EDP ANALYZER

DECEMBER, 1976
VOL. 14, NO. 12

TOWARD THE BETTER MANAGEMENT OF DATA

With the advent of the computerized data base, the importance of the effective management of data is finally becoming recognized. As Charles W. Bachman said in his ACM Turing Award paper, heretofore the programmer has always viewed things from inside the computer, with data passing into the computer from tape files. But with the integrated data base, the programmer may have to become "a mobile navigator who is able to probe and traverse a data base at will." There is a tremendous amount of activity in data base technology, to the point where the literature is almost inundating. In this report, we have drawn upon a few landmark conferences and publications in order to give you an overview of where the field stands today and what may evolve in the next five years or so, in terms of the better management of data.

A session at the 1976 National Computer Conference in New York City addressed the subject of "data base decisions" from the user's point of view. Executives from three large user organizations were asked to answer five questions: (1) What is your status and plans on using data base management systems (DBMS)? (2) Would the availability of a standard DBMS change your plans? (3) Will data base technology change the way you audit and monitor your data collection and its usage? (4) Have government regulations, such as the new privacy regulations, changed your data base plans? and (5) Has the pace of technological improvements changed your data base plans?

We will summarize the statements of these executives.

Ford Motor Company

Mayford L. Roark of the Ford Motor Company gave his views on these questions. Ford, with sales of over $24 billion per year and over 416,000 employees worldwide, has 20 large divisions in the U.S. plus affiliates in other countries. The systems office in Dearborn, Michigan, provides central control over the acquisition of computing hardware and software, including DBMS.

Ford started using DBMS in 1970, with the installation of a TOTAL package. By mid-1976, 34 DBMS were in use in North America. Thirteen of these operated on IBM equipment; these included three IMS systems, nine System 2000, and one TOTAL. Twenty of the systems were Honeywell IDS-1 systems, operating on Honeywell equipment, and one was a Burroughs DMS-2 system on Burroughs equipment. In addition, seven new DBMS application systems were under development. The active data bases involved 4½ billion characters of data, which represented less than 10% of the computerized files.

Why the variety of DBMS, Roark was asked. Historically, different segments of the company have preferred to use different brands of hardware and the company has continued to use those brands. The choice of the DBMS has followed the choice of computers. But the selection of a DBMS has also been influenced by any prior DBMS that were used, by the file size involved, and by what is desired from the DBMS. For example, IMS has been used on IBM equipment where very large files ex-

ist, where flexibility of operation is desired, and where a query capability is not too significant. On the other hand, System 2000 has been used with smaller files where a powerful query capability is desired.

If the DBMS is relatively easy to learn to use, its use tends to proliferate more quickly, Roark said.

On the question of a standard DBMS, Roark said that the existence of such a standard system would not necessarily change their planning. The hardware selection and the application are the first determinants of the DBMS. He contrasted a standard DBMS with COBOL. In the 1960s, Ford adopted COBOL as a standard programming language not so much because it represented a "standard" as because it represented a real step forward over assembly languages. Similarly, a "standard" DBMS would have to represent a significant step forward over what else was available to the company, in order to influence the company's plans. Neither the CODASYL approach (very similar to IDS-1) nor the relational approach currently appears to be that significant an improvement over what they already have, Roark added.

Concerning auditing and monitoring, Roark said that so far none of the data base application systems have involved financial data. So while Ford has an active EDP audit group, it has never been called upon to audit a data base system. However, the company has standard design practices that apply to all application systems, such as bringing in the internal auditors during the design stage. So it appears that their data base systems will be auditable, he said.

Similarly, the new privacy regulations do not apply to the private sector and so the company has no direct experience with these regulations in a data base situation. The question of meeting proposed privacy regulations for the private sector has been studied in some depth, and the company believes that it can comply with those regulations at an increase of expense.

As far as new technology is concerned, the most attractive thing on the horizon is the distributed data base, Roark said. The technical problems of the distributed data base are still to be coped with.

Roark had some advice for beginning or early users of DBMS, based on Ford's experience. Don't assume that a modest success with a pilot system guarantees success on a big DBMS project, he said. Don't try to do too much; work by degrees. Do only as much as is necessary and allow a period for debugging and tuning the system. And be sure to give adequate training to all of the people involved, he cautioned.

Tenneco Inc.

Gary Bearden of Tenneco Inc., Houston, Texas, gave his views on the questions. Tenneco is a diversified holding company of manufacturing, oil and gas transmission organizations. Sales are over $5 billion annually and the company has over 83,000 employees. The company follows a policy of maximum decentralization within its seven major divisions; all operating decisions are made by division management and profit center management. Long range (capital) decisions are made by corporate headquarters, which also provides consulting help to the divisions.

Several of Tenneco's divisions are using DBMS, Bearden said. These include DBOMP, IMS, and IDS-1. In all cases, the divisions had to cost justify the use of the DBMS.

"No," said Bearden, "it is not likely that a standard DBMS would change our plans. For one thing, it isn't too likely that one standard system will prove to be cost effective in all Tenneco situations." So he sees little advantage to a standard system, from an application standpoint. However, if a standard system were available and in use at some Tenneco sites, it *would* help in personnel recruiting and transfer. Today, it is hard to evaluate the DBMS experience of people.

EDP auditing is just beginning, Bearden feels. EDP auditors are just learning to audit conventional application systems and are in most cases not ready to take on data base systems. Moreover, it may be a number of years before EDP audit *is* ready to tackle this area. So DBMS will have an effect on the audit function, by making it more challenging.

As for the impact of government regulations, Bearden feels that regulations other than privacy may have more effect on data base usage. He cited the occupational safety and health program as one example where an organization might be required to move to a central data base in order to comply with the regulations.

The biggest impact of evolving technology will come from those developments leading to lower costs, Bearden believes. But do not wait for to-

morrow's technology, he added; use today's technology to solve today's problems, as long as that technology is cost effective.

Bearden seconded Roark's words of caution for new and early users of DBMS. Top management comes to data processing with a problem, not with a request to use a particular solution such as a data base. Select a solution to that business problem in which you have confidence, he said. If the solution is risky, try it out on something small. Don't use new technology until you have confidence in your ability to use it. Further, don't immediately jump from a small data base system to a large one; rather, work your way up from the small to the large systems.

Social Security Administration

William Hanna of the Social Security Administration, Baltimore, Maryland, presented his views on the questions. SSA is an agency of the U.S. Department of Health, Education and Welfare. Its files contain over 230 million records on individuals, 150 million of whom are still living, and on some 20 million employers. SSA programs include payments to retired persons, to survivors of insured persons, to disabled persons and to people eligible for supplemental security income, as well as payments for health insurance benefits (Medicare).

SSA has more than 100 computers in use. They have made some use of DBMS; most of these cases have achieved modest success but there was also one particularly unsuccessful case.

Three competing teams of system designers have been established to develop alternative plans on the way SSA will conduct its operations. Seven major alternative plans have been proposed and are being investigated; all of these involve the use of data base technology. It is expected that the selection of the best approach will be made at the end of this year, after which it will take about five years to implement that approach. So the use of data base technology for *huge* files has been investigated in considerable depth.

If a standard DBMS system were available, it probably would not change their plans, Hanna said. But it might make the overall implementation task somewhat easier, he felt.

The use of DBMS probably will make the audit function more difficult, in his view. The auditors may have to shift to a dynamic audit approach. With a huge data base, it will not be feasible to allow time for dumping that data base or for interrupting operations for long periods, to meet audit needs. Also, with conventional systems, it is relatively easy to assign accountability for data integrity. In a data base environment, this will be much harder.

SSA is subject to the Privacy Act of 1974, and so has had a chance to assess the impact of that legislation on computer operations. Such government regulations will increase the complexity of a data base structure, Hanna feels. Further, this increase in complexity will lead to higher costs and hence more difficult cost justification. Thus government regulations will affect plans for using data base technology.

Recent technological developments have brought mass storage systems with huge capacities to the market. These mass storage systems will allow SSA to do things that just were not practical before. Further, data base management is becoming available for these mass storage systems. So technological developments are affecting SSA's plans.

As for advice to new and early users, Hanna urges that an organization start using DBM technology on a small application and allow adequate time for its implementation. The biggest troubles arise when management allows, say, only half the time period for implementing a system that the data processing staff believes that it will take. Avoid such situations like the plague, Hanna warns.

So here are the views of three executives, from quite different environments, on how data base management technology is being used in their organizations. We will return later in this report to the views of some other people on these same subject areas—new technology, standards, auditing, and government regulations. But first let us consider where the computer field stands today in the use of data base technology.

Stages of evolution

Gibson and Nolan (Reference 1) have pointed out a useful pattern of evolution in computer use. We will paraphrase and expand on their stages of evolution, as applied to the use of new technology.

Early successes. The first stage is the beginning use of new technology. While some stumbling generally occurs, successes also occur—or else the use of that particular new technology stops right there. Consider, for example, the success of some early order entry systems using data communications, such as the Westinghouse system.

Proliferation. Based on the early successes, a rapid growth of interest in the new technology develops. New products and/or services based on the technology come to the marketplace. These are tried out in a variety of application systems. Following our example, consider the growth in the use of data communications products and services following the success of the order entry systems. (The order entry systems should not be given full credit for the growth of data communications but they were probably the single most important application of the technology at the outset.) This proliferation stage is the learning period for the field, both for uses and for new products and services.

Control of proliferation. A point is reached where it is apparent that control must be applied to the proliferation. Among users, costs of using the new technology get too high, in management's mind. Much waste is observed from using a variety of approaches. The integration of systems is attempted but proves difficult. From the suppliers side, efforts toward standardization occur. In the area of data communications, management calls a halt to setting up a new data communications link for each new application. Out of this control comes a network structure for serving all applications, such as IBM's SNA and NCR's DCU, which we discussed in the July 1976 report.

Mature use. At this stage, the use of the particular new technology might be considered mature. The stage has been set for introducing still other new technology, wherein the pattern is repeated.

In what stage of evolution is the use of data base technology today? We see it as in the end of stage 1 and the beginning of stage 2. There appears to be in the order of 2,000 true DBMS installed worldwide, and the rate of installations seems to be increasing. There are a number of competing systems in the marketplace, using a variety of system architectures. Here are some of the more popular systems which we have discussed in previous issues of EDP ANALYZER: "conventional" DBMS: IDS, IDMS, IMS, TOTAL, DMS-1100, plus some other CODASYL-type systems; (2) inverted file DBMS: System 2000, ADABAS, and Model 204; (3) self-contained systems: INQUIRE, RAMIS, ROBOT; and (4) file management systems: MARK IV, ASI-ST.

The file management systems have some but not all of the characteristics of DBMS. If they were to be included in the census, they would almost double the count; there have been over 1000 installations of MARK IV, for instance. Both MARK IV and ASI-ST have convenient programming and report specification features and both have been interfaced with the more popular DBMS so as to provide enhanced capabilities.

It is not unusual to find two or more systems in use at the same installation. For instance, in an IBM environment, IMS or TOTAL may be used for managing the data base, MARK IV or ASI-ST tied to the DBMS for providing a reporting capability, and selected data pulled off to special INQUIRE or RAMIS files for handling certain ad hoc queries.

With only 2,000 or so DBMS installed, it is evident that only a small fraction of today's computers are using a DBMS. But that number has doubled in the past three years, from our estimates, and will probably more than double again in the next three years. And it is also clear that there is a variety of non-compatible DBMS on the market. So proliferation already exists, from the supplier standpoint, and is beginning to occur from the user standpoint.

Problems with proliferation

The computer field has had lots of experience with proliferation. This is the period when everyone wants to "do his own thing" or "go his own way." It can be illustrated by an example that is familiar to most everyone in the computer field—the proliferation in application system development methods.

With people "doing their own thing" in the system building process, it is not surprising that the following proliferation occurred. No standard development methods were used. No standard data definitions were developed. Data files were fragmented; when a new application system was being built, none of the existing files would meet its needs so a new file was set up, partially duplicating other files. Without standard development methods, documentation tended to be minimal or non-existent. Maintenance and enhancement

costs turned out to be high, because the systems had not been designed according to standards that would have made changes easier. Special analyses programs were hard and costly to create, often because of the proliferation of data files.

It has been the waste and poor productivity associated with such practices that has led data processing management to demand standard practices and standard data definitions.

In other words, it has been the inefficiencies of the proliferation stage that have led to the control of the proliferation. The proliferation stage is the learning period, the trial-and-error phase. It does serve a very useful purpose. But if continued too long, the harder it becomes to bring activities under control.

We recently talked to a data processing executive in a multi-national company. He had inherited a stage 2 situation wherein the company had a number of computer installations, all of which had been going their own way for a number of years. He was in the process of instituting control. He estimated that it would take *several more years* to get the degree of standard methodology that he was seeking.

This same phenomenon is happening in other aspects of the computer field. It is happening now with respect to mini-computers and word processing. And it is happening with DBMS.

Stage 2 cannot and should not be skipped; it performs a valuable service. The trick is not to let it continue too long.

So, in this report, we are addressing the question: what is the computer field learning in the present (proliferation) stage of data base technology? How soon should users think about moving into stage 3?

An overview of data base technology

There are several reports that do a particularly good job of giving an overview of the present state of the art in data base technology. We were well impressed with the Infotech State of the Art Report *Data Base Systems* (Reference 2). It covers the rationale for using a DBMS, the strengths and weaknesses of the leading approaches to data base management (network, hierarchical, inverted file, and relational), and user experiences with some of the leading DBMS (TOTAL, System 2000, ADABAS, IDMS, IDS, IMS, DMS 1100, and MARK IV).

The March 1976 issue of the ACM *Computing Surveys* (Reference 3) provides an excellent historical development of data base technology, a discussion of the CODASYL, relational, and hierarchical approaches, and a comparison of the relational and CODASYL approaches.

Why use a DBMS? Infotech surveyed a good amount of data base literature to find the reasons most frequently claimed for installing data base management. Following are the nine characteristics most frequently cited as the reasons for going to DBMS.

Data independence. It might be better to use the term "program/data independence." The term means a mechanism for isolating the application programs from the physical storage of the data, so that one might be changed without a widespread impact on the other. Full independence is an unrealized (and perhaps unrealizable) goal. But modern DBMS design *does* provide an improved degree of program/data independence. For instance, see Curtice's analysis of TOTAL, IMS, ADABAS, and System 2000 in Reference 4.

Data independence implies a degree of inefficiency, as the Infotech report points out. There is the translation (mapping) overhead for converting from the data definitions as used by the programs to the data definitions of the stored data. There is also the inefficiency caused by the concealment from the programmer of the consequences of his decisions. "Tuning" the data base—by adjusting the physical storage of the data and the inter-relation mechanisms—can help improve performance but at the expense of some loss in data independence.

Model real world relationships. This modelling capability is the factor that, perhaps more than any other, differentiates the DBMS from the earlier file management systems. A mechanism is provided for relating, for instance, a customer record with the current open orders which that customer has entered. These open orders, in turn, call for products so the ordered products are related to the inventory records for those products. The inventory records, in turn, are related to procurement records for those products. Without searching the whole data base, it is possible to move from customer to customer order to product to product inventory to procurement order.

It is interesting to note that this issue of modelling real-world relationships is at the center of

the arguments of different approaches to DBMS. We will have more to say about this subject shortly.

Data integrity. A data base has both advantages and disadvantages with respect to data integrity, when compared with conventional application files. On the advantage side, data may be stored only once; when this is so, it eliminates the possibility of inconsistencies in the same data item that can occur when that item is stored in two or more files. On the disadvantage side, since the data base is used by multiple departments, accountability for errors is harder to assign. With the creation of the data administrator function, however, it is possible that even greater attention will be paid to data integrity than has been true of application files.

Security. A data base makes data available to all user departments who have a need for access to the data. Moreover, it is possible that some sensitive data will be stored in the data base—for example, rates of pay, bonuses paid, and so on—that should have restricted access. So one of the main requirements for a DBMS is an effective security mechanism, for controlling access.

Somewhat like data independence, security is proving to be evasive. Some operating system/DBMS combinations have more effective security mechanisms than others. These mechanisms prevent most accidental unauthorized accesses and the less intelligently planned of the deliberate ones. But they still do not offer much protection against the well planned intrusion.

Elimination or reduction of redundancy. In general, data will be stored only once in a data base. Where redundancy is used, it will be used deliberately in order to provide better performance or backup.

Multiple entry points to data. Records in a sequential file are accessed generally in only one way—by a serial search of the file. Index sequential files provide two access methods, by way of an index or by a serial search. Data bases generally provide even more ways for accessing a given record, via the relationships that each record has with other records.

Infotech points out that shared data in a data base *requires* multiple access paths, because the different applications will access the data differently. But the multiple access paths mean a more complex structure. Recovery may well be needed more frequently than in conventional files and may take a longer time, too, as we discussed last month.

Centralized control over data and its use. The data base is forcing more awareness of data as a corporate resource. Because data in a data base is used by all authorized departments, it has become clear that the definition of the data cannot be left to the individual departments.

So the function of the data administrator (or data base administrator) has emerged to exert control over data and data definitions.

One disadvantage is that individual departments will not have as much liberty in setting up new data fields to meet their individual desires or needs. Any new data field stored in the data base will probably have to be cleared or authorized by the data administrator.

Concurrent access. Since the data base serves multiple application systems, no one application should lock up the data base and prevent access by the other applications. So a DBMS should support concurrent access by multiple application programs.

Two problems arise from this feature. One is the possibility of concurrent updating and the other is deadlock; we discussed these problems in last month's report. The first program to access a record for the purpose of changing it is given exclusive use of that record until the change has been made and the record released. Deadlock can occur when two programs are each waiting for the other to release a needed record. A DBMS is expected to automatically handle and recover from these situations.

Ad hoc query capability. In theory, ad hoc queries can be answered by means of retrieval programs working with conventional application files. But when the answer to a query involves data in multiple files, the problem becomes complicated.

With a data base, all of the data is available. So the use of generalized retrieval and reporting/display routines becomes attractive. A DBMS should provide a powerful ad hoc reporting and query capability, preferably one that can be used by a non-programmer type of person.

These then are the nine reasons that Infotech found were most frequently cited for moving to data bases, as well as the functions that the DBMS were expected to perform.

Types of systems

Infotech (Reference 2) gives a good discussion of the types of DBMS. Also, see Reference 3 and the paper by Olle in Reference 5a. DBMS can be classified by the means they use for expressing data relationships. As Infotech points out, such classification is only approximate because a specific DBMS may use multiple mechanisms for expressing relationships.

Relationships by access paths. This is the common way of expressing inter-relationships between data items today. There are four general categories of access path mechanisms: linking of files, file inversion, defined hierarchies, and defined networks. In each case, an explicit set of pointers and/or indexes are used to tie together the related data items.

Relational approach. Adherents of the currently popular relational approach—popular, that is, among researchers and academics—argue that relationships should *not* be defined by access paths. The user's view of the data is too much influenced by the specific DBMS and by performance considerations, if access paths define relationships, they say. Instead, relationships should be defined as simply and as cleanly as possible, and let the DBMS worry about the accessing.

In one sense, the relational approach is simple in concept. A relationship is expressed by means of two or more records containing the same key plus one or more data items. (We won't use the terminology that the relational approach people use.)

To illustrate, consider the following three types of records:

(1) EMPLOYEE NUMBER, NAME, DEPARTMENT, JOB CODE, RATE OF PAY;

(2) EMPLOYEE NUMBER, PRIMARY SKILL CODE, YEARS OF EXPERIENCE;

(3) EMPLOYEE NUMBER, SECONDARY SKILL CODE, YEARS OF EXPERIENCE.

Now assume that a secretarial position is open within the company, and it is desired to search to see if someone already in the company is qualified to fill it. The search request might specify that either primary skill code or secondary skill code must be such-and-such, minimum years of experience in that skill code must be such, and, say, that the person must *not* already be employed in one specific department.

The relational approach says that, given a data base that includes records of the above three types, the DBMS would extract all records that meet the search criteria. Note that record types (2) and (3) are both needed in the search for skill code and years of experience, and that record type (1) is needed in the search to determine that the person is not already employed in the specified department.

The query in this case does not tell the DBMS how it is to make the search. The query just says *what* is wanted and leaves it up to the DBMS to determine how to retrieve the records. The user sees no pointer fields in the records and the user has no idea whether indexes are used or not.

Two points should be made. One, the redundant use of the employee number in all three kinds of records is a hidden definition of relationship. Relationship is being expressed by redundant data. Two, the query language itself is independent of how relationships are defined—by access paths or by redundant data.

How does the relational approach find the desired records? That is the hitch; there are no good answers yet. One solution is a complete search of the data base, but that is time consuming for all but very small files. Another solution is the completely inverted file, but the completely inverted file is quite inefficient for rapidly changing data because the indexes have to be updated with every change. Another solution is to put the entire data base in an associative memory; when using such a memory, calling for a specific value of a field results in all records having that value of that field being automatically retrieved. The problem is, large associative memories are beyond the state of the art and even when they become available they may be at least two to three times as expensive as conventional storage devices. Still another solution is a parallel search, which is expensive in terms of hardware. The net result, we have been told, is that the relational approach is not very practical today for file sizes above ½ to 1 million characters—although we have not yet attempted to verify this claim.

Why, then, is anybody interested in the relational approach? One claimed advantage is that it has a good, theoretical foundation to it. The "relationship by access method" approaches do not have such a foundation. Another claimed advantage is that is relatively simple but powerful user

interface. It embodies the ad hoc query capability carried to the "ultimate" of today's technology. Still another claimed advantage is that computer scientists can use mathematical techniques in conjunction with it.

The main debate in data base technology today is between the adherents of the CODASYL approach and the adherents of the relational approach. Here are the arguments. *Relational:* Relational is the way to go; it provides a clean, simple user interface. We may not know how to implement it yet but we are working on it. Please do not standardize on the CODASYL approach that has no theoretical basis and uses a programmer-type interface. CODASYL: Why wait for something that is beyond today's technology and for which no one can predict when the technology will be available? We spend less time on theories, although our theories are complete, and so we can spend more time on solving real problems. Our approach works; it is available. Furthermore, there appears to be no reason why a relational-type interface cannot be developed for a CODASYL-type DBMS.

As a matter of fact, at least one such interface already exists. In June 1976, Honeywell announced the Multics Data Base Manager that supports both CODASYL and relational user interfaces. In addition, MDBM allows a variety of data base structures, including hierarchies, networks, and relational organizations.

This discussion of the debate leads us into the question of a possible standard DBMS.

Standards in DBMS

A working conference on future directions in data base systems was held in October 1975, jointly sponsored by the Association for Computing Machinery and the U.S. National Bureau of Standards. It consisted of an invited group of knowledgeable people in the field who met for 2½ days. The conference was divided into five working panels, each addressing one of five subject areas. Reference 6 is a report of the conference, the production of which was the major aim of the conference.

One of the subject areas was standards in DBMS; we will briefly review the results of that working panel here and will treat three of the other four working panels later in the report.

The standards working panel recognized four different groups of standards as being needed in the next five years. These were terminology, criteria, components, and usage standards. For instance, the panel pointed out that data base system designers must communicate with organizational management. If these designers continue to use terms such as "schema" (instead of the more common synonyms such as "plan" or "design" or "description"), then they will have trouble communicating with management. So a usable, useful standard terminology would be welcome.

Perhaps the main point made by this working panel dealt with a standard DBMS. *If* a data base system standard is to be adopted during the next five years, that standard must be based on the CODASYL specifications, said the panel. The main reason was that, to the working panel's knowledge, no other system has yet been submitted for standardization. Further, it takes about seven to ten years from the time that something is submitted as a standard until it is adopted (assuming that it *is* adopted), based on such prior cases as FORTRAN and COBOL. If some other data base system were submitted now for standardization, it is very unlikely that it could be adopted as a standard within five or even ten years. Also, most of the popular DBMS (TOTAL, IMS, IDS, System 2000, ADABAS, etc.) are proprietary packages; their suppliers might be reluctant to submit them as possible standards because they would then lose control of the systems.

Let us repeat the working panel's point: *if* a data base system standard is to be adopted during the next five years, it must be based on the CODASYL specifications. If someone wants something else as a standard, it should be submitted forthwith.

The working panel went further and urged that bodies such as the U.S. National Bureau of Standards support the adoption of the CODASYL specifications as a standard within the U.S., and that the support of an international body such as the Intergovernmental Council on ADP also be sought.

This acceptance or not of the CODASYL specifications as a standard is an emotional issue. Much the same group of people that have looked down their collective noses at COBOL have argued vehemently against the CODASYL data base standards. Our position right along has been that the

CODASYL specifications were well thought out, they can be supported by today's technology, and user interests would be well served by adopting them as a "common"—and eventually as a "standard"—approach.

It might be well to quote a paragraph from our March 1972 issue on "The Debate on Data Base Management": "Our opinion is that the arguments against the (CODASYL) proposals have not provided the needed burden of proof (of inadequate or unsatisfactory design). The opponents of the (CODASYL) proposals have not clearly demonstrated a superior approach to the design of the languages. They have presented no solid evidence that the (CODASYL) approach cannot evolve so as to adapt to future needs. They have given no evidence that a better data base management system will be available sooner—or even five years later, for that matter—if the (CODASYL) proposals are scrapped." NOW, almost five years later, we see no reason to change that statement. The CODASYL specifications still represent the only proposal of stature for a common or standard approach to data base systems.

Some of our friends have taken us to task for this stand and have pointed out what they consider to be serious flaws in the CODASYL specifications. We have checked some of these out with technologists we respect and have concluded that the points raised are not so much flaws as they are differences of opinion. As far as we have been able to determine, the CODASYL specifications *have* stood up under close scrutiny.

The ANSI/SPARC study group

Bachman (Reference 7) gives a good overview of the work of the ANSI/SPARC data base study group, of which he was vice chairman. Also, a summary of the report of the study group will be found in *Database Journal* (Reference 8) and the whole report has been published in Reference 5c.

The American National Standards Institute (ANSI) is divided into divisions; the X3 division is concerned with standards for computers and business machines. X3, in turn, has a number of active committees studying various proposed standards, plus a "broad look" committee, the systems planning and requirements committee (SPARC). In 1972, SPARC established a study group on data base systems. The charter of this study group was to determine whether or not the data base area was ready for standardization.

After much discussion and definition, the group decided that the only things standardizable are the interfaces. These include the man-machine interfaces, the interfaces between software functions, and the interfaces between hardware functions.

The study group developed a concept of what a data base system of the future might look like, from which a series of over 30 interfaces was identified. The concepts of the overall system are too complex to discuss in this brief summary, but following are some of the highlights.

Administrators. The overall function of data administration was divided into three components by the study group, primarily to support data independence. The *enterprise administrator* is a business-oriented individual, perhaps on the staff of the executive vice president of the firm. The role of this individual (or perhaps more than one person) is to say, "Here is how management would like you to model this enterprise." This is the organization's view of the data, from which a conceptual schema (oops, conceptual plan) of the enterprise is recorded. This plan deals with real world entities—people, products, money, etc. It describes them in terms of their attributes and relationships. The *application system administrator,* using the conceptual plan, develops the "external data base plan"—the data definitions and relationship definitions as seen by the application programmers, report specifiers, query specifiers, and update specifiers. This plan is essentially the same as the sub-schema concept in the CODASYL specifications. Finally, the *data base administrator* creates the "internal data base plan," defining the way the data is stored in the data base to achieve performance and economy objectives.

It is the conceptual plan of the data base that enhances the concept of data independence, says Bachman. There is one such plan and it remains relatively stable, changing only when management's views of the business change. There are a number of external plans, as used by the user programs; each of these must be transformed (mapped) to the conceptual plan. Then there is one or more internal plans, dealing with the physical storage of the data in the data base; a map-

ping to the conceptual plan must be provided for each one.

In such an arrangement, says the study group, if new external plans are added or existing ones changed, all that need be done is change or add external-to-conceptual mappings. Similarly, if the data base administrator changes the physical storage of the data to improve performance or save space, all that is needed is a new internal-to-conceptual mapping. The conceptual plan provides a fixed point of reference, isolating changes in programs and changes in the physical storage of the data from each other.

If this concept of data base organization is accepted, then the 30-odd interfaces identified by the study group are candidates for standardization. The final report of the study group does not propose specific actions, part of the reason being that the group got hung up on the relational-versus-CODASYL (or the "CODASYL-versus-anti-CODASYL") debate at that point. The group did acknowledge that the CODASYL specifications represent an approach that can be implemented with today's technology.

CODASYL End User Facility

As we indicated above, the CODASYL specifications were aimed at the application programmer and involve a level of detail about the same as the COBOL programming language. The original task group (DBTG) that developed these specifications recognized that an "end user" interface would be desirable, for serving non-programmer users, but deferred action on the question.

This area is now being studied by another CODASYL committee, the end user facility committee. A progress report of this committee's work is given in Reference 5b.

What this committee proposes is a forms oriented approach. Thus, samples of forms would be used to illustrate what data is in the data base and what the data relationships are. Output would occur in certain user-designated forms. Note that these forms need not be just printed on paper; they can be displayed on a terminal. The forms approach was selected because end users as a class are very familiar with forms; they are part of most everyone's life.

The proposal is just at the beginning of the review and debate process, somewhat like the CODASYL data base specifications were in 1967-68. So it is likely that it will be a number of years before an end user facility is officially adopted by CODASYL.

CODASYL Data Definition Language

The CODASYL Data Definition Language Committee is developing specifications for a general DDL that will be compatible with many programming languages. The committee is very active, meets regularly, and has several active working groups. The committee published a Journal of Development in 1973. Since then, there have been enough changes made to the specifications that some committee members are now arguing for an updated version of the JOD. Others feel that some important changes are imminent, so that publication should be delayed for a time.

In brief, progress is being made toward a common data definition language.

Other determinants of data base directions

The working conference held in October 1975 (Reference 6) considered factors in addition to standards that would shape data base technology of the future. These included user experiences, audit considerations, existing and proposed government regulations, and the likely developments in new technology. We will briefly discuss the last three of these.

Audit considerations

The working panel on audit considerations discussed not only the impact of data base technology on the auditor but also how audit needs are likely to influence data base technology.

The panel observed that the concept of separation of duties, so basic to an auditor's ideas of internal control, probably is violated by a central data base. This observation might imply that auditors would be much happier with a well designed distributed data base than with one central data base.

Closely related to this point, the "going concern" concept was singled out as an important problem area. If a company loses its data base and cannot recover that data base, it might be out of business in a hurry. The auditor is expected to look for real risks that threaten the continuity of operations. Further, the external auditor might feel compelled to note such a risk in the certification statement for the financial reports. So the

"going concern" concept might well argue for the distributed data base, as opposed to the central data base.

The working panel recognized that EDP auditors were having a hard time learning conventional computer technology, and that advanced technology such as DBMS only compounded the problems of training these people.

Current audit software, in the main, cannot cope with the data base. The auditor needs an independent interface to the company's files of records, to make sure he gets *all* the desired records. With conventional systems, today's audit software can provide that interface. A similar interface is needed for data bases.

Audit functions may have to be built into DBMS and/or into the application systems.

Also, while auditors do not seek the responsibility, they may be asked to perform compliance tests to see that government regulations are being followed. Such regulations include the existing and proposed privacy laws.

Two reviewers of our draft commented at some length on this discussion of audit considerations. In brief, they questioned whether integrity was a matter of centralization versus decentralization of the data base. They pointed out that the question of integration versus non-integration of the data is also important; an integrated data base might be more difficult for the auditors to audit. In any case, they felt that regardless of approach, the proper integrity features should be designed into the systems.

Government regulations

How might existing and proposed government regulations influence the use of data base technology? This question was addressed by another panel at the October 1975 working conference (Reference 6).

The panel identified 20 areas in which government regulations are anticipated to impact information systems. Some of these regulations might impact data base systems in a special manner; that is what the panel looked for.

These 20 areas included system certification, information protection, limits on the interrelating of data, access authorization, continuity of operations, and others.

The panel identified those aspects of privacy legislation that promise to have the greatest impact on data base usage. For instance, when some personal information is changed, the need to notify all past recipients of that information about the change poses a non-trivial technical challenge and a significant cost challenge to data base users.

The working panel pointed out that a DBMS might well make it less costly for an organization to adhere to new or changed regulations. If data is scattered throughout numerous application files and processed by many programs, changes to the data and programs are much harder to implement than if the data is all in a data base. In fact, with increasing data independence, the overall impact of changes should be much less in a data base environment.

Likely developments in technology

The evolving technology working panel (Reference 6) considered the developments that were most likely to occur within the next five to ten years, as well as the developments that are most needed and on which research should be supported.

To make data base systems more usable, the panel saw a need for several developments. One is a formal methodology for design and restructuring; right now, these are done by trial-and-error which really is not adequate. Tools for tuning the data base, to improve performance, are needed and in fact are being developed. Improved integrity, fault detection, and recovery facilities are needed, and again progress is being made in this area (but perhaps not as much as technologists are capable of doing). Also, improved data independence is needed by means of a greater isolation of physical data from application programs. As we indicated earlier in this report, technologists are well aware of this need and it remains to be seen how well suppliers implement it.

In the area of data base architecture, the panel visualized two kinds of distributed intelligence related to the data base. One is a back-end processor for handling all data base management functions. In the same way that communications front-end processors have taken most or all communications functions out of the host CPU, so might a back-end processor take over most of the DBM functions. The other form of distributed intelligence is the storage hierarchy controller, which would control the movement of data between levels in the storage hierarchy (cache

memory, main memory, fast access random memory, disk storage, mass storage). This controller would also have responsibilities in the area of integrity and recovery. We have been cautioned, though, that a storage hierarchy controller concept involves substantial problems.

The panel noted that numerous suppliers are working on the concept of a distributed data base system. The group expected that distributed data base systems will be commercially available within five years.

In the area of data models (methods of expressing relationships), the working panel identified at least five models of major interest—network, hierarchical, relational, binary association, and set theoretic. Only the first two of these are available in the commercial marketplace. The panel made a most interesting point: none of the data models appeared to be "best" and it was hard to conclude which one will be considered "best" five years hence. For instance, it will take about five years to gain enough experience with the relational models to determine whether they are to some degree more useful than previous technologies.

Determining which model is "best" is a complex question of relative efficiency, said the panel. The ease of use of a particular model by a user may be the key factor, both now and in the future.

It should be noted that this panel consisted of 13 invited participants representing most of the main schools of thought on data base technology. Among them were the creators of some of today's most advanced data base technology. So the panel's consensus views carry considerable weight, in our opinion. We hope we have captured the gist of these views but interested readers are referred to the report (Reference 6).

What the panel was saying, we believe, is that there are no big surprises just over the horizon, as far as data base technology is concerned. There might be some interesting "packaging" of existing technology, but do not expect any breakthroughs.

The panel also identified a number of areas in which research is being conducted. While the potential value of this research might be great, it is unlikely to affect the use of data base technology within the next five years—because of the research, development, and implementation times involved. One area of research is "data semantics"—the meaning assigned to the data by the users which is not conveyed by the physical representation. It would be desirable to add semantics to the data, in an attempt to get each user to give the same interpretation to the data. Data semantics might not only help to avoid meaningless operations (such as adding weight and time) but might also allow the system to make inferences from the data.

Another area of active research is in natural language query systems. Currently, all query systems require the user to state the query in a formal manner—even though some sales literature claim that the queries can be expressed "almost" in the English (or other) language.

As we say, these are interesting areas of research from which practical results may not be expected for a number of years.

Selecting a DBMS

What is the main message of these conferences, working sessions, and papers? The message that we have received is the following. Today's technology is a fair indicator of what you will be seeing in the next five years. There will be continued progress but no radical changes. If you have been holding off on using data base technology waiting for that "significantly better system," you probably are wasting some opportunities as well as making eventual conversion more costly. Use today's technology to solve today's problems. But start small with data base technology and then work your way gradually toward larger, more complex systems. And do not forget audit requirements as well as existing and expected government regulations.

Since the great bulk of today's computer sites are not yet using a DBMS, the selection of a DBMS becomes an important decision area.

Mayford Roark pointed out that the selection of a DBMS generally followed the selection of hardware. In the majority of situations, that is true. The hardware is either already in place or is selected for reasons other than the available DBMS. But in a few instances, we have encountered users who made the DBMS selection *first*, and from this came the hardware decision.

How can an organization go about selecting a DBMS effectively? As we have tried to point out in this report, the subject area is complex. Luckily, the CODASYL Systems Committee has prepared a report that can help the user organization consider the essential factors when making this deci-

sion. The report is entitled *Selection and Acquisition of Data Base Management Systems* (Reference 9).

Space will not allow us to go into more than a brief overview of this report. If you are considering getting a DBMS for the first time, or are expecting to upgrade from your present DBMS, we think you should read this report. It could help you avoid some of the problems that others have encountered when converting to a DBMS.

It really is not possible to give a summary of the report. In essence, it is a large, annotated checklist. It raises points that you should review and consider, to see which ones apply to your situation.

The report has six sections: (1) a rationale for installing data base systems, (2) user needs, (3) primary capabilities of DBMS, (4) relation of DBMS to other system software and to hardware, (5) the selection process, and (6) how the evaluation team might be organized.

The main message that comes through from the report is, of course, that the selection and installation of a DBMS is a complex process. There are many points to be considered—not all of which need apply in a particular instance, however, But each point should be analyzed to see if it *does* apply.

To illustrate the depth of the report, here are some of the topics that are treated under "user needs." *User characteristics* include the types of expected users, the number and geographic dispersion of the users, types of access needed, response time needed, and degree of data sharing expected. *Data characteristics* include current volume and expected growth in the data, volatility, structural complexity and volatility, and geographic distribution. *Program characteristics* include current volume of computer programs, plus the growth in volume and the obsolescence of programs, volatility, size, and modularity. Finally, *support services needed from the DBMS* include services for system programmers, the data base administrator, and operations, plus the need for the DBMS to interface with other software packages and the need for the DBMS to have recovery capabilities. The report discusses each of these points.

In some environments, the selection and installation of a DBMS may in fact be relatively simple. Such situations occur when there is only one DBMS available for the particular hardware, where one fairly small application is being converted to the data base, and where the data structures are not complex. Experience with such a case may give the erroneous impression that all data base systems go in this easily. The CODASYL System Committee's report can help point out the complicating factors that can come into play in a more typical situation.

We have pointed out in this report that data base technology, and the use of that technology, is in the early part of stage 2, the proliferation stage. This is the trial-and-error learning period and it plays an important role. But if an organization stays too long in stage 2, waste and difficulties become magnified. The objective is to gain the basic lessons from stage 2 and then get into stage 3.

It looks to us as though DBMS users ought to be thinking of getting into stage 3. There is a variety of types of DBMS in use. Improvements will occur but no great breakthroughs are expected. So start looking for common or standard solutions that embody the best of what has been learned to date.

REFERENCES

1. Gibson, C. F. and R. L. Nolan, "Managing the four stages of EDP growth," *Harvard Business Review* (Soldiers Field, Boston, Mass. 02163), January-February 1974, p. 76-88;
2. *Data Base Systems,* Infotech International Limited (Nicholson House, Maidenhead, Berkshire SL6 1LD, U.K.), 1975, price £75.
3. *ACM Computing Surveys* (ACM, 1133 Avenue of the Americas, New York, N.Y. 10036), March 1976, Special Issue on Data Base Management Systems; price $8.
4. Curtice, R. M., "Data independence in data base systems," *Datamation* (1801 S. La Cienega Blvd., Los Angeles, Calif. 90035), April 1975, p. 65-66, 71.
5. *FDT, The Bulletin of ACM SIGMOD* (ACM, address above), price $2 each:
   a) Vol. 7, No. 3-4, 1975
   b) Vol. 8, No. 1, 1976
   c) Vol. 7, No. 2, 1975
6. Berg, John L. (ed.), *Data Base Directions, The Next Steps,* report of a working conference jointly sponsored by ACM and the U.S. National Bureau of Standards in October 1975. Order from the Superintendent of Documents, U.S. Government Printing Office, Washington D.C. 20402; Cat. No. C13.10:451, September 1976; price $2.40. (Also available from ACM, address above.)
7. Bachman, C. W., "Trends in data base managemen," in *Implementations of CODASYL Data Base Management Proposal, October 1974* (British Computer Society, 29 Portland Place, London W1N 4HU, U.K.
8. "The ANSI/SPARC Report, a synopsis," *Database Journal* (322 St. John Street, London ED1V 4QH, U.K.), Vol. 6, No. 11, 1976; price £5.50.
9. CODASYL Systems Committee, *Selection and Acquisition of Data Base Management Systems,* 1976; price $12. In U.S., order from ACM (address above). In Europe, order from IFIP Applied Information Processing Group (40, Paulus Potterstraat, Amsterdam 1007, The Netherlands).

Common application systems, designed to meet the needs of many users, have been sought for many years. There have been numerous unhappy experiences by pioneer users who have found that they have had to either completely revamp or fully replace common systems. But recently some notable successes have occurred, to the point where common systems may at last be viable solutions. Next month we will discuss how some companies have reduced the duplication of development costs, cut maintenance costs, and have made system outputs more comparable by way of common systems.

SUBJECTS COVERED BY EDP ANALYZER IN PRIOR YEARS

1973 (Volume 11)

Number

1. The Emerging Computer Networks
2. Distributed Intelligence in Data Communications
3. Developments in Data Transmission
4. Computer Progress in Japan
5. A Structure for EDP Projects
6. The Cautious Path to a Data Base
7. Long Term Data Retention
8. In Your Future: Distributed Systems?
9. Computer Fraud and Embezzlement
10. The Psychology of Mixed Installations
11. The Effects of Charge-Back Policies
12. Protecting Valuable Data—Part 1

1974 (Volume 12)

Number

1. Protecting Valuable Data—Part 2
2. The Current Status of Data Management
3. Problem Areas in Data Management
4. Issues in Programming Management
5. The Search for Software Reliability
6. The Advent of Structured Programming
7. Charging for Computer Services
8. Structures for Future Systems
9. The Upgrading of Computer Operators
10. What's Happening with CODASYL-type DBMS?
11. The Data Dictionary/Directory Function
12. Improve the System Building Process

1975 (Volume 13)

Number

1. Progress Toward International Data Networks
2. Soon: Public Packet Switched Networks
3. The Internal Auditor and the Computer
4. Improvements in Man/Machine Interfacing
5. "Are We Doing the Right Things?"
6. "Are We Doing Things Right?"
7. "Do We Have the Right Resources?"
8. The Benefits of Standard Practices
9. Progress Toward Easier Programming
10. The New Interactive Search Systems
11. The Debate on Information Privacy: Part 1
12. The Debate on Information Privacy: Part 2

1976 (Volume 14)

Number

1. Planning for Multi-national Data Processing
2. Staff Training on the Multi-national Scene
3. Professionalism: Coming or Not?
4. Integrity and Security of Personal Data
5. APL and Decision Support Systems
6. Distributed Data Systems
7. Network Structures for Distributed Systems
8. Bringing Women into Computing Management
9. Project Management Systems
10. Distributed Systems and the End User
11. Recovery in Data Base Systems
12. Toward the Better Management of Data

(List of subjects prior to 1973 sent upon request)

PRICE SCHEDULE

The annual subscription price for EDP ANALYZER is $48. The two year price is $88 and the three year price is $120; postpaid surface delivery to the U.S., Canada, and Mexico. (Optional air mail delivery to Canada and Mexico available at extra cost.)

Subscriptions to other countries are: One year $60, two years, $112, and three years $156. These prices include AIR MAIL postage. All prices in U.S. dollars.

Attractive binders for holding 12 issues of EDP ANALYZER are available at $4.75. Californians please add 29¢ sales tax.

Because of the continuing demand for back issues, all previous reports are available. Price: $6 each (for U.S., Canada, and Mexico), and $7 elsewhere; includes air mail postage.

Reduced rates are in effect for multiple subscriptions and for multiple copies of back issues. Please write for rates.

Subscription agency orders limited to single copy, one-, two-, and three-year subscriptions only.

Send your order and check to:
EDP ANALYZER
Subscription Office
925 Anza Avenue
Vista, California 92083
Phone: (714) 724-3233

Send editorial correspondence to:
EDP ANALYZER
Editorial Office
925 Anza Avenue
Vista, California 92083
Phone: (714) 724-5900

Name_____

Company_____

Address_____

City, State, ZIP Code_____