DOCUMENT RESUME

ED 095 918 IR 001 103

AUTHOR	Sawtelle, Thomas K.
TITLE	The Emerging Role of the D_ta Base Manager. Report No. R-1253-PR.
INSTITUTION	Rand Corp., Santa Monica, Calif.
SPONS AGENCY	Air Force Directorate of Operational Requirements and Development Plans, Washington, D.C.
REPORT NO	R-1253-PR
PUB DATE	Dec 73
NOTE	34p.
AVAILABLE FROM	The Rand Corporation, Publications Department, 1700 Main Street, Santa Monica, California 90406 ($2.00)
EDRS PRICE	MF-$0.75 HC-$1.85 PLUS POSTAGE
DESCRIPTORS	Computer Oriented Programs; *Computer Programs; Computers; *Confidentiality; *Data Bases; *Data Processing; Information Storage; Information Systems; *Management Information Systems
IDENTIFIERS	Advanced Logistics System; Air Force; ALS; Data Base Integrity; *Data Base Management

ABSTRACT

The Air Force Logistics Command (AFLC) is revising and enhancing its data-processing capabilities with the development of a large-scale, multi-site, on-line, integrated data base information system known as the Advanced Logistics System (ALS). A data integrity program is to be built around a Data Base Manager (DBM), an individual or a group of people who provides the focal point for the coordination and control of data base functions. Data-related computing system activities serve as the basis for developing a statement of responsibilities for a DBM. The DBM can use education and indoctrination to create awareness of the role that each member of the organization plays in assuring data integrity. (Author/WH)

R-1253-PR
December 1973

The Emerging Role of the Data Base Manager

Thomas K. Sawtelle

A Report prepared for

UNITED STATES AIR FORCE PROJECT RAND

Rand
SANTA MONICA, CA. 90406

PREFACE

The Air Force Logistics Command (AFLC) currently has under development a large-scale, integrated data-base information system known as the Advanced Logistics System (ALS). At the request of AFLC, members of Rand's Information Sciences Department have been supporting the ALS implementation in specified technical areas. One of these areas has been the development of a data integrity program.

This report serves as an introduction to the problems and requirements of data integrity in an ALS-type system. It contains guidelines for the establishment of a data integrity program built around a new organizational element, the Data-Base Manager. It is assumed that readers of the report have a basic familiarity with data-processing concepts and terminology.

SUMMARY

The Air Force Logistics Command (AFLC) is revising and enhancing its data-processing capabilities with the development of a large-scale, multi-site, on-line, integrated data-base information system known as the Advanced Logistics System (ALS). This report serves as an introduction to the problems and requirements of data integrity in an ALS-type system. More specifically, it contains guidelines for the establishment of a data integrity program built around a Data Base Manager (DBM).

As ALS-type information systems become larger and more complex, they are increasingly sensitive to the ways in which their data bases are created, maintained, used, and protected. Inasmuch as these data-base functions involve people throughout the entire organization, there must be a degree of centralized coordination and control over these functions. In this report, that coordination and control--and the associated organization, charter, and procedures--are collectively defined as a *data integrity program*.

The key element in a data integrity program is a new organizational element, often referred to as the Data Base Manager. The DBM is an individual or a group of people that provides the focal point for the coordination and control of the data-base functions. This concept is a relatively recent development in the computing field and--although general agreement exists on the need for a DBM--there are many approaches to the questions of its responsibilities, organization, and staffing.

Section II discusses data-related computing system activities as the basis for developing a statement of responsibilities for a DBM. Two questions are addressed: Which activities should be within the purview of the DBM? What should be the degree of the responsibility in each case?

Section III presents alternative approaches to creating a DBM. One issue is that of placing the DBM within an organization. The nature of the DBM's role requires (1) that the lines of communication and authority be sharply drawn, regardless of the position in the organization; (2) that there be firm, consistent management support. The

DBM will normally be established as the system becomes operational. However, many of the DBM's activities begin during the system design and development phases. With prior planning, much of the work done during these phases will be useful to the DBM. In addition, there is a natural transition of people from several of the specialized development groups (e.g., those responsible for data conversion) into the DBM staff.

Although differences of opinion exist as to how the DBM work should be organized and whether staff members should be specialists or generalists, there is agreement on the personal characteristics required for success. These include substantial data-processing experience, excellent communications skills, and a working familiarity with user operations.

As an essential element in an ALS-type system, the DBM will continue to evolve in terms of responsibilities, organization, and staffing.

A data integrity program must provide for establishing and maintaining an awareness of the role that each member of the organization plays in assuring data integrity. Section IV discusses how this condition can be attained through a program of education and indoctrination that begins during system development and continues throughout the life of the system. The DBM can play an important role in creating and maintaining this awareness.

ACKNOWLEDGMENTS

Several organizations with large-scale, complex information systems were visited during this study effort. Information gathered during these visits was drawn upon extensively in preparing this report. The author would like to thank the following persons who generously shared their time and experience:

- Edward Ferris, Milt McElhaney, and Joe Simpson, American Airlines
- Alphonse E. Frei, Mutual Computer Services, Inc. (a subsidiary of Continental Airlines)
- Wil n Grafton, North American Rockwell Corporation
- Rob t S. Wakefield, Southern Pacific Company
- Joseph Williams, Western Airlines
- James B. Woods, Hughes Aircraft Company

The author would also like to thank Robert L. Patrick for his many valuable suggestions, and Rand colleagues N. Donald Cohen, Gerry Levitt, and Marco Fiorello for their careful review and comments.

CONTENTS

I. INTRODUCTION

THE NEED FOR A DATA INTEGRITY PROGRAM

Developers, managers, and users of computer-based information systems have spent a great amount of time and energy improving the technical performance of their systems. Although the track record of technical achievement has not been perfect, there have been steady gains.

In contrast, the organizational and procedural aspects of computer-based information systems have received relatively little attention. As a result, many of the most serious problems are not technical, but rather have to do with organizations and procedures [1]. So much concern has been put into making the system work from a technical standpoint that these other aspects have not received their proper share of attention. Systems have failed to perform countless times because:

1. There has been a person-to-person communication failure. For example, the user informally requests modifications to the logic of programs that process his data. As a result of the lack of a standard method of communication, the programmer misunderstands the request and codes incorrect logic into a program.
2. Inadequate or incorrect procedures have been used. Data files have been destroyed, or at least degraded, innumerable times because backup procedures have either been nonexistent or have been inadequate to cover the particular situation. Another problem has been the absence of validation procedures and criteria, resulting in incorrect computer output being generated and released. Tables required by computer programs are improperly updated or incorrect versions are used because procedures are not adequate.
3. There has been a failure to "human engineer" some part of the system. An example is the poor form design that leads to an unacceptably high input error rate in the data entry process [2].

4. Responsibilities for error resolution have not been clearly assigned.

The above list of typical system problems growing out of procedural and organizational shortcomings is merely representative.

This report will present an approach that addresses many of the organizational and procedural problems related to data integrity.

SIMPLE VERSUS COMPLEX INFORMATION SYSTEMS

Computer-based information systems come in all sizes and shapes, and can be classified in many different ways, depending on the characteristics that are chosen to describe them. Three data-related characteristics that are useful in classifying a system in a discussion of data integrity are (1) the update mode (batch vs. on-line), (2) the access scheme (sequential vs. random), and (3) the degree of functional integration* (if any) in the data base.

Considering these characteristics, we find on one hand the simple batch-oriented system using sequential files and serving a single functional user. Typical examples are Air Force Logistics Command (AFLC) data systems that are largely built around magnetic-tape files. At the other extreme is the complex on-line system organized around functionally integrated data bases--e.g., the Advance Logistics System (ALS), under development by AFLC, with its on-line updating of massive random-access data bases. (See Appendix.) In between these extremes are systems with varying degrees of complexity. (See Table 1.)

Experience gained with complex systems, particularly the on-line, integrated data-base variety, has demonstrated that many aspects of the organizations and procedures that were developed to administer and operate simple systems are inappropriate. There is a growing awareness of the sensitivity of an entire system to the way in which the data base is created, maintained, used, and protected [5]. It becomes apparent that a carefully developed effort is required to coordinate

---

*Functional integration refers to the combining of data from two or more functional areas, such as personnel records and accounting, into a single data base [3,4].

Table 1

SYSTEM COMPLEXITY

System	Characteristics: Update	Access Mode	Data Base Functional Integration
Simple (AFLC today)	Batch	Sequential	No
.	.	.	.
.	.	.	.
.	.	.	.
Complex (ALS)	On-line	Random	Yes

these functions across the entire organization. This effort and the resulting organization, charter, and procedures may be considered a *data integrity program.*

The potential costs of a failure to maintain the integrity of the central data bank are great in terms of time, dollars, and degraded performance; and the issue of data integrity deserves both early and continuing attention.

Anyone who has lived through the trials and tribulations of designing, implementing, and operating an ALS-type system is well aware of the magnitude and complexity of the task of assuring data integrity in the data base. In addition to presenting guidelines for the establishment of a data integrity program built around a Data Base Manager, this report serves as an introduction to some of the problems and requirements of data integrity in an ALS-type system for those who have not had such an experience.

Section II contrasts the complex, ALS-type system with simpler systems in terms of key data-base related activities and associated implications. Section III considers issues associated with the creation of a Data Base Manager. Section IV discusses creating a system-wide awareness of accountability for data integrity.

II. DATA-RELATED ACTIVITIES IN SIMPLE AND COMPLEX INFORMATION SYSTEMS

This section contrasts simple and complex systems, as defined above, to illustrate the need for coordination and control of the various activities. Obviously, however, the comments apply to all computer-based systems, and the need becomes greater as the complexity of the system increases [6-9].

Following is an examination of a number of key system activities in terms of the requirements they impose on the design, implementation, and operation of an ALS-type system.

DATA-ELEMENT DEFINITION

Data-element definition is the process of identifying the attributes of an item of information (data element), such as "part number" or "salary." Some of these attributes are data-element size, recording mode, format, and relationships to other data elements.

In simple systems, data-element definition is usually the responsibility of a programmer/analyst working in conjunction with the user. Both the initial definition and subsequent changes are coordinated and implemented by these parties.

Data-element definition in an ALS-type system can be, and usually is, a totally different activity. A data element and its attributes may be of interest to a number of users from different functional areas. These users may have conflicting requirements, resolvable either through compromise or the establishment of additional data elements. For example, the individual files of payroll and personnel records departments may each contain a data element known as "employee address." The address in the payroll file may identify the location to which all payroll-related correspondence should be sent; the personnel file, the employee's legal address. If information from these files is integrated, this difference will have to be resolved as described above.

The nature of an ALS-type system requires that changes in data-element definitions be made quickly and with minimum impact on users.

This condition affects the approach to incorporating data-element definitions into user programs. In simple systems, the definitions are usually written directly into each program; whereas in the ALS-type system, the definitions are often maintained in a system-control table and are available for use as required throughout the system. A central coordination and control point for data-element definition is essential. The nature and degree of control may vary, but the function must exist.

DATA-BASE STRUCTURE DEFINITION

Data-base structure definition is the process of selecting a method or methods for organizing data elements that satisfies both the user's requirements for logical relationships among the data elements and the computing system's requirements regarding physical organization.

Sequential processing and a single primary user make structure definition relatively straightforward in most simple systems. In an ALS-type system, however, new issues make this activity much more difficult. The structure is affected by questions of storage-device characteristics, access methods, data-management system capabilities, and response-time requirements. Again, coordination is required to insure effective structure definition--both initially and as the system changes.

DATA-BASE CREATION AND INITIAL CERTIFICATION

Data-base creation and certification involve the construction, to design specifications, of a data base, usually from some combination of automated and manual sources. This task is often a difficult and perilous part of implementation. Some particularly critical tasks in an ALS-type system are:

1. Coordinating the process of selecting from among multiple sources of data when preparing to create an integrated data base.
2. Analysis of the source data to determine their suitability, a complex task involving not only the validation of specific data values but also questions of time phasing, intra-record

consistency among data elements, and validation of the source selections.

3. Establishing criteria for the analysis and certification, a difficult task that often brings to the surface deficiencies in the system design. It requires substantial attention and coordination during both the development phase and system operation.
4. Planning and coordination of any clean-up required in preparation for the data-base creation. This task, requiring participation of system designers, developers, and users, is unfortunately often of low-visibility.

DATA-BASE UPDATE

Updating is the process of applying current information, in the form of transactions, to a data base, causing the creation of new records or the modification/deletion of existing records. In an ALS-type system, updating is concerned with such questions as:

1. Which program(s) is authorized to update each data element?
2. For a given data element, is the same program authorized to perform all of the update functions (add, delete, modify)?
3. How is the authority to update controlled, with respect to both operational control and changes in authority?
4. Under what conditions can updating take place? For instance, is updating allowed during only certain times of the day?
5. How are program changes coordinated with associated changes in procedures, personnel assignments, documentation, etc.?
6. How is background processing of the data base coordinated with on-line updating to insure proper time-phasing?

Implicit in these questions is the need for a degree of control and coordination.

TRANSACTION EDITING

Transaction editing may be broadly defined as the process of examining input transactions and accepting or rejecting them, based on previously established criteria. Editing is often done through a combination of manual and automated procedures.

This process is normally straightforward in simple systems, involving one edit program, the batching of transactions generated from a single source, and a relatively long error detection/correction/resubmission cycle.

In an ALS-type system, on-line input from many sources (and the short error-processing cycle inherent in this type of operation) requires the careful development of input procedures and the thorough training of terminal operators--as well as very complete edit logic in the transaction-processing programs. To the extent that the data base serves multiple users, the definitions of edit criteria must be coordinated. In addition, continuous monitoring of the editing process will help to identify problems in input procedures, edit logic, and/or system operation before they can seriously degrade the data base.

DATA-BASE QUALITY ANALYSIS

Quality analysis involves continuing and/or periodic analysis of the data base to insure that it satisfies the requirements of the system specifications with respect to both structure and data values. Because the potential for problems is great in an ALS-type system, a coordinated approach is required. Some key considerations are:

1. What is the value of an error-free data base? Can some level of data-base degradation be tolerated, and how should it be measured? Degradation can be measured in terms of both incorrect data values or flaws in the data-base structure.
2. What kind of auditing techniques should be used in the analysis process [10]?
3. What are organizational responsibilities for:
   a. Specifying evaluation criteria?*

*Probably the same or similar to criteria used both in the analysis of source data in preparation for data-base creation and in transaction editing.

b. Selecting tools and techniques and developing procedures to be used in the data-base quality analysis? Specifying, designing, and developing software required to support the analysis?
c. Training and supervising those responsible for the analysis?
d. Conducting the analysis and coordinating corrective actions?

DATA-BASE ACCESS CONTROL

Access control must protect against all unauthorized access, whether deliberate or inadvertent.

Access control in the typical tape-oriented simple system has not been a major problem. Physical control of magnetic tape(s) and certification of programs authorized to process specified files have normally sufficed.

However, significant new problems of access control are introduced by the ALS-type system. The on-line nature of the system makes the data base potentially accessable by each system user, subject only to the effectiveness of protective features built into the hardware and software, and operational constraints [11].

An ALS-type data base may be more attractive to the deliberate penetrator for two reasons. First, functional integration of data often creates new information in the sense that the data were not previously available for coordinated processing. Second, the centralization of data may decrease the exposure the penetrator must risk. A determined penetrator can usually breech a system thought to be secure [12].

The development and operation of an access-control plan requires the coordination of system designers, developers, operators, and users.

CHECKPOINTING, RESTART, AND DATA-BASE RECOVERY

The three closely-related activities of checkpointing, restart, and data-base recovery are among the most critical from the standpoint of data integrity [13,14]. On-line operation utilizing direct storage devices creates many new sources of potential trouble. For instance, operating systems are typically much more complex than those in simple systems, increasing the probability of software-related failure.

Checkpointing is the "snapshot" recording of such pertinent information about a computer system as the contents of computer memory, message queues, the state of the operating system, and the status of data-base processing. Checkpointing may also refer to the recording of the entire contents, or a selected portion, of a data base. Checkpointing helps to protect against unexpected problems in the functioning of a computing system.

Checkpointing techniques are governed by, and influence, such considerations as software efficiency, the allocation of data bases to storage devices, and data-base size and activity.

Restart is the process of reestablishing the status of the computing system as it existed at the time of an interruption, planned or otherwise, and resuming system operation. Checkpointing supplies the information needed in the restart procedure. Restart, which often has been a troublesome activity in simple systems, is a major source of concern in the ALS-type system. Error analysis and resolution are usually difficult, complex procedures must be developed and followed, and actions coordinated among several parties.

Data-base recovery is the restoration of a data base after it has been damaged, either physically or logically. This task is extremely demanding in an ALS-type system, involving both system users and operators. The data-base processing characteristics of continuous update and update-in-place require elaborate procedures to assure recovery capability [15]. In fact, data-base recovery is a major design consideration in the development of an ALS-type system.

MAINTENANCE OF CONTROL TABLES

ALS-type systems make extensive use of control tables of various types. Some of these tables are used to control system functions and operations. Examples are tables containing the characteristics of communication lines and input/output devices.

Other tables of control information are application-oriented. Typical tables of this type contain information on transaction identification and validation, data-base characteristics, and update authorizations for data-base maintenance.

Because the impact of changes in these tables is so far reaching, centralized documentation and control of table maintenance is essential. Responsibility for table content should remain with the appropriate group, but all changes must be coordinated.

DATA-BASE ACTIVITY MONITORING

Activity monitoring involves the on-going recording and analysis of data-base maintenance and usage. Typical items of interest include the number of transactions processed per unit of time and the activity ratio (the ratio of records accessed to all records).

There are several reasons for the establishment and coordination of this data-base activity monitoring in an ALS-type system: (1) it assists in the continuing validation of application design, and often provides a basis for program modification and redesign; (2) it provides input into hardware review and reconfiguration; (3) it is essential in the allocation of data bases to the most appropriate storage medium; (4) it supports periodic review of the adequacy of data-base structures and provides a basis for restructuring.

APPLICATION PROGRAM TESTING, CERTIFICATION, AND CONTROL

Application programs must be tested for (1) concurrence with design specifications, (2) successful integration with other programs, (3) conformance to constraints imposed by the operating system, and (4) such other considerations as programming standards and impact on overall system performance. This testing is required in both initial program development and subsequent modifications. Successful testing should be followed by a formal process of certification, after which the program becomes part of a certified program library. The establishment and control of such a library substantially increases the quality of system performance.

Program testing, certification, and control are required in all computing systems. But the nature of the ALS-type system greatly increases their importance. Typically, the responsibilities for these activities are spread throughout the organization. However, they are interrelated and must be coordinated.

The above comments are at least equally applicable to user-developed system software. Software programs become extremely complex and highly interdependent in an ALS-type system. This complexity, combined with more stringent response-time requirements and the potential for rapid propagation of errors, makes centralized testing, certification, and control essential [15]. Vendor-supplied software must also go through a process of certification and control.

SYSTEM UTILITIES

The design and development of program tools to support many of the activities described above is a major requirement in an ALS-type system. Although various utilities--e.g., sort, merge, and dump programs--are normally available from the vendor, it is often necessary to modify them to meet special needs. In addition, it is usually necessary to develop unique utilities for the particular system. Examples of the latter are special-purpose data-base display and modification programs. Coordinated design and development of these programs is highly desirable.

TRAINING, STANDARDS, AND DOCUMENTATION

Each of the activities discussed above requires (in varying degrees) consideration of training, standards, and documentation. They are mentioned separately to emphasize that an ALS-type system is particularly sensitive to careful training, the creation of positive and practical standards, and the establishment and maintenance of appropriate and usable documentation.

* * * * *

The above discussion of data-related activities illustrates two points:

1. Data integrity in an ALS-type system is complex in terms of the problems encountered, the procedures and resources required, and the skills needed.

2. An ALS-type system imposes broad accountability for data integrity in both development and operation of the system.

These two factors--increased complexity and broad accountability--have led to the development of a new organizational element, as ALS-type systems have been implemented, that provides a focal point for the issues and problems described above [7,16]. This individual or group may go by any of several names--Data Base Manager and Data Base Administrator are the two most frequently used. For consistency, Data Base Manager (DBM) is used in this report in reference both to the person(s) involved and to the office.

Section III examines some current approaches to the Data Base Manager as the cornerstone of a data integrity program.

III. CURRENT APPROACHES TO THE DATA BASE MANAGER

ANALYSIS OF CURRENT APPROACHES

Although general agreement exists that a Data Base Manager (DBM), or the equivalent, is needed in ALS-type systems, such specific issues as a charter of appropriate activities, when the DBM should be established, its placement in the organization, and the size and skill requirements for its staff are being approached in a variety of ways.

In gathering information on the Data Base Manager concept, a number of companies with information systems containing existing or evolving data integrity programs were visited.* These companies were selected for their similarity to the planned ALS. The selection criteria were:

1. Common data banks of at least several hundred million characters;
2. Terminal orientation;
3. On-line update;
4. High transaction rate;
5. Operational for at least a year.

ROLE OF THE DATA BASE MANAGER

No consensus has evolved on the subject of a charter of activities, or role, for a Data Base Manager [17]. Substantial variation is found in the activities of existing DBMs, resulting from such factors as personnel capabilities; preexisting organizational responsibilities, and the relative maturity of a system.

How then can the role of a DBM for a particular system be defined? One approach is to answer two questions. First, for which activities will the DBM be responsible? As noted above, there is no standard

---

*North American Rockwell (Downey, Calif.), Hughes Aircraft Company (Fullerton, Calif.), Western Airlines (Los Angeles), Continental Airlines (Los Angeles), American Airlines, (Tulsa, Okla.), and Southern Pacific Company (San Francisco).

answer to this question. The list of data-related activities in Sec. II provides a good starting point. Though not exhaustive, this list covers activities that usually concern a DBM. The second question deals with the *nature* of the DBM's responsibility for each of the activities. Each activity embraces a spectrum of potential responsibility, with decisionmaking power at one extreme and administrative/clerical functions at the other. The role of the DBM, therefore, can be outlined in terms of (1) the activities he is responsible for, and (2) the nature of his responsibility for each of the activities.

The DBM (or equivalent) in each company visited was analyzed in terms of whether or not it performs the activities listed in Sec. II; and if so, the nature of its responsibilities. There is general agreement with respect to some of the activities (Table 2). However, *all* of the activities listed were performed by one or more of the DBM's. As would be expected, the highest levels of responsibility are associated with those activities directly related to the data base.

Table 2

DATA-BASE MANAGER RESPONSIBILITIES

Activity	Nature of DBM Responsibility			No Consensus
	Decide	Coordinate	Administer	
Data-element definition		X		
Data-base structure	X			
Data-base creation and certification	X			
Data-base update				X
Data-base activity monitoring				X
Data-base quality analysis			X	
Data-base access control		X		
Checkpoint, restart, and data-base recovery	X			
Maintenance of control tables	X			
Transactions editing				X
Application program testing, certification, and control				X
System utilities	X			
Training, standards, and documentation		X		

FITTING THE DATA BASE MANAGER INTO THE ORGANIZATION

Different approaches have been followed by managers of ALS-type systems in positioning the Data Base Manager organizationally. Figures 1, 2, and 3 are edited organization charts for three such systems.

A number of factors have combined to account for these different approaches to organization:

1. The considerable variance in the *total* set of responsibilities from one system to another affects where the DBM is located organizationally. For example, if the thrust of responsibility is toward user service, the DBM tends to be associated with the group that has the most user-orientation--often applications programming.
2. DBM assignments tend to migrate toward groups containing individuals that have both technical strength and strong interests in such activities.
3. The preexisting structure of the organization, and the management's willingness to change, also affect the DBM's location.
4. The DBM is an evolving group whose location within the organization may change as the system matures.

Despite the differing approaches to organizational placing of the DBM, it is important to remember that in each of the companies the need for a DBM has been empirically recognized and, within the constraints and considerations described above, a new organizational element has been created.

PLANNING FOR THE DATA BASE MANAGER

None of the companies visited had had a DBM or its equivalent when their systems began operations, yet each soon found such a function necessary. Developers of ALS-type systems can profit from this common experience by planning for a DBM during the system development phase.

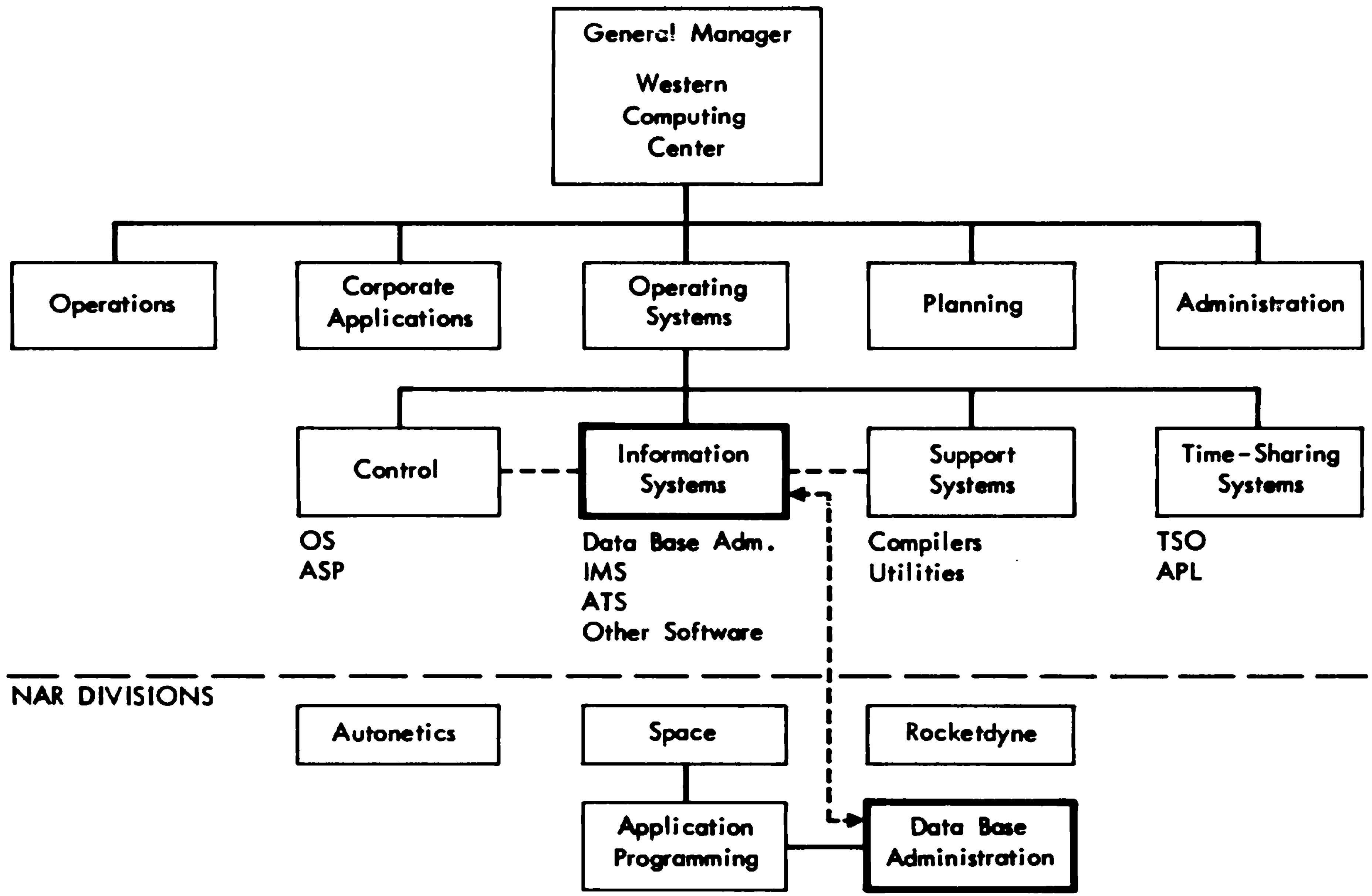


Fig. 1 — Organizational chart, North American Rockwell (now Rockwell International)

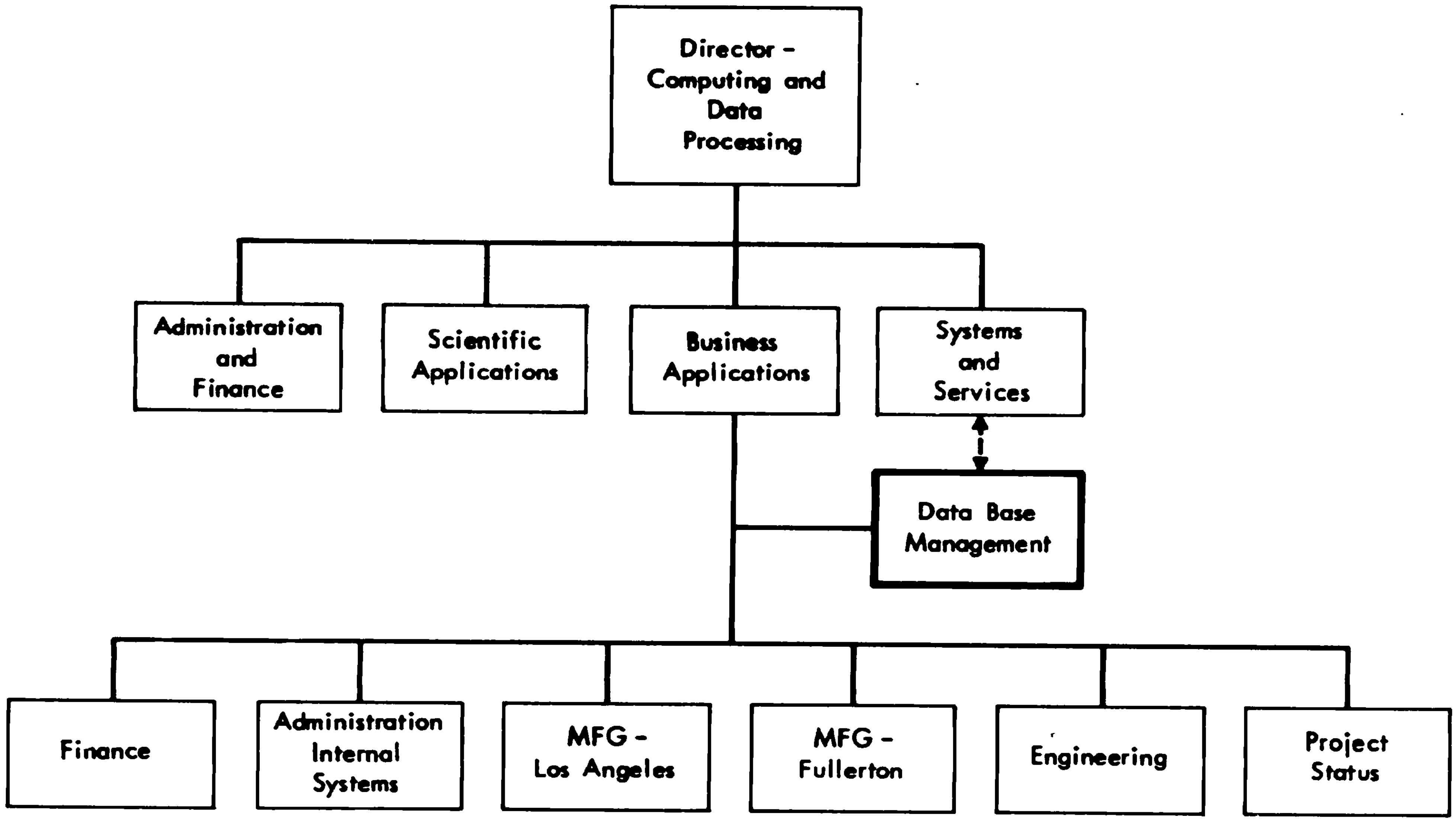
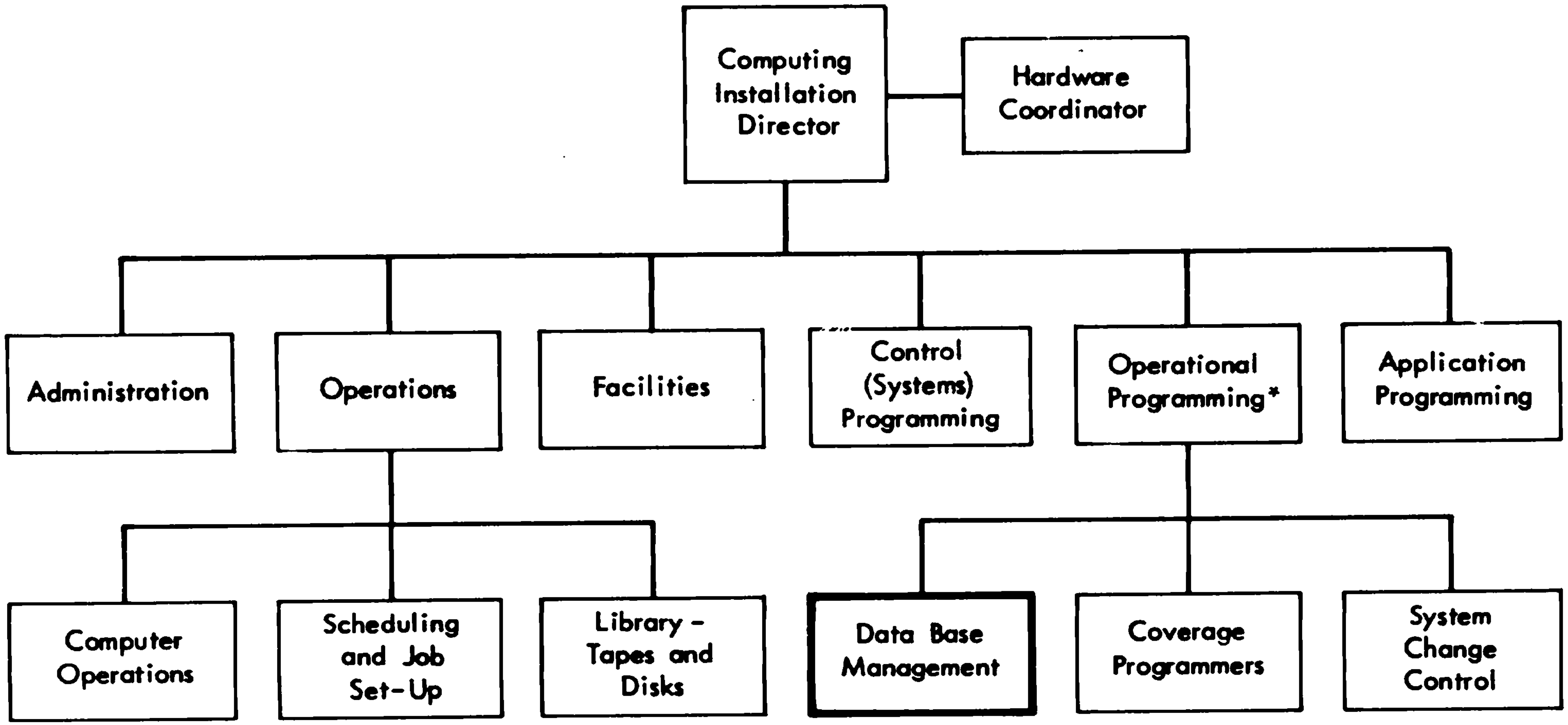


Fig. 2—Organizational chart, Hughes Aircraft Company



* Some aspects of the Data Base Manager are found in each section of Operational Programming

Fig. 3—Organizational charts, American Airlines

Typically, the organization that develops an ALS-type system finds that several of its groups either disappear or take on somewhat different roles after the system is implemented. These groups include system testing and integration, data conversion, and initial data-base design and structuring. Their work is the logical antecedent of many of the activities that may be assigned to a DBM. Most of the DBM activities listed in Sec. II begin, in one form or another, during the early stages of system development.

Thus, it is logical to plan for the establishment of a DBM, as the foundation of a data integrity program, early in the development process. This planning should cover people, procedures, and special programming.

- *People and Procedures*--The gradual transition of the people in the groups mentioned above into a DBM is a natural process. For example, after the integrated data base is successfully loaded, a data-conversion group as such is no longer needed. However, there is a need for people with data-conversion type experience to effectively deal with such requirements as data-base checkpoint and recovery. Data-conversion people are normally concerned with these subjects during the conversion planning and implementation. By the same token, many of the procedures developed for use in system implementation can, with proper planning, have a high degree of usefulness after the system becomes operational.
- *Special Programming*--With a reasonable amount of extra planning, much of the special programming written to support system implementation can be used in an ongoing data integrity program. An example is the development of standard test cases. With a modest extra effort, the test cases employed during system development can be standardized for continuing use after the system becomes operational. Another example is the data-conversion programs written to load the data base

and to deal with exceptions occurring during the loading. These programs can be designed for direct use, or easy modification and use, by the DBM in data-base recovery.

In summary, the foundation for the DBM can and should be laid during development and implementation so that its formal establishment is essentially an acknowledgment of an accomplished fact.

In planning for the DBM, two other observations are useful:

1. Inasmuch as the DBM has heavy interface responsibilities among different elements of the computing organization and the user community, it is extremely important that clean lines of communication and authority, both formal and informal, be established.
2. As with any group whose responsibilities cut across those of other groups, it is essential that the DBM have firm and consistent management support. The DBM will absorb responsibilities that have been traditionally held by other groups within the organization, and clear management support is essential for DBM success.

STAFFING CONSIDERATIONS

Organizations with Data Base Managers generally agree on the characteristics of a successful DBM staff:

1. Experience: Typically, members have from one to six years experience in data processing--the average being two to three years. These data-processing personnel normally come from either applications programming or computer operations groups. However, people from user organizations have also been successfully integrated into DBM staffs.
2. Communication skills: The heavy responsibilities of interfacing at many levels and among diverse groups places a premium on communication skills for the DBM staff. It is essential that they be able to convey accurately their ideas both orally

and in writing. At least as important is sensitivity to the import of the words and actions of others. These communication skills also imply tactfulness in dealing with high-stress situations.

3. The DBM staff must also have had at least an exposure to user operations. One of the keys to success is the DBM's ability to understand user problems and needs in the user's vocabulary.
4. The DBM staff should be relatively stable. High turn-over is potentially a major problem, as much of the work done by the DBM may be viewed by the data-processing staff as system-maintenance work and, as such, not particularly attractive or stimulating. Data Base Managers have therefore found it necessary to develop ways of continuously motivating their people. One way has been to assign more responsibility than is normal at the DBM staff salary level. This tactic requires extreme care in selecting people, with the result that DBM staffs tend to be very high quality. Another way in which DBM people have been motivated is by the granting of special privileges. An example is providing them with a special terminal having privileged access to the data base and other capabilities not generally available to the programming staff. Yet another means of insuring the stability of the DBM staff is to encourage the development of automated tools and procedures that remove much of the drudgery from the work. Though not necessarily cost-effective, such procedures make the work more attractive.
5. DBM personnel should be of the type who thrive on problems. People vary tremendously in their abilities to work effectively in the face of uncertainty. Such ability should be one of the selection criteria in assembling a DBM staff.

Most DBM staffs number from five to ten people, including the supervisor. The number appears to be somewhat independent of the size and complexity of the system, the number of programmers involved, and (to a large degree) the number of applications supported.* Rather, staff size appears to be a function of range of responsibility.

Two totally different approaches have been followed in organizing the work of the DBM staff. One involves the development of a team of specialists, each one responsible for an activity such as data-base design, data-base language, data communications, system definitions (table maintenance, user coordination), procedures, training and forms, and standards development and enforcement. The other approach has been to develop a staff consisting of broadly trained members, each of whom, while tending to work in one or two areas, is capable of working at an acceptable level in all the areas of responsibility. Both approaches appear successful. The key seems to be the supervisor's ability to motivate the staff rather than the manner in which responsibilities are assigned.

THE FUTURE OF THE DATA BASE MANAGER

As more experience is gained with ALS-type systems, it is reasonable to expect increasing agreement on the role of the DBM. The nature of such systems--the complex data integrity requirement and the associated broad accountability--will lead to increased responsibilities for DBMs. This growth in responsibility will in turn affect the organization, size, and type of staff.

In the long run, however, it is not clear that the DBM will retain its identity as its role continues to develop and grow. The DBM may be but an evolutionary feature in the way complex information systems are managed. It is possible, for instance, to conceive of the DBM being incorporated into a System Manager or Administrator who would be separate from the traditional programming, analysis, and operations group, and who would be responsible for the coordination and varying degrees of control of *all* operational system activities in much the same way as the DBM is responsible for data-related activities.

---

*In one case, the number of applications supported doubled over a period of a year-and-a-half with no corresponding increase of the BDM staff.

IV. EDUCATION/INDOCTRINATION FOR DATA INTEGRITY

The development of a pervasive awareness and acceptance of the broad data-related accountability discussed above is necessary in a data integrity program for an ALS-type system. Many people in computing system organizations are not accustomed to "thinking data integrity," and a data integrity program should address this problem. The key to success is the involvement of all personnel through their understanding of the impact of their work on the data base.

This effort should begin during system development and implementation. Specific actions that might be taken include:

1. A series of indoctrination seminars covering such subjects as data-base structuring, description of system-control tables and their uses, programming standards, and data conversion. These seminars would be coordinated by the development manager and conducted by the appropriate groups. The seminars would emphasize the interrelationship among the many activities, and might be given at several organizational levels.
2. A newsletter that would reinforce and document the seminars and provide a forum for constructive comments and suggestions.

The education/indoctrination effort should continue after system implementation. New people will be joining the organization, and there will be new developments of interest to the entire staff. At that point, responsibility for the effort may logically be assigned to the DBM even though the dissemination of information not directly related to data integrity may be involved. Specific actions could include:

1. A tailored indoctrination program for new people that would include an overview of the system with emphasis on their particular roles and a reading program to bring them up to speed in appropriate areas.

2. Continuation of the seminar series, with the emphasis shifting to new developments.

The success of this effort requires the commitment of management and the resulting willingness to allocate resources to carry it out. The mechanisms for implementation suggested above are not obviously "productive" in the normal sense, and there would be great temptation to bypass or devitalize them under pressures of time and/or budget limitations.

V. CONCLUSIONS

Information systems organized around large, integrated data bases require a high degree of coordination and control of data-related activities. Developers and operators of such systems have found that an effective way of accomplishing this has been the creation of a Data Base Manager/Administrator (DBM).

At this time there appears no optimum location for the DBM within an organization. Several successful approaches have been followed by companies that have implemented DBMs. The most important factor in placing the DBM is the over-all thrust of its responsibilities.

In the case of an information system under development, it is possible, with modest additional efforts in many of the activities, to develop the people, procedures, and program tools that will eventually constitute the DBM. This is particularly true with respect to activities such as data conversion and data-base creation.

The DBM function requires people with the somewhat unusual combination of strengths in both technical areas and interpersonal relationships. Inasmuch as most DBM staffs have substantial interaction with system users, familiarity with user operations is another key requirement.

A continuing program of education and indoctrination for data integrity is an essential ingredient in the operation of a complex system. Such a program may be administered by a DBM as part of its overall concern with the issue of data integrity.

Data Base Managers are playing essential roles in the operation of ALS-type information systems. As noted in this report, many questions and alternatives must be considered in the establishment of a DBM. The exact nature of the specific responsibilities and the authority assigned to a DBM may vary from system to system, but central awareness, coordination, and control of data-related activities are vital.

APPENDIX

BRIEF DESCRIPTION OF THE ADVANCED LOGISTICS SYSTEM

The Advanced Logistics System (ALS) represents a substantial redesign of the AFLC Logistics management. When fully implemented, the ALS will be built on one of the largest computing systems to date. There will be six large- and one medium-scale installations, each located at a major logistics center. These installations will eventually replace most of the computing equipment presently used by the Command.

A key feature will be the creation and maintenance of massive data bases at each processing site. Information from many logistics functions will be integrated into a relatively few data bases organized around logistical processes, thus making possible the development of management information previously unobtainable and the elimination of much of the data redundancy inherent in present systems.

The data bases will be collectively referred to as the Unified Data Bank (UDB), one at each ALS site. The UDB design will be standardized, although the contents for each site will be unique.

Much of the ALS processing will be done in real time or near-real time, in contrast to the sequential-file, batch-input mode characteristic of present AFLC data systems. Transactions and queries will be input from remote terminals throughout the Command by AFLC personnel, with the system supporting interactive operation from any of the terminals.

REFERENCES

1. Cattaneo, E. R. (ed.), "Key Issues in Data Base Management Are Non-Technical," *Data Processing/Computer Digest*, Vol. 7, No. 6, June 1972, pp. 1-2.

2. Staggs, Earl W., "Maybe It's Your Forms," *J. of Sys. Man.*, March 1972, pp. 8-12.

3. Cooke, Milton J., "The Data Base Revolution," *Sys. and Procedures J.*, March/April 1968, pp. 20-22.

4. Nicholson, Charles H., Jr., "Building Data Banks for Multiple Uses," *Sys. and Procedures J.*, May/June 1968, pp. 18-22.

5. Canning, R. A., "Using Shared Files," *EDP Analyzer*, Vol. 8, No. 11, November 1970, 14 pp.

6. Lyon, John K., "The Role of the Data Base Administrator," *Data Base* (ACM Special Interest Group on Business Data Processing), Winter 1971, pp. 11-12.

7. Canning, R. A., "The Data Administrator Function," *EDP Analyzer*, Vol. 10, No. 11, November 1972, 14 pp.

8. Erwin, D. B., *User Experience in Data Base Administration*, Proceedings of Guide 34 Conference, May 1972, pp. 316-338.

9. Timerman, Jules, "Managing the Data Base," *Data Processing Magazine*, Spring 1972, p. 7.

10. Haber, S. E., F. W. Segel, and H. Solomon, "Statistical Auditing of Large Scale Management Information Systems," *Naval Research Logistics Quarterly*, Vol. 19, No. 3, September 1972, pp. 449-459.

11. Hirschfield, R. A., "Security in On-Line Systems--A Primer for Management," *Computers and Automation*, September 1971, pp. 15-17, 25.

12. Canning, R. A., "Data Security in the CDB," *EDP Analyzer*, Vol. 8, No. 5, May 1970, 14 pp.

13. Yourdon, E., "Approaches to Error Recovery," *Modern Data*, June 1972, pp. 38-46.

14. Canning, R. A., "Restart and Recovery," *EDP Analyzer*, October 1968, 10 pp.

15. Grafton, W. P., "Data Base Recovery with IMS/360," *Data Base* (ACM Special Interest Group on Business Data Processing), Vol. 4, No. 1, Spring 1972, pp. 9-12.

16. Eddy, R., and R. Ramm, *The Role of Data Base Administration (DBA)*, Proceedings of Guide 34 Conference, May 1972, pp. 224-230.

17. Stieger, W. H., "Data Administrator Functions," Proposed statement of functions prepared for the CODASYL systems committee, July 1970.