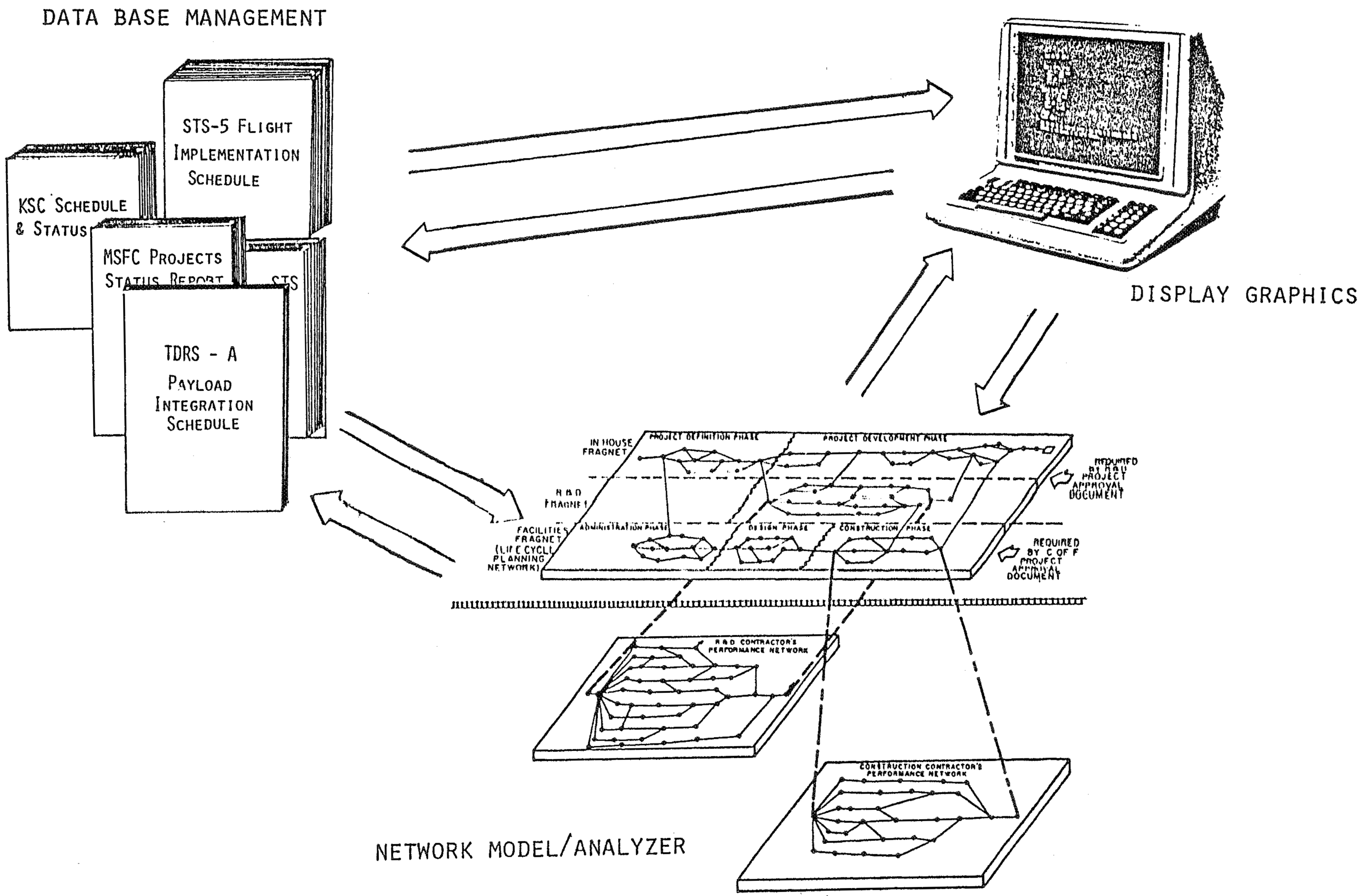
- SCHEDULE MILESTONES:
  - 35-150 FLIGHTS
  - 5 SCHEDULES
  - 5 LEVELS

  } 30,000+ MILESTONES
- USERS:
  - ACTIVE
    - PERMANENT — HQTS, JSC, KSC, MSFC, VAFB, DOD-SD
    - TEMPORARY — 25 STS USERS (PAYLOADS)
  - MODERATE
    - PERMANENT — GSFC, JPL
- ACCESS FREQUENCY:
  - DAILY STATUS REQUESTS
  - WEEKLY MASTER AND PRODUCT SCHEDULE STATUS UPDATES
  - WEEKLY TURNAROUND REPORTS
  - MONTHLY STATUS REVIEW SUPPORT
    - STS OPERATIONS REVIEW
    - OSTS MANAGEMENT COUNCIL
    - ADMINISTRATOR'S REVIEW
- REPORTS:
  - BAR CHARTS
  - TABULAR DATA

RELATIONSHIP AMONG TEMPLATES, TIERS AND STS FLIGHT SCHEDULE



BASIC SYSTEM COMPONENTS



SISRS DESIGN OBJECTIVES

- SUPPORT PLANNING, CONTROL AND OPERATIONS FUNCTIONS OF STS PARTICIPANTS
  - PROVIDE VISIBILITY TO MANAGEMENT WITH REGARD TO INTEGRATED SCHEDULING
  - INSURE COMPATIBILITY OF SCHEDULES WITH BUDGETARY PLANNING AND REQUIREMENTS
  - FURNISH UNIFORM AND ACCURATE STS OPERATIONS SCHEDULE INFORMATION
- ASSESSMENT OF STS OPERATIONS HEALTH STATUS
  - ANALYSIS OF PAST OPERATIONS
  - FORECASTING OF FUTURE PLANS AND CAPABILITY
  - PLANNING OF UPCOMING ACTIVITIES
  - REPORTING STATUS

NASA CENTER ADMINISTRATIVE DBMS ACTIVITIES

At the request of NASA headquarters, each of the centers represented at the conference prepared a brief report of their activities in the Administrative Database Systems area. These reports, arranged alphabetically by center name, are included in this section.

AMES RESEARCH CENTER

Description of Data Base Management Activities

Background:

The Administrative Applications Analysis Branch (RKM) is in the Computation Division of the Research Support Directorate. RKM's complement consists of a civil service staff of nine, reponsible for systems analysis, design and production scheduling and a contractor staff of 19 carrying out the program maintenance, development and production setup.

The Center's administrative systems had historically processed on the Centers scientific computers, on non-prime shift, until the July 81 installation of the dedicated IBM4341. Computer systems included:

IBM1401	Early 1960s
IBM7094/7040	Mid 1960s
IBM360/50	Late 1960s
IBM360/67	Early 1970s
IBM4341	July 1981

Ames administrative systems total about 35 made up of approximately 500 COBOL programs and an equal number of report generator (IRS) reports complementing the main systems.

Requirements:

An evaluation of ARC's current and future data processing needs has led to the identification and definition of requirements for improved data processing capabilities.

1. Centralized control of system data files. Establish a DBA/data base administrator position with responsibility for management of the Centers numerous data bases.

2. Programmer tools to improve efficiency of performance. With the limited funding for programming support it is necessary to get the maximum productivity possible from the programmers.

3. Direct and timely access to information. Presently, the user submits query requests to the data processing department where they are prioritized with other queries and then batch processed using a report generator.

4. On-line data entry. With the merger of the Dryden facility with ARC on-line data entry, edit and updates have become mandatory for timely operation and reporting.

Solutions:

It was determined more feasible to purchase a DBMS software package which would help accomplish the above requirements than to develop the necessary software in-house. Therefore, ARC is proceeding with the procurement of a DBMS system to meet its current and future data processing needs.

Specifications:

A long list of specifications has been developed which will meet all of ARC's requirements. The main criteria listed are:

1. An integrated data dictionary which interfaces with all subsets of the DBMS software using a common set of field labels and descriptions.

2. A "User Friendly" query language.

3. A sophisticated applications programming language executable in on-line and batch mode without change in syntax.

4. Relational data structure. One that will handle many-to-many relationships.

5. Complete language interface with COBOL.

6. Audit trail of updates and complete backup and recovery features.

Two other highly desirable features have been identified:

1. The DBMS must be capable of operating on the IBM4341 computer under VS1 in batch mode and VM/CMS in on-line operations.

   Additionally, the DBMS must operate on a DEC VAX computer.

2. It is considered by ARC highly desirable to purchase a DBMS currently in use by other NASA Centers provided it meets all of ARC's requirements. This will allow for a sharing of application software packages between Centers further reducing software and programming costs.

Planned implementation and use of a DBMS at ARC.

It is planned to eventually migrate all of ARC's business data files to the DBMS environment in order to provide to ARC management total and complete on-line access to its data resources.

Current applications will continue to operate as they do now only interfacing with the DBMS when their files are moved to the data base.

All future applications will be developed in or interface with the DBMS environment.

Implementation Plan:

1. Procurement of DBMS package

2. Install and test software

3. Establishment and selection of Data Base Administrator

4. Development of pilot system of DBMS

5. Concurrent development of new applications, update functions of present systems on DBMS

6. Development of interface with current applications with the DBMS

JET PROPULSION LABORATORY

Description of Data Base Management System Activities

INTRODUCTION

H. D. Strong
Office of Computing and Information Systems, 207

In September 1981, the JPL Computing and Information Services Office (CISO) was established. One of the major responsibilities of this office is to develop and maintain a JPL plan for providing computing services to the JPL management and administrative community that will lead to improved productivity. The CISO plan to accomplish this objective has been titled "Management and Administrative Support Systems" (MASS). The MASS plan is based on the continued use of JPL's IBM 3032 Computer system for administrative computing and for the MASS functions. The current candidate administrative Data Base Management Systems required to support the MASS include ADABASE, Cullinane IDMS and TOTAL. The MASS schedule requires the selection of a commercially available DBMS by the end of the FY-82.

Previous uses of administrative Data Base Systems have been applied to specific local functions rather than in a centralized manner with elements common to the many user groups. System 2000 has been the most widely used ADBMS with about 25 applications. Its primary use has been for the personnel and security records with a few applications in organizations responsible for financial management, telephone/ communications services, property and ADPE Inventory. More recently, limited capacity data base systems have been installed in microprocessor based office automation systems in a few Project and Management Offices using Ashton-Tate dBASE II. These experiences plus some other localized in-house DBMS uses and some offlab services such as Boeing's Executive Informations Systems (EIS) have provided an excellent background for developing user and system requirements for a single DBMS to support the MASS program.

The selection of a DBMS for MASS will be made with due consideration for its ease of use for a broad spectrum of administrative, management and technical personnel throughout the Laboratory. Is is expected that the system selected will be in use at least ten years. In order to obtain some first hand experience with the candidate Data Base Managememt Systems identified in the first paragraph, it is planned that one or more Service Bureaus will be used to try various commercially available DBMSs prior to the selection of and associated commitment of dollar resources for a DBMS for the JPL MASS program.

CURRENT ACTIVITY

The current Johnson Space Center (JSC) administrative Data Base Management System (DBMS) activity includes the utilization of Index Sequential Access Method (ISAM), Multiple Sequential Access Method (MSAM) COBOL file managers, the Resource Information Management System (RIMS) Data Management System (DMS), the United Information Services (UIS) Commercial Time Sharing System, and the Advanced Information Management (AIM) DMS. The RIMS, AIM, and COBOL file managers reside on Univac 1100 hardware and the UIS uses Control Data Corporation (CDC) hardware.

Two major applications use the COBOL file managers. These are:

(1) Interactive Basic Accounting System (IBAS) - The IBAS provides the JSC Financial Management Division with concurrent online input and retrieval capabilities to support the basic accounting function.

(2) Integrated Procurement Management System (IPMS) - The IPMS provides the JSC Procurement Operations Office with an online/batch system for collecting, developing, managing, and disseminating procurement related data supporting procurement managers, center management, and NASA Headquarters.

The RIMS DMS supports several functional areas and tasks within JSC. Some of these are:

(1) Personnel Office - Tasks supported are the Personnel Management Information System (PMIS) Ad Hoc Inquiry; Merit Pay reporting and inquiry of "model" results; status/tracking of job offers and job applications; and status/tracking of cooperative education students, upward mobility, interns, and supervisory training.

(2) Procurement Operations Office - Tasks supported are: Monthly status reports for Purchase Requests and contracts for the buying branches; award fee contract information and dates; NASA procurement regulation clauses from Headquarters and whether or not they are required, applicable, or optional; and tracking due dates for major procurements.

(3) Facilities Engineering Division - The primary task supported is the status and tracking of Facility Engineering Work Requests from initiation through the design phase.

(4) Budget Management/Preparation - Integration of directorate/office POP submissions and comparison of budget guidelines/marks against those submissions for Program Office and center management reviews prior to submission to Headquarters. Maintain division and branch level budgetary data to support the JSC POP requirements.

(5) Other tasks -

(a) Telecommunications (data) System Configuration, Inventory and Maintenance accounting, and status of installation/move/removal of terminal stations.

(b) Access production systems performance and usage data to answer Ad Hoc questions from users and management.

(c) Various status and tracking and inventory management applications.

The UIS is used primarily to prepare and analyze budget requirements at several organizational levels and to develop summary budget information for submission to the Center POP.

The AIM System has two applications. They are:

(1) Program Management and Tracking System (PMATS) - The PMATS provides the Management Services Division with status and tracking of current operations including budgets, schedules, procurement actions and contractor/civil service manpower utilization.

(2) Shuttle Avionics Evaluation Requirements (SAVER) - Allows the tracking, reporting, and modification of Shuttle and Shuttle Avionics Integration Laboratory (SAIL) measurement data.

NEAR TERM PLANS

The JSC administrative users are evolving into a more interactive environment.

Some of the existing batch systems will be replaced with RIMS and other status and tracking functions that are performed manually will be put on RIMS. The near term DBMS activity is involved in two main areas. One is investigating common file managers available on Univac to satisfy the information interchange between the financial, procurement, budget systems and the end users. GDMS's that are available commercially are not well suited for some of the unique data processing requirements; however, a common data access method would be very beneficial for data interchange.

The second area is to reduce JSC's dependence on commercial time sharing systems. In addition to a potential cost savings, a major benefit would be to have a Center-wide budget system. This would permit all levels of management to have access to budget data consistent with their work breakdown struture and still permit easy integration of data at the next higher management level.

LYNDON B. JOHNSON SPACE CENTER

Shuttle Program Information Management

System (SPIMS) Data Base

The Shuttle Program Information Management System (SPIMS) is a computerized data base operations sytem. The central computer is the CDC 170-730 located at Johnson Space Center (JSC), Houston, Texas.

There are several applications which have been developed and supported by SPIMS. A brief description of these applications follow.

- Configuration Management Accounting/Crew Systems Division (CMAS/CSD)

  Provides an automated means of managing information about changes to a large number of crew-related GFE contractor end-items. Lists all changes, their status and disposition. Contains delivery dates, costs and contract information.

- Document Index System (DIS)

  Contains an index of all documents in the JSC Library. Provides for on-line query against key word, author, document number, etc. from remote locations to support request for loan of hardcopy and microfilm material.

- Drawing List File (DLF)

  Contains an index of all drawings in the Engineering Drawing Control Center (EDCC) that were generated by NASA-JASC. Provides for on-line query against key words, engineer, drawing number, etc. from remote locations to support requests for hardcopy and microfilm material.

- Flight Manifest and Hardware Tracking System (FMAHTS)

  An automated data base for the management of the Flight Manifest which contains information that identifies, integrates and documents approved equipment to be added to the Orbiter vehicle for each Space Shuttle flight. Also contains inventory and tracking information for JSC bonded storage areas.

- Master Meaurements Data Base (MMDB)

  The MMDB provides an automated information storage and retrieval system for use as a single authoritative source of measurement/stimuli information for the Shuttle Program.

- Open Action Item Data Base (OAIDB)

  Provides an automated system for tracking action assignments resulting from Level I and II meetings such as the PRCB. Results are published in the SS Open Action Item Report ("Pink Book").

- Shuttle Automated Mass Properties System (SAMPS)

  Integrates mass properties information from contractors, and calculates data for JSC-08934, Shuttle Operational Data Book, Volume II (Mass Properties).

- Shuttle Planning and Analysis System (SPAS)

  Contains 6 major programs as follows.

  (1) NASA PERT Time II: Critical path analysis with subnetting capability.

  (2) NASA PERT Time III: Advance critical path analysis with reduced core requirements.

  (3) EZPERT: Graphic program for plotting networks, schedules (Bar and Milestone), resource graph and pre-networks.

  (4) Q-GERT: Network modeling vehicle and computer analysis tool with queing and statistical analysis capabilities for risk analysis.

  (5) AGIPLOT: Graphic program for plotting milestone-type schedules.

  (6) SISRS: Automated retrieval system for Payload Integrated Schedules and Flight Implementation Schedules. Retrieval may be tabular or graphic.

- Shuttle Requirements Traceability (SRT)

  An interactive system used for searching and tracking requirements in Level II documentation.

- Problem Data System (PDS)

  An automated system which provides a means of storing and tracking the status of problems and showing the corrective action for Level II problems encountered in Shuttle hardware and development and production. Also includes Level III Orbiter Project, GFE and GSE problems.

JOHN F. KENNEDY SPACE CENTER

Data Management Applications

DATA BASE MANAGEMENT APPLICATIONS AT KSC

Kennedy Space Center's primary institutional computer is a 4 megabyte IBM 4341 with 3.175 billion characters of IBM 3350 disc storage. This system utilizes the Software AG product known as ADABAS with the on-line user oriented features of NATURAL and COMPLETE as a Data Base Management System (DBMS).

This DBMS system was procured in August 1981 after careful review of the DBMS software market. It is operational under the OS/VS1 and is currently supporting batch/on-line applications such as Personnel, Training, Physical Space Management, Procurement, Office Equipment Maintenance, and Equipment Visibility. Some batch processing systems currently utilizing only the DBMS query capability will be restructured in the future to take advantage of the data management and exception reporting of data concomitant with the KSC mission needs.

A second application is known as the Space Transportation Accounting and Resource System (STARS) which is operational on the HP3000 configuration. This is a dedicated system utilizing the IMAGE/3000 DBMS with a mixture of batch and integrated on-line capability. The HP3000 hardware and software was procured based on the on-line transaction processing architecture required by STARS and consequently utilizes all of the interactive features of the IMAGE/3000. The STARS system is currently processing 300-400 transactions on-line each day through as many as 12 terminals with over 1 billion characters of on-line storage available.

A third and by far the largest DBMS application is known as the Shuttle Inventory Management System (SIMS) which is operational on a Honeywell 6660 (dedicated) computer system utilizing Honeywell Integrated Data Storage I (IDSI) as the DBMS. The SIMS application is designed to provide central supply system acquisition, inventory control, receipt, storage, and issue of spares, supplies, and materials. This system also provides inventory accountability and visibility with status tracking and expediting of customer requirements for inventory and equipment necessary to meet KSC mission requirements. SIMS is on-line with interactive terminals for status and update under control of the IDSI DBMS. Characteristics of the DBMS architecture are single-level and multi-level hierarchial data structure (TREES) and plex data structures (NETWORKS). The use of these data structures allows logical structuring one-to-one, one-to-many, many-to-many, and many-to-one relationships. The DBMS also utilizes a page ranging concept for management of record placement and retrieval. SIMS processes an average of over 20,500 transactions per day with an average response time of 7 seconds. In addition, over 2 million characters per hour of output is generated back to terminals. The current DBMS contains approximately 500 million data characters which does not include overhead associated with IDSI software. There are over 100 record types contained in the DBMS to define record relationships.

GEORGE C. MARSHALL SPACE FLIGHT CENTER

Data Base Management Systems Activities

TABLE OF CONTENTS

UNIVAC DATE MANAGEMENT SYSTEM-1100 (DMS-1100)

The Date Management System-1100 is designed to operate in conjunction with the UNIVAC 1100 Series Operating System on any 1100 Series computer. DMS-1100 is divided into the following four major software components:

* Data Definition Languages (DDL)
* Data Manipulation Languages (DML)
* Data Management Routine (DMR)
* Data Base Utilities (DBU)

A DMS-1100 data base is defined by a "schema" and one or more "subschemas" (actually sets of tables used by the system to satisfy requests to access the data base). The schema created by using the DDL is used to describe the various characteristics of the entire data base independent of individual applications, and subsequently, describes partial "views" of the data base by creating subschemas and limits a user's view of a data base to a subset of the areas, records and items (within records) that are described by the schema for that data base. A subschema permits a user to "see" only as much of the data base as he is authorized to see. The schema and subschema processors interpret the DDL statements and produce the set of tables mentioned above. It is important to note that the schema and subschemas comprise a definition of the data base and not the data base itself.

A Data Manipulation Language consists of commands used to manipulate the data base. These commands are embedded in high-level host languages in which the application programs are written. At present, there are DMLs for COBOL, FORTRAN, and PL/1. In addition, two End User Facilities, QLP 1100 and RPS 1100 are used to manipulate data at an even higher level than the DMLs.

The Data Management Routine is the online interface between all application programs and the data base itself. It is the principal software component of DMS-1100, that is, the set of routines which actually comprise most of the features of DMS-1100.

A Data Maintenance Utility (DMU) provides a set of terminal-oriented privileged functions to monitor and maintain the data base in an operational state. A second utility, the Data Reorganization Utility (DRU) provides the capability to reorganize an existing data base to maintain optimum efficiency throughput. DRU includes a reorganization control language to indicate the portions of the data base to be reorganized and the manner in which the reorganizations are to be done.

DMS-1100 has the following capabilities:

- Allows multiple run units to access the data base simultaneously (multithread).
- Separates data base design and implementation from programs that operate on the data.
- Provides a wide variety of storage structures to the Data Administrator, combining efficiency with the ability to model complex network organizations.
- Eliminates the need for redundant data.
- It provides a wide variety of access techniques and programming languages through COBOL, FORTRAN and PL/1 DMLs.
- It is the foundation for two nonprogrammer (end-user) facilities: The Query Language Processor (QLP 1100) and the Remote Processing System (RPS 1100).
- Provides a wide range of features to maintain the integrity of the data base, including manager levels of recovery.
- Provides mechanisms for data base security and data validation.

DMS-1100 ACTIVITIES

COMPUTER ACCOUNTING SYSTEM FOR HCC (CASH)

CASH uses an interactive, online data base. Each of retrieval and storage of data is facilitated by using the UNIVAC Data Management System 1100 (DMS-1100) software package. In addition, DMS 1100 provides the security needed to prevent unauthorized access to private information and it allows the implementation of Query Language Processing (QPL) for accessing the data base from a terminal.

CASH receives data from two large main-frame processors, one UNIVAC 1100, and one IBM 360 processor. Data for each job on the UNIVAC 1100 is written by the executive onto a raw accounting tape. At a predetermined time each day, the dialy raw accounting tapes are processed against the data base. Data from the IBM 360 is accumulated for 7 days before it is forwarded for inclusion into the CASH data base.

The data base is comprised of seven record types. The first two record types are the detail datum for production and for downtime, each of which are stored for 45 days. Whenever data for a new day is added to the data base, the oldest day's data (data added 45 days ago) is deleted. The next three record types are summary records containing production, downtime, and site summary information. These records contain separate summaries for each month in a 12 month period. A sixth record type contains information about the dates of all the detail data in the system, and it is used to arrange the detail dat in chronological order. The final record type is a summary control counter used to accumulate totals for a verification routine.

STANDARD CHANGE INTEGRATION AND TRACKING (SCIT) (IN DEVELOPMENT)

The Standard Change Integration and Tracking (SCIT) System is used to record changes requested or proposed by engineers working with the various programs at MSFC, including the Space Shuttle program (SA01), which must be processed for approval and then integrated with existing hardware and/or other changes at associated Centers and contractors.

DOCUMENT RELEASE SYSTEM (DRS)

The Document Release System provides MSFC Design Engineers with and automated technique for controlling all Engineering Document Releases. It also maintains a data base of all officially released Engineering Orders (EOs). The DRS is used for configuration control accounting by processing Engineering parts List (EPL) revisions, or EOs as configuration changes to the drawings occur. The system serves as a central repository for all engineering design data relative to the major program vehicles for which MSFC is responsible.

REQUEST FOR SERVICE AND PROBLEM REPORT (RFS/PR) SYSTEM (IN DEVELOPMENT)

The RFS/PR System provides the tracking of work task authorization and commitments by the support contractor. It also provides the tracking and timely resolution of problems within the Huntsville Computer Complex (HCC). The two sources of input to the system are the RFS/PR form that initializes a trackable record and the timecard accounting records that accumulate charges against an RFS/PR record. Information stored on the data base can provide utilization tracking by customer, functional organization, category of service, time schedule commitments, and estimates versus actual charges.

MARSHALL INFORMATION RETRIEVAL AND DISPLAY SYSTEM (MIRADS)

MIRADS is a data base management system which provides users with a variety of general purpose data management functions as well as terminal graphics capability. These functions can be performed online for immediate retrieval with demand terminals or in a batch process. A data base is first defined to MIRADS through the use of a simple data definition language. After the data base is defined, it is written into a mass storage file. The user is able to manage the data via a simple interactive terminal language.

MIRADS data management functions allow the user to select data base records using various search techniques, sort the selected records, perform computations on the data, format or print reports containing data fields from the selected records, and edit and update data fields in the selected records.

MIRADS graphics provide the user with a full range of commands for graphics data management, and for linking drawings to a MIRADS alphanumeric data base. The graphics command permits offline generation of drawings through a manual digitizing process or online generation of drawings with a graphics terminal. Online commands include the capability to delete or update portions of a drawing; to magnify or reduce drawing sizes; and to label drawings with variable size text that can be rotated.

Over 53 data bases and 26 terminal users are presently being supported by MIRADS.

NASA HEADQUARTERS

Office of Aeronautics and Space Technology Office

Automation Pilot

BACKGROUND:

A research and development project in Management Information Systems Technology involving Headquarters Codes E, R, S, and T has been defined in a memorandum signed by the Associate Administrators of the respective codes on June 23, 1981. Known as the Action Information Management System (AIMS) Project, it builds on efforts initiated within Code S in 1980. A fully developed and coordinated plan for the Code R participation in the AIMS project was not available for inclusion in the June 23 joint memorandum, although reference was made to Code R acquisition of "intelligent terminals" to be defined, and a number of Digital Equipment Corporation WS-278 communicating word processors. A Code R committee was then established to work out a detailed plan for the Code R participation under the AIMS project. This report documents the coordinated plan and initial actions to be taken by Code R in fiscal year 82.

OBJECTIVE:

Evaluate emerging office automation technologies in application to NASA Technical Program Management. An important concept of the AIMS approach is to evaluate this technology in the context of "hands on" use by technical program managers in the conduct of routine daily business transactions, and to gain appreciation of human acceptance difficulties which may accompany the transition to a significantly changing work environment. The improved productivity and communications which result from application of office automation technology are already well established for general office environments, but benefits unique to NASA are anticipated and these will be explored in detail. The following technology areas are addressed by the AIMS Project:

1. Word Processing (document creation and editing)
2. Data Base Management (information storage, retrieval, sharing)
3. Data Processing (computation, report generation)
4. Communications (electronic mail, document transmission)
5. Computer Networking

OAST enthusiastically supports the objectives of AIMS and plans to participate fully in the evaluation of the applications capabilities developed by the Project. However, OAST feels that the technology issues being explored under AIMS should encompass the broadest possible scope in order to maximize the knowledge gained in this collective learning experience. In particular, we see an important opportunity to extend the applications in several significant ways which are not addressed under the present AIMS Project. These extensions are as follows:

1. Word Processing: Application of automatic spelling checking and composition analysis programs.

2. Data Base Management: Application of a local distributed data base.

3. Data Processing: Application of stand-alone interactive financial modeling programs (VISICALC).

4. Communications: Application of microcomputers with high resolution graphics capability to real time teleconferencing. Access to technical data services provided by other agencies and commercial sources.

5. Computer Networking: Implementation of a high speed local network for distributed data and hardware resource sharing.

Of these extended applications, the local network implementation is considered to be the most significant because it represents the most advanced form of office automation known today, and will probably be an essential feature of an operational system that NASA will procure in the future. Local networking provides the means for rapid access to shared data files by collaborating offices and will relieve demand for communications services to remote computers which would otherwise become saturated as more users acquire office automation equipment.

A key issue deliberately avoided by the AIMS project to date is that of heterogeneous vs. homogeneous equipment mix. In the present state of office automation technologies, many barriers preclude the direct interconnection and information sharing among a network of diverse equipment types and manufacturers. The source of these barriers is found principally in the lack of standard data structures and interchange protocols used by the various equipment manufacturers. While it is technically possible to overcome many of these barriers through the development of specific software modules on a case by case basis, we lack the knowledge of the costs associated with so doing and the extent to which it is practical to undertake. The AIMS project has chosen the approach of using equipment of a common manufacturer in order to reduce costs and provide maximum capability in the near term. The potential benefits to be gained from intermixing heterogeneous equipment include reduced equipment costs and access to increased capabilities resulting from free market competition. These benefits can be obtained however, only by undertaking the added cost of special software development with its associated risks. OAST plans to explore this important area as a part of our participation under AIMS, and the knowledge gained will provide important information on which to base a future acquisition of an operational system.

APPROACH:

Because of the broad objectives of the OAST office automation pilot program and the desire to obtain comparative information on the capabilities and performance of alternate office automation technologies, OAST plans to initially pursue three relatively independent courses simultaneously:

1. Implement a small network of DEC WS-278 communicating word processors and DEC VT-100 terminals similar to those existing and planned within Codes D, E, and T. This equipment will be capable of communicating with the VAX computers at GSFC and other centers.

2. Implement a small network of Apple III microcomputers which will provide similar functional capability to the DEC WS-278 and VT-100, and in addition will provide high resolution graphics and stand-alone data processing through a significant body of available applications software.

3. Implement a small network of microcomputers which support the CPM operating system. This provides functional capabilities similar to the Apple III network, but through a different and larger body of available software.

The objective of this approach is to provide a vehicle to evaluate the comparative merits of alternate microcomputer technologies against the DEC homogeneous network as the established baseline. The scale of the alternative networks is chosen to be small, yet large enough that significant numbers of personnel can have daily access to the machines as a new medium to support their work activities. At the conclusion of some period of operational use of this equipment (on the order of nine months), a comparative evaluation will be made of the effectiveness and scope of capability provided by the alternate networks. Subsequent to this initial evaluation of alternate approaches, one or more will be selected for augmentation in FY 83 for operational use by a larger community of users within OAST.

IMPLEMENTATION:

The OAST initial activity under AIMS is partitioned by organization as follows:

RT	responsible for DEC-homogeneous network
RS	responsible for Apple III network
RJ	responsible for CP/M network

RT: DEC-Homogeneous Network Plan

The Research and Technology Division will implement a network of DEC WS-278 communicating word processors and VT-100 terminals to provide an early functional electronic information management system based on configurations and capabilities established within Code E. The intended applications of the equipment include general office automation functions such as word processing and electronic mail, as well as functions to aid the technical program manager such as financial planning and access to data base management systems residing on the VAX computers at GSFC. The equipment will also allow access to commercial data services such as TYMNET and AUGMENT, as well as the DoD ARPANET. These services constitute a medium for the joint authorship and review of program plans, RTOP's, and other documents by a geographically diverse technical community.

PROCUREMENTS PLANNED FOR FY 82:

- 8 DEC WS-278 Communicating Word Processors
- 4 DEC LQPSE Letter Quality Printers
- 7 DEC LA34-WA Draft (matrix) Printers
- 8 Communications Option for WS-278
- 5 DEC VT-100 terminals
- 13 BELL 212 Data Modems

Two of the WS-278 word processors and two VT-100 terminals will be deployed in Code RP.

RS: Apple III Network Plan:

The Space Systems Division will implement a small network of Apple III microcomputers which are expected to provide all of the functional capabilities provided by the DEC WS-278 and VT-100 equipment, plus additional capabilities such as self-contained computation and high resolution graphics, at lower per-unit cost. However, the ability to communicate with the GSFC VAX computers will require special software to be developed, and there is some inherent risk that the required development effort may turn out to be larger than anticipated at the outset.

The primary applications of the microcomputers in the Space Systems Division will be coordination and information exchange between Headquarters Program offices and field center technology project managers, as well as local budget planning and document preparation. The information to be exchanged will include documents, operating budgets, administrative messages, and graphics. Initial experience has shown significant advantages in using electronic mail services with field centers on the west coast as a means of overcoming the difference in local time with that at Headquarters. For example, JPL has originated program plan documents and transmitted them to a Headquarters mailbox during the JPL afternoon working hours when Headquarters has been closed for several hours. Headquarters personnel have then accessed the mailbox the following morning, edited the documents, and re-transmitted the revised documents back to the JPL author's mailbox before they report to work that morning.

A concept for real time teleconferencing has been defined which uses a network of microcomputers communicating via voice-bandwidth data circuits in conjunction with a voice conference circuit. Thus, a geographically distributed group of persons could participate in a conference using voice communications augmented by display of "electronic view-graphs" on the microcomputer screens. Appropriate software would allow real time transmission of charts from the speaker's terminal to the listener's terminals, and could support a movable cursor by which the speaker could point to items which are under discussion at any instant. There presently is no commercially available software package which supports such an application, but there appears to be substantial benefit to be gained by this technique as a substitute for travel to inter-center meetings. The Space Systems Division plans to develop a pilot demonstration of this concept to assess its feasibility and usefulness.

Space Systems Division personnel, assisted by GSFC conducted an informal survey of the microcomputer market in early CY 81. On the basis of technical capability, software availability, cost, and adequate product maturity, the Apple III was identified as the most promising candidate. Two such units were acquired via a competitive procurement through GSFC and evalauted. The evaluation has been very positive thus far, and software has recently been developed which allows the Apple III to emulate the essential characteristics of the DEC VT-100 terminal, and in the near future, the VT-125 graphics version will be emulated.

When this capability is operational, full functional operability with the other elements of the AIMS project will be demonstrated. In addition, substantial self-contained computation and analysis capability is provided through a library of available software products which include the VISICALC interactive financial modeling system and a business graphics package.

PROCUREMENTS PLANNED FOR FY 82:

The Space Systems Division has initiated action to procure six additional Apple III systems for deployment in several of the Division offices. Each Apple III will be capable of stand-alone word processing and financial analysis as well as having dial-up access to the VAX computers at GSFC and the commercial data services discussed above. Two of the Apple III systems will be deployed in Code RM.

A subsequent procurement is being prepared to perform a study of the office automation needs within OAST with the objective of defining advanced office automation concepts which promise strong benefits in the NASA working environment. An option will be requested to develop a demonstration of the concepts proposed in the study as a means of proving their validity and evaluating their effectiveness. For example, the concept demonstration may consist of linking the existing microcomputers through a high speed local network to a large central disk which will serve as the Division's working files.

RJ: CP/M Network Plan

The Aeronautical Systems Division will implement a local network of CP/M-compatible microcomputers which will provide functional capabilities similar to those provided by the Apple III network, but through a different and larger body of available software products. CP/M is a registered trademark of Digital Research Corp. and it is the name of a software operating system that dominates the 8-bit microcomputer market because of its large user population. Hence, a very broad offering of CP/M-compatible software products is available that addresses general as well as specific applications.

The nature of the aeronautical technology program indicates high potential benefit from the ability to access external data bases such as NTIS and CAB, and commercial technical literature services such as Lockheed's DIALOGUE. The primary applications of the CP/M network will include coordination and information exchange between Headquarters program managers and field center project managers, as well as local budget planning and document preparation. The teleconferencing concept described under the

Apple III network plan or a variation thereof will be explored in the context of the CP/M network. The ultimate integration of the CP/M network with the other local networks in Code R is a highly desirable goal, and will be an objective in selecting a network interchange protocol.

PROCUREMENTS PLANNED FOR FY 82:

Code RJ is presently defining the equipment requirements and specifications for the CP/M network. Procurement plans for FY 82 will include acquisition of 7 to 10 terminals, most of which will provide stand-alone computation capability in addition to communicating with remote computers and data services. Some of the terminals will be portable units capable of operating with the rest of the network. Appropriate peripheral devices such as letter-quality printers, matrix/graphics printers, modems, and disk storage units will be included in the procurement.

NASA NATIONAL SPACE TECHNOLOGY LABORATORIES

Database Management Systems Activities

Tony Cortese

CSC/INSTL

NASA/NSTL has recently acquired a VAX 11/780 utilizing DBMS-32, which is a CODAYSL compliant network type data base management system marketed by the Digital Corporation.

Currently personnel of the Date Systems Lab are involved in the requirements analysis phase of development. The primary applications which will be designed to run on the NASA/NSTL Program Support Computer System (PSCS) are in the financial management and inventory control functions.

The data base will be designed to accommodate the collection and control of all base support associated costs incurred in support of the NASA/NSTL mission.

SUMMARY OF ATTENDEES COMMENTS

At the conclusion of the conference, the attendees were asked to complete evaluation sheets. Nineteen such evaluation sheets were received. The summarized results from those evaluations follow.

SUMMARY OF EVALUATION COMMENTS

1. Most valuable features

There was a high degree of uniformity in responding to this question. Fifteen evaluators listed "Bringing ADBMS people together" and 14 listed "Knowledge gained about specific applications and ideas".

2. Less valuable features

The responses were quite varied. The most significant comments were:

Five evaluators indicated the need for a stronger conference theme or structure.
Five evaluators thought the reports were too long or too detailed.
Four respondents commented on the uneven quality of presentations.

Other less valuable features were scattered in content and received even fewer listings.

3. Should follow-up conference be held?

There was a high level of agreement on this question with 16 evaluators answering "Yes, a follow-on conference should be held". As to the time period, the vote was evenly divided between six months and 12 months. Proposed locations were GSFC, JSC, KSC, LaRC, and NASA Headquarters.

4. Future subjects and other comments

Many interesting and worthwhile suggestions were received as listed in the following tabulation:

(1) Commercially available DBMS (vendor presentations)
(2) Use of commercial versus in-house DBMS
(3) Implementation and evaluation experience
(4) Five-year HQ and center plans for DBMS applications
(5) Cost analyses of DBMS
(6) DBMS selection criteria
(7) Produce and issue DBMS newsletter
(8) More overview presentations
(9) More academic presentations
(10) Administrative topics other than DBMS
  (a) Networks and telecommunications
  (b) Application systems
  (c) Project management
  (d) User requirements/system design
  (e) Office automation

NASA ADMINISTRATIVE DATABASE MANAGEMENT
SYSTEMS TECHNOLOGY CONFERENCE
MAY 26-27, 1982

PARTICIPANTS AND ATTENDEES

ARC

David Durbin
Janet Kenrath
Michael Orlando
Doug Pearson
Mike Romiez

Don Roof (CSA)

GSFC

Ron Brunner
Jim Head

JSC

Ken Martindale
Jon Symes

Kay Burns (Rockwell)

KSC

Jimmy Akin
Claude Blackstone

Rod Smith (CSC)

LaRC

Paul Davidson
Fred Waynick

MSFC

Lowell D. Martin

Robert Kirk (Boeing)

NSTL

Tony Cortese (CSC)

NASA HQ

Ronald L. Larsen
James P. Nolan
Jim Radosevich
Sam Venneri

NASA Resident Office-JPL

Fred Bowen
Jim Bogush
Kathryn Leaman
Liz Marean
Peter Tackney

JPL

Bernie Buss (Caltech)
Nellis Adams
Joe Anderson
Barbara Carter
Elaine Dobinson
Michaelk Ebersole
Dr. Rick Green
Robert Iverson
Krista Kelly
Merle McKensie
Tav Steinacher
Dr. Joseph C. Stoltzfus
Herb Strong
Don Vincent

MITRE

Peter Rozett
Malcolm Stiefel

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CP-2254		
4. Title and Subtitle NASA Administrative Data Base Management Systems		5. Report Date January 1983
		6. Performing Organization Code NXD-2
7. Author(s)		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address National Aeronautics and Space Administration Washington, DC 20546		11. Contract or Grant No.
		13. Type of Report and Period Covered Conference Publication
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

The National Aeronautics and Space Administration Headquarters ADP Management Branch sponsored an Administative Data Base Management Systems (DBMS) Technology Conference at the Jet Propulsion Laboratory May 26-27, 1982. The purpose of the conference was to provide an open forum for exchange of information among NASA technical personnel who deal with Administrative DBMS needs. The presentation material and papers are organized in this proceedings in the sequence of their presentation, and are printed in the form provided by the authors.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Administrative Data Base Management Systems, DBMS, DBMS Selection, DBMS Development, Database Machines, Database Management System	Unclassified - Unlimited Subject Category 82

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	280	A13

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1983