GEORGE SCHUSSEL

BUSINESS EDP MOVES TO DATA BASES

Requirements, advantages, and problems

George Schussel is vice-president of the Information Systems and Services Group of American Mutual Liability Insurance Company.

Over 500 American companies are successfully using data base systems in their EDP areas. This new technique automates much work currently handled by programmers. It allows for more programmer productivity and provides for broader systems control. As the evolution toward data base becomes universal, companies will consider recasting their systems to achieve these advantages. To enjoy these benefits, however, the user will have to invest in more hardware and software overhead. Even if the decision is not to proceed, the investigation of data base is a valuable test of the integrity of existing basic data.

A new concept in business data processing is achieving significant acceptance by both computer manufacturers and major users of computer systems. This new technique uses standard "data management systems" (DMS) software packages to implement business data processing on a "data base" (DB). Currently the number of American DMS users is estimated to be approximately 600. DMS availability offers the possibility of reducing the long-term costs of data processing and of increasing the capabilities of the business programmer by automating many of the functions now performed manually. The DB concept has so developed over the last decade that it can be implemented by most medium- and large-scale users of data processing equipment. In the 1960s COBOL programming language emerged as the standard language for business DMS; in the 1970s a combination of data bases and COBOL will become the standard procedure for implementing business data processing systems.

In the two decades of the computer age, data processing applications have progressed from a few scientific and military areas, involving complex calculations, to applications comprising most information processing in government and business throughout the United States. In 1953 a typical large-scale computer (of which there were only a handful) was the UNIVAC I, costing between $1.0 and $1.5 million. In minutes it performed operations that dozens of mathematicians using hand calculators would have taken months to accomplish. The IBM 370/165, which is a typical, large-scale computer of 1972, costs approximately $4 million. For this amount of money, however, the purchaser will have 50 to 1,000 times more capability. The present generation machines, in combination with their programming software, outperform UNIVAC I in thousands of ways.

It is probably safe to say that the IBM 370/165 is at least 100 times more powerful

"The development of operating systems software—master programs that are the brain of a large computer system—has provided peripheral tools which improve the productivity of the computer hardware."

than the UNIVAC I, while costing perhaps three to four times as much. That is to say, the capabilities of computers have increased dramatically while the price of computing has drastically decreased over the last two decades. The cost-performance ratio for large-scale computers has improved by a factor of at least 30 to 1 over the last twenty years.

In contrast, the cost of programmers has risen with the development of computer capabilities; a senior programmer's salary has increased from approximately $5,000 to $14,000 a year. For efficient use of the computer the functions of the programmer must be skillfully integrated with computer power. Basic economics dictates that when computer costs are decreasing and programmer costs are increasing, more emphasis should be on the use of the former than on the latter.

This is, in fact, what has happened over the years. High-level computer languages, such as COBOL and FORTRAN, have been developed; they require less programming expertise than their predecessors. The computer itself now performs many of the housekeeping chores previously delegated to the programmer. The development of operating systems software—master programs that are the brain of a large computer system—has provided peripheral tools which improve the productivity of the computer hardware. Repetitive functions—input, output, sorting, and report generation—have been standardized into programs that any programmer can call upon to complete his job.

DB, the next phase in this evolution, has existed for about five years, but only today is it beginning to gain widespread acceptance in the data processing community. The DB concept involves software packages (usually called DMS) that remove many of the functions of the business programmer from his direct control and vest those functions in the DMS package. The package standardizes all information for the systems application while supplying it to the programmer in a more retrievable form.

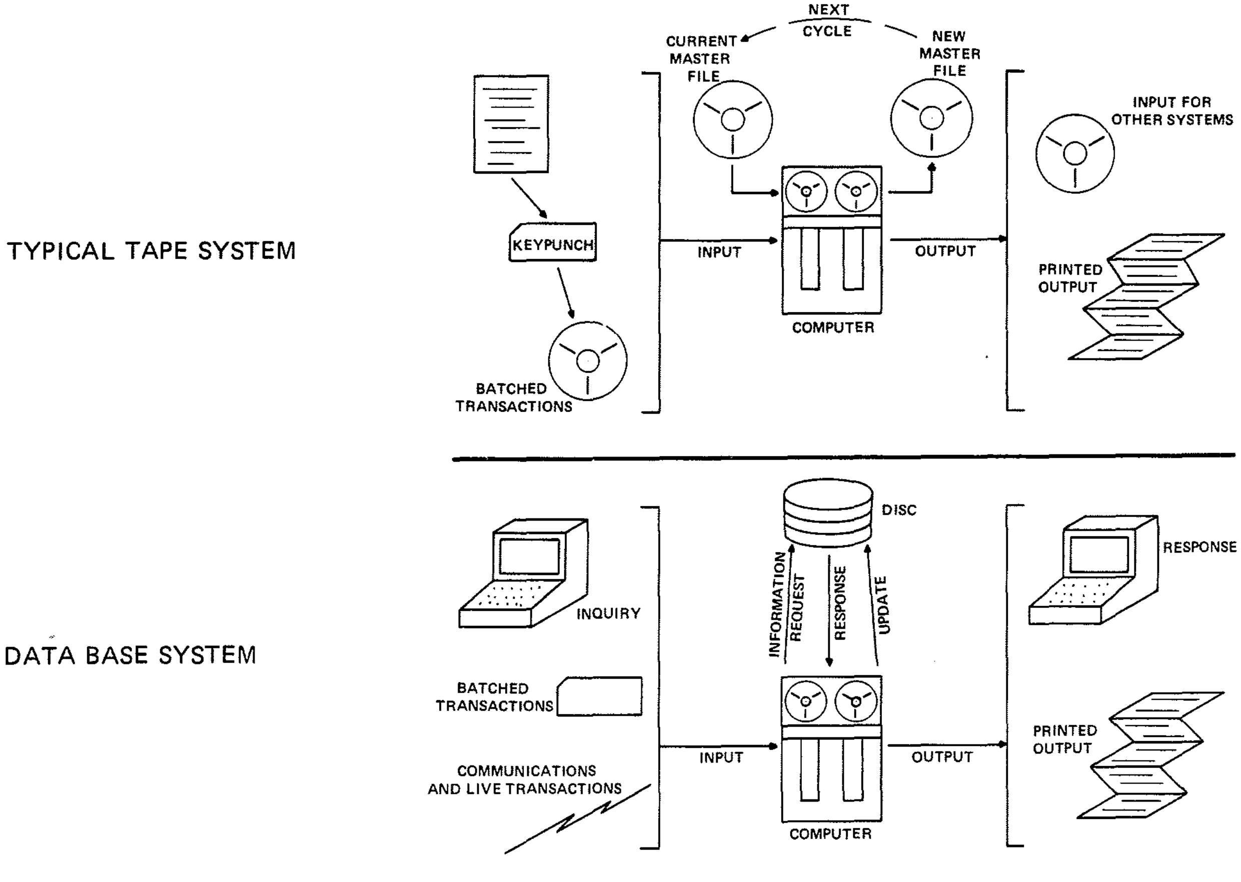
Historically, business data-processing files have been oriented toward storing information on tape. Each application had its own master file containing its own information. This file often contained data which was repeated on other master files.

The DB concept stands in stark contrast to this. It says, in effect; "Let us put all our eggs in one basket. Instead of storing identical information in different places, we'll put it all in one place. Everyone will know where it is and how to get it; it will be on an accessible piece of equipment and we'll control updating."

Figure 1 contrasts the DB approach with tape systems. In the typical tape system, information is keypunched into computer format. Next, a number of similar transactions are batched. Periodically, these are put into the computer system along with the program which processes these data. At this time the current master file to be updated is loaded. The application program residing in the computer then matches the transaction against the master file, produces the output reports summarizing these transactions, produces input for other systems, and produces a new, updated master file. This updated master file becomes the "current master file;" it will be used in the next processing cycle for this application. The previous "current master file" becomes what is called a "father tape."

FIGURE 1

The Data Base Approach and the Tape System



This processing technique contrasts with DMS processing on a DB system. Many of the transactions entering the DB will evolve similarly to the way they are currently developed in the tape system, from paper to key punching and batching. However, other transactions and inquiries will come through more immediate on-line techniques such as local terminals, communications with other computer systems or remote users, and computer-assisted instantaneous monitoring of transactions (as in point-of-sale monitoring of the checkout register at a retail store).

The programs required to process the input and the master files against which input will be processed are located on disc storage and available in thousandths of a second. Data entering the computer are analysed for what program is needed for processing. This program is called into the primary memory from the disc and is activated. As the program proceeds, it will require certain data off the master file. Although the entire master file exists on discs, only a small portion of it is related to any single program. Therefore, only necessary pieces of DB information are entered into the primary memory, updated, and returned to the disc.

There are two ways to communicate with other systems. Either the input data are kept in the computer and other application programs are brought in to use the same information, or intermediate storage files—on the disc or sometimes on tape—are created. These subsequently serve as the input for updating the other systems.

The output from the data base computer is often similar to output reports generally obtained. In addition, however, the on-line characteristics provided by the DMS enable some responses to be provided directly to teletypewriter or cathode ray tube terminals. Only seconds elapse between the inquiry and the answer.

Companies now utilizing time-sharing or interactive man-computer systems already appreciate some of the advantages of the DB approach. These interactive systems permit input and response from inquiry terminals (or through the use of communications) in the same way that DMS do. However, the interactive system is only one-half the answer since the data necessary for a particular application typically require special loading onto a disc. If it is already there, it is not interfaced with other applications for uniform use and updating by several systems. In other words, the DB approach will combine several master file updates into one operation.

RECENT DEVELOPMENTS

The advent of the large-scale disc, with its capacity of 100 million characters on one removable disc pack, was the engineering breakthrough which made practical the DB concept. It is now possible to retain up to several billion characters at any one time on line to a computer, and to summon any piece of that information in less than one-tenth of a second.

The concept of a total DB on a direct-access device such as a disc drive is not new. Initial approaches were tried on second generation computers with smaller, slower discs ten years ago. These approaches were not widely adopted because they did not employ DMS, standard programming packages, to do much of the tedious, yet necessary, programming housekeeping. These packages sell or lease for a small fraction of the cost of writing such a system. Several DMS packages are available in today's market. The best known are:

Information Management System (IMS and IMS-II), developed by the IBM Corporation for the IBM 360 and 370 lines of computers.

Integrated Data Stores (IDS), developed by General Electric (now Honeywell), and available on the Honeywell 6000 series of equipment.

TOTAL, marketed by Cincom Systems of Cincinnati and available for several computers, including the RCA Spectra 70 (now UNIVAC) line, the Honeywell 200 series, and the IBM 360 and 370 series of equipment.

In the United States alone, reliable estimates place the number of IMS installations already in production between 125 and 150; of IDS installations at around 100; and of TOTAL installations at approximately 250. Table 1 lists several well-known companies currently running some of their systems under a DMS package. Considering all of the other general-purpose packages marketed by the independent software firms, an estimate of 600 companies with active DB systems is probably accurate for the present time.

The DB approach does raise real questions for management: the answers reveal both the advantages and the disadvantages of the system. In addition, it has become readily apparent that a DB system necessitates a specifically qualified DB administrator. "Is a DB system something that we should be considering right now?"—this question reveals a whole new dimension for management consideration.

DATA BASE CHARACTERISTICS

The first important advantage in the DB approach is that the corporate DB contains all pertinent information that can be correlated and cross referenced. The over-all plan of this total corporate approach can be organized in several different ways. For example, the operation of a typical insurance company could be divided into two parts. The larger area would be the actual insurance operation: premium collection, claim processing and

TABLE 1

Companies Utilizing Data Management Systems

Banking

Cleveland Trust
Federal Reserve Bank of Kansas City
First National Bank (Dallas)
First National Bank of Cincinnati
First National City Bank (New York)
Manufacturers National Bank of Detroit
Northern Trust (Chicago)
Northwest Bancorporation (Minneapolis)
Seattle - First National Bank
United States National Bank of Portland

Insurance

Aetna Life and Casualty
Equitable Life Assurance
Employers of Wausau
Fireman's Fund
Hartford Insurance Group
Home Life
INA
John Hancock Mutual Life
Liberty Mutual
Massachusetts Blue Cross
Northwestern Mutual
Travelers

Manufacturing

Abbott Laboratories
Aluminum Company of America
Boeing
Caterpillar Tractor
Chevrolet Division
Combustion Engineering
Eastman Kodak
General Electric
McDonnell Douglas
North American Rockwell
Pillsbury
Raytheon
United Aircraft (Pratt & Whitney)
U. S. Steel
Westinghouse
Weyerhaeuser

payment, accounting and controls, and administration, including payment of expenses. The smaller segment concerns investing premiums to provide a steady income and capital appreciation. If the operation of these two areas is clearly segregated, there may be no need to correlate investment information with insurance results except in the annual statement where all results are gathered—usually manually. Given these definitions, investment information should dwell in a separate DB. Considering the limitations of today's computer hardware, this usually saves money.

A necessary criterion in establishing the DB—and one of its most essential characteristics—is that creators and users must agree on a common set of definitions for all information in the DB. This is not so easy as it might seem. Examination of data definitions in almost any medium- to large-scale user of data processing immediately exposes three classic problems: (1) synonyms, or identical items of data called by different names in different applications; (2) alternative definitions, or different systems using the same name to describe two different pieces of data; and (3) close definitions, or two different names used to describe different pieces of data with definitions so similar that there should be only one name and one definition.

To benefit from the advantages of DB, the corporate user must be willing to surrender his esoteric definitions in order to live with a common set acceptable to all. First, the designers must develop a complete dictionary of definitions. After they obtain the support of the users for these definitions, they must then keep it current and enforce it for new applications. The dictionary need not be particularly sophisticated. For example, it does not have to be on-line in a computer; it can simply be a loose-leaf binder with all of the data items listed alphabetically and carefully defined.

The next essential characteristic of the DB is that it resides in a *secondary* storage device, usually a disc drive. *Primary* memory is the very fast and very expensive magnetic core or integrated circuits. The essential benefit of the disc drive or other secondary device is that it renders all data quickly accessible. This contrasts with tape or card storage. It can

"Once the DB is established, the administrator is responsible for maintaining its security . . . . it is extremely important to prevent unauthorized or improper access to the DB."

take a computer 150 times longer to locate a piece of data nearer the end of a tape reel than the beginning. The DMS retrieves data more swiftly and in desired sequence.

Although disc drives are the most common direct access storage devices, either drums or magnetic cards or data cell devices are also used. Drums have faster access times than discs, but they are much more expensive on a per-character-stored evaluation. Magnetic cards or data cells are much slower but cheaper. Neither is in general use for DB. By 1980, completely new, massive random-access-memories (RAM) based on new technologies such as molecular physics (bubble memories) or optics (laser memories) should make electromechanical discs, drums, and data cells obsolete.

Another essential feature of the DB system is that it can be interrogated by on-line terminals, usually cathode ray tube, TV-like devices. While not all DB systems have this feature, the trend toward DB allows for useful exception reporting, and the most efficient method for this is through inquiry and on-line terminals. Most of the leaders in DB development and marketing who do not presently have on-line features for their systems plan to include on-line and terminal access.

Most DMS may be implemented as extensions to one of the common languages used in business programming. Typically this language is COBOL, the universally standard language for business. Not all DMS are COBOL extensions, however; a couple have provided a new language as a complete replacement for COBOL. Nonetheless, this approach is not so useful nor so powerful as the basic COBOL language extended through DMS facilities.

The final key characteristic of a DB is the concept of multiple users. Users from different departments and for different application programs will be using one common DB. This has several important implications for the company electing this approach. Two of these—common data definitions and the data element dictionary—have been discussed. A third is the creation of a totally new position, the data base administrator.

DATA BASE ADMINISTRATOR

The DB approach is oriented to data or information. This concept differs from most current business data processing which is geared to applications and treats data as an adjunct to programs implementing those applications. Now, however, the definition and creation of the corporate DB is primary while the various application programs, which select needed data and operate against the shared total DB, are secondary. This stresses the importance of the way the information is created, maintained, defined, and handled. Many DB benefits result from requirements placed on the systems department for the establishment and maintenance of the total DB.

The data administrator's job resolves into two broad areas. First is the creation of the DB. Until the first run, this may be a full-time job. But even after the first few applications are running, new data continue to enter the business stream and old ones are culled. The second part of the administrator's job is, of course, administration. He is responsible for the control and usefulness of the DB. During the creation of the DB, the administrator confers with various users to develop his data element dictionary and to achieve concurrence on definitions.

He must also structure relationships among different data within the DB, describing how one piece of information is related to or belongs to another. For example, the concept of "Age" may belong to a record entitled "Employee," which is part of a file marked "Personnel."

Once the DB is established, the administrator is responsible for maintaining its security and for developing procedures for recovery from disaster. With all corporate information eggs in one basket, it is extremely important to prevent unauthorized or improper access to the DB.

Through DMS tools, statistics on data usage, frequency of access, and total volume are readily available. By carefully monitoring these statistics the administrator can optimize the data locations on different peripheral devices and suggest new definitions or the elimination of unused information. For example, some DMS permit structuring of the total DB on several peripheral devices. A drum peripheral generally demonstrates high cost per piece of information stored but offers swift access when compared to a disc. Therefore, data which are needed frequently during the day may be more efficiently stored on a drum. By determining the frequency of access, the response times required, and the mode of retrieval, the administrator is able to optimize the use of the hardware for achieving best results through the DB.

The data base administrator is in a position of crucial importance to the data processing department and the entire corporation. Depending upon the organization of the data processing department, he can report either to the manager of all data processing or to the manager of systems design and programming. Any point lower within the systems organization will not give sufficient visibility to the importance of this function.

ADVANTAGES OF THE DB APPROACH

The "total corporate management information system" concept, which was discussed in numerous journals during the second half of the 1960s, proved to be a disaster in the real world. Out of this era, however, the concept of the DB approach to the development of management systems emerged as a clear and viable concept. Because the software evolution has remanded more and more programming tasks to the computer in an attempt to make programming easier, it is clear that a DB tool-arsenal for the programmer will emerge as the next stage.

The primary reason for adopting a DB is to realize significant savings in programming and debugging time when developing new business applications. The few studies that have appeared on this subject show that once the DB has been established—by no means a trivial task—implementing comparable programs with competently trained programmers can be two to four times faster than with the COBOL programmer who must develop his own master files. Since the cost of hardware in most large data processing departments has dropped to below 50 percent of the budget, and in many cases to 25 percent of the EDP budget, any plan which saves on labor costs provides a persuasive argument.

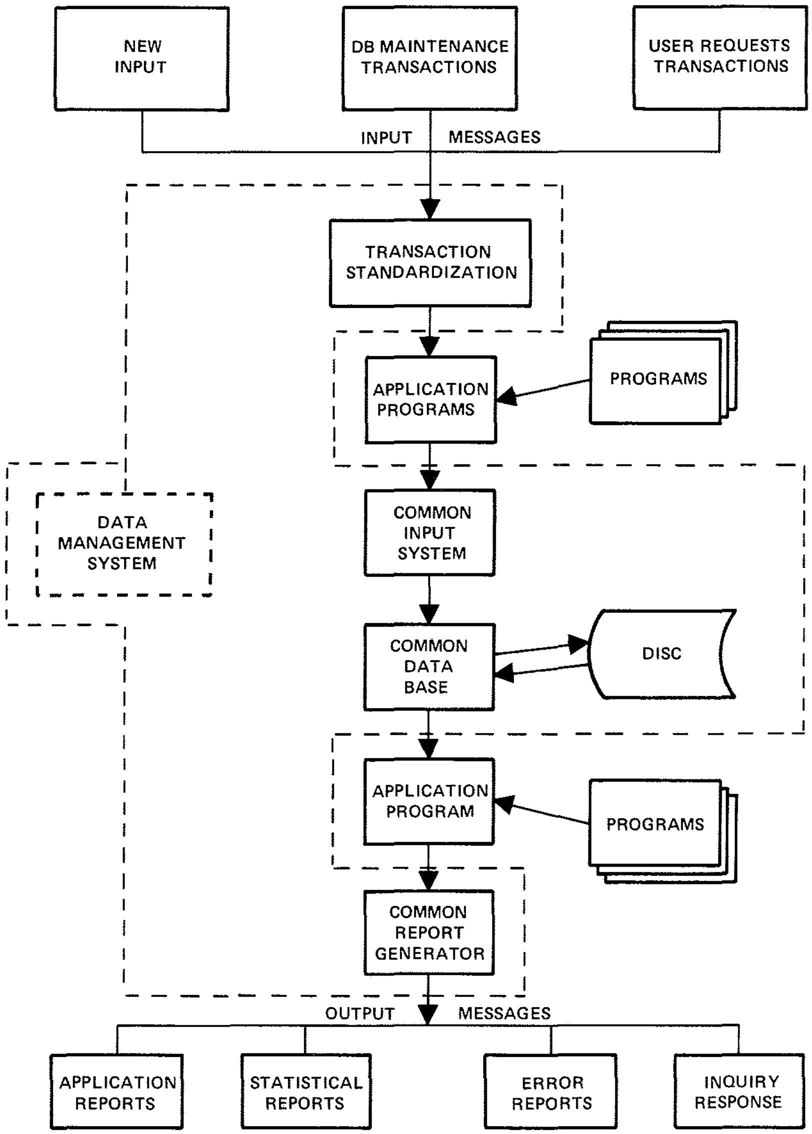
The second important reason is the efficiency in data storage achieved through the nonredundancy of information that derives from clustering all data in one file. Also, when the information is in one file it can more easily be updated, is more readily accessible, and is therefore more reliable.

Since the hardest part of business data processing is building and maintaining the master files, the ease of developing new applications on an existing DB is significant. With the programmer relieved of much of the day-to-day work on the master files, the systems shop can use less experienced programmers for its routine responsibilities and become productive more quickly under the DB environment.

One of the DB benefits most helpful to top management is the ability to process one-shot programs for decision making or operations research against data already resident in the corporate DB. Because they currently require centralization of numerous tape files, such programs are either impossible to implement or prohibitively costly and time-consuming in today's typical environ-

FIGURE 2

Data Management Systems Job Operating on a Data Base



ment. It is, of course, this type of information which is one of the most valuable products of a data processing department.

Figure 2 is a flow-diagram of a representative DMS job operating on a DB. The question can be asked: "How does this help solve systems problems?" The problems and the improvements offered by a DB approach are explicated in Table 2.

Because of these multiple advantages, companies presently using second-generation equipment or emulating with third-generation equipment should study assiduously the concept of redesigning major applications for a DB environment. Merely rewriting for COBOL or another language compatible with third-generation computers is obviously not enough.

PROBLEMS WITH THE APPROACH

As many companies have already discovered, implementing a DB is not a simple job. Numerous problems should be fully understood before commitments are made.

TABLE 2

Advantages to Data Base Systems

SYSTEMS DEPARTMENT PROBLEMS	SOLUTION WITH DATA BASE
Firm Commitments for Systems During the design stage of a new system, it is necessary to firm up the specifications so that effective programming can commence. The specifications that have been generated by analysts may be incomprehensible to management personnel. Therefore, even though approval is given for a set of specifications, numerous changes may be required once management sees the final product.	Since DMS provide an easier mechanism for changing processing and output formats, implementation under this type of system is easier to alter to meet changing or uncertain requirements of management.
Report Format As different managers come into new jobs, the requirements asked of data processing often change. Reports change to cover different scopes of information with different formats.	Changes in the information contained in reports and even the scope of the data processing are made considerably simpler through the use of report generator modules available in DMS.
Acceptable Data Base In the typical business environment there is difficulty in identifying, defining, and agreeing upon definitions or data items that are repetitively used in different systems.	Instead of having numerous different data bases which were set up at different times with different definitions, the DB is standardized at one time and subjected to continuing review. This procedure minimizes the problems due to this factor.
Data Duplication Data that repetitively exist in multiple files cost money because of multiple costs of storage.	The whole concept of the DB approach eliminates this problem by storing one value per item in one location only.
Master File Manipulation The development and maintenance of master files represent the most difficult job facing the business data programmer.	Much of the work required in establishing the master files has already been done once the common DB is established. The application job is, therefore, much easier.
Programming The cost of programming is high and rising all the time. It is difficult to obtain productivity increases in this area.	Because the problem of master file creation has been drastically simplified, the resulting application programming is much simpler and on a higher level with the DB approach.
Data Interchange Among Systems The various systems that constitute the total management information system for the company were designed by different people in different locations at different times. This has resulted in significant problems in effectively interchanging data among these systems.	The common data of the DB approach alleviates this problem by permitting a much simpler systems interface.
Interaction With Management Top management has difficulty in interfacing with the data processing department because of the specialized jargon used by EDP personnel.	The DB approach is one that is understood by almost all nontechnically oriented top management people.

(Continued on next page)

TABLE 2, *Continued*

SYSTEMS DEPARTMENT PROBLEMS	SOLUTION WITH DATA BASE
Modern Systems Design Availing your systems of the latest technology for on-line interface and readily available flexibility remains a continuing problem.	The DB approach is a modern, flexible approach towards business data processing. Most DMS that do not have on-line system characteristics at the present are being designed with future capabilities for access by terminals.
System Evaluation Every time a new system is designed and implemented, oaths are taken to continue to review the usefulness of the system and modify it so that it stays current and useful. In practice, this never happens.	The ability to monitor a system and its data is far simpler under the DB approach since the DMS can develop continuing statistics on both the data and the application.
Documentation Poor system documentation is prevalent since it is considered one of the least desirable tasks of programming. As a result, later maintenance on the system becomes more difficult because of inadequate documentation.	Documentation under the DB approach is simpler since documentation of the master file structure has been made simpler.
Large Maintenance Load The Systems Department always seems to be behind in implementing the many maintenance level changes that are requested by user departments.	Because program patches are easier to apply under the DB approach, maintenance of programs becomes easier and the backlog can be cleared away with fewer programmers.
Large Program Development Costs Because implementing new applications often requires numerous programs and complex manipulations among many different files, the cost—especially labor costs—of implementing new systems are too high.	The DB approach with central maintenance of the master files simplifies and reduces the costs of implementing new applications after the DB has been initially created. This can be especially important in programs that are written for "one-shot" decision-making purposes.

The first and probably the thorniest problem is that, while DB is conceptually advanced, there is a concomitant lack of well-trained people to implement it. Time will overcome this lag. Nothing in the DB approach precludes its universal acceptance in the 1970s just as COBOL was welcomed during the 1960s. Training is readily available from computer manufacturers, software vendors, and management firms specializing in executive and technical seminars. The passage of time should remedy this shortage of people; already, several hundred applications of DB systems have been implemented and some knowledgeable people are available.

A second DB problem is that it withdraws functions the programmer once performed and delegates them to the integrated hardware/DMS system. By so doing, a need for computers with larger memories and greater processing power is created. To date, most DB users have been those who can afford powerful computer systems. This is likely to be the situation for a few years, until new computer capabilities can depress the overhead cost to a level feasible for smaller users.

Another problem related to DB is also an associated opportunity. Rarely are a company's existing systems merely converted onto a DB. The resourceful company will use the conversion as a complete redesign springboard for systems and will develop a more versatile product. For example, it will generate exception reports instead of grinding out massive stacks of paper for inquiry use. One requirement for a redesign is a total commitment of money and time; after all, we're talking about systems that may have had a ten-year genesis. They will not be replaced quickly or easily, even with tools such as DB. The fact, therefore, that DB implementation merges with redesign of major systems infers that it may involve more time and more money than other systems. For example, redesign and implementation of the complete back-office computer system for one of the New York Stock Exchange's largest member firms were recently completed after four years at a cost of $3 million. This project used straight COBOL. DB was not considered because of the state of the art four years ago, but the same project with DB would have been of the same magnitude. [For more information see Schussel and May, "Wall Street Automation: a Primer," *Datamation* (April, 1970).]

Along with the possibility that the new system may not be just a simple conversion of the old but, in fact, be new and different comes the concomitant problem of inability to use parallel runs. These computer runs allow old runs to be compared with new runs to reveal any errors in the new system. This means that the output from the new DB will have to be visually examined, a much more laborious process than is involved in many standard system conversions.

While DB technology permits more junior programmers to become productive sooner, it also necessitates greater expertise on the part of the systems programmers. These people, who maintain the computer's operating system, will also have to control the DMS. They are typically among the most highly skilled and highly paid data processing professionals.

"One requirement for a redesign is a total commitment of money and time; after all, we're talking about systems that may have had a ten-year genesis. They will not be replaced quickly or easily . . . ."

Malfunctions in computer hardware or software during the running of today's typical business applications call for reruns. Normally the program is rewound to a check point or to the beginning, the original master file and transaction file are loaded, and the run is regenerated. Even if a particular master file is destroyed, one can return to the "father" tape—the original master from which the current master was created—and rerun a couple of times to make the system current. Under the DB system, several programs may be updating and accessing the DB at the same time; therefore, the results of a malfunction in the hardware of software are not so easily cured. The primary technique for recovery in this situation involves continual journaling of transactions against the DB and the use of these lists to restore. Because of the large number of potential interactions, this procedure is more complex and requires more time than would be necessary with current procedures.

One final problem is the necessity for top management's commitment, initially and throughout the project. In a complete evaluation of the company's systems, previously undiscovered errors will almost certainly come to light. Data processing and top management must be prepared to accept these as normal and must take necessary steps to eliminate such inefficiencies in the new system design. Management commitment is also necessary because unfamiliar definitions whose use is mandatory to interface with the DB may be generated.

On a short-term basis, a company's conversion to DB will also be more expensive, since it represents a major new project that would not otherwise exist. Also, because of its size the project will almost certainly take time—one of the most frequently cited cardinal sins of data processing projects. Unless top management understands the objectives and is committed to the DB approach, some of these perils may lead to the premature death of the project.

CHARACTERISTICS OF SUCCESSFUL DB IMPLEMENTATION

Some valuable lessons have been learned by comparing successful and unsuccessful DB approaches. Successful users have taken the following steps.

Corporate Committee

A team of middle managers representing all major business functions was formed at the outset, sometimes during or sometimes after the preliminary feasibility study had been conducted by the data processing department. This team defined data and heard appeals regarding conflicts in approaches. All DB-related functions were represented on this team.

Logical Organization

The DB structure itself and the list of applications to be programmed against it were organized functionally: marketing, production, engineering, and so forth. Peculiarities of an organization, such as a specific individual holding a certain job at only one point in time, were not stressed in the scheme.

New Approach to Systems

The corporate committee, in concert with the data processing department, identified the decision-making processes in the business and designed the system accordingly. In other words, current systems and their requirements were not accepted as the prototype for the new system. A more detailed investigation into information needs formed the basis for the DB system.

Dictionary Development

Early in the development the DB elements were defined and put into a dictionary format. It was at this stage that the relationship among the data elements was often finalized. The analysis of this data was structured to eliminate duplication and redundancy and to develop standardization.

Data Administrator Control

A system for data control, usually managed by the data administrator, was established and implemented before system activation.

Phased Implementation

Finally, a schedule was developed for phased implementation of reports and information out of the DB system. While over-all implementation may have been lengthy, products were identified which could be successfully delivered at early points and were made available as soon as possible throughout the development cycle.

The computer is almost exactly what man is not. It is capable of paying undivided attention to unlimited detail; it is immune to distraction, precise and reliable; it can carry out the most intricate and lengthy calculation with ease, without a flaw and in much less than a millionth of the time that would be required by its human counterpart. It is emotionless, or so we suppose. It suffers from neither boredom nor fatigue. It needs to be told only once; thereafter it remembers perfectly until it is told to forget, whereupon it forgets instantly and absolutely.

When man and machine work together, the shortcomings of each are compensated by the other. . . . The potential of such a combination is greater than the sum of its parts.

—Steven A. Coons in
Information