EDP ANALYZER

OCTOBER, 1976
VOL. 14, NO. 10

DISTRIBUTED SYSTEMS AND THE END USER

Numerous reasons account for the growing interest in distributed systems, we believe. For instance, one reason is the reduction in overall operating costs that these systems promise. Another reason is that they give each organizational unit the resources to do its own data processing—to "control its own destiny," in the words of one executive. At the same time, distributed systems may well bring radical change in the whole data processing environment, as we know it today. Almost all aspects of the data processing function—system development, programming, data entry, and computer operations—may shift in the direction of the end user. Here are the experiences of some organizations where such a shift has already started. They may represent what tomorrow's data processing function will be like.

Interesting changes have been taking place in recent years in the data processing activities of Citibank N.A., at the bank's headquarters in New York City. Citibank is the second largest bank in the U.S. in terms of assets and first in terms of net income, according to *Fortune* magazine.

What Citibank has been doing is to gradually remove the big computers at headquarters and transfer the application systems to mini-computers located in the various segments of the bank. This process has been under way since 1974.

Citibank still has some large computers operating in a multi-programming mode, of course. As of the time of writing, they had three large IBM 370 computers plus a DECsystem-10 for in-house time sharing. Also, they still have "central" computers at satellite centers outside of New York City. But the general trend is away from such systems.

Three factors account for this change in policy.

Basic change in the bank's operating philosophy. A few years ago, Citibank used the conventional organization and operating philosophy followed by most banks—that is, a functional organization of the operating activities. For instance, there was an Item Processing Division that processed all checks, debits, and credits for input to the demand deposit accounting system.

Now Citibank has organized its operating activities in terms of marketing segments. Such segments include bank retail services, corporate services, governmental services, and so on. In turn, many of these are sub-divided; demand deposit accounting has six segments, for instance. Presently there is a total of 42 segments in the operating activities at headquarters.

Management of each segment has been given total control of the production operation of that segment—and this includes the data processing operation. Hence the use of mini-computers in the various segments.

"Economy of scale" arguments aren't valid. Bank officers make two points here. For one thing, mini-computers represent a new state of the art in computing technology. They provide more processing per dollar expended than do the older, larger computers. Secondly and perhaps even more important, a serious imbalance exists

between a large central machine and the operating staff. The staff is forced to meet the machine's production schedules and deadlines. People efficiency is sacrificed in the name of machine efficiency. Mini-computers provide a better balance between people processing and machine processing.

The cost of running the bank's processing operations has not increased since 1970, even in the face of inflation and a doubling of the number of transactions. The main reason has been breaking the big activities into segments, so that planning, directing, and control can be done effectively. The economy-of-scale claim for big systems is a mirage, they say.

Concern for data security. While not the main motivation for the move to minis, nevertheless minis *do* serve Citibank's data security interests. The bank processes transactions involving huge sums of money. With today's technology, there is just no way to assure that one program in a multiprogramming job mix cannot interact with another program in that mix, either accidentally or as a result of a deliberate fraud attempt. The risk of monetary loss possible in a multi-programming environment casts doubt on the ultimate value of the technology of multi-programming, they say. The isolated minis in the various segments are run, in general, in an off-line batch mode. Today's mini-computers make this policy economically feasible.

Of course, the bank does have on-line services for answering queries on customer account balances and such. But in present systems, no on-line updating of major asset files is possible.

In brief, at Citibank the data processing function is moving toward the end users—that is, toward the bank's operating segments that directly serve its markets.

A large manufacturing organization

A large U.S. manufacturing organization, with operations in Western Europe, has taken an interesting step in providing data processing services to its organizational components. At first glance, this company's approach appears to be the opposite of that of Citibank. We will show, though, that the two approaches are complementary.

Several years ago, this company had some 65 data processing locations in the U.S. and Western Europe. The greatest number of these locations used IBM 360 model 20s and 30s, but there also were some IBM 370s in use—135s, 145s, 155s, and 158s.

After studying the problem of growth in data processing costs, this company started to move toward a central data processing service that was tied to all company locations by data communications. What has evolved in the past few years is *one* center to serve all company locations in the U.S. and Western Europe. Multiple levels of minicomputers are used in the network for performing network functions and local processing and data storage. But the bulk of the processing and data storage is done in the main center.

System development and maintenance have not been centralized; each location is responsible for its own systems. Also, there has been no attempt to impose centrally developed standard systems on the various locations. Rather, the company just provides the various locations with computer power.

What have been the economic results of this network? In three years' time, the direct operating expenses for computer equipment have been cut more than 50% while during the same period workload increased over 20%. The bulk of the savings have come from the reduction of operating staff costs and space costs that had existed with the many stand-alone centers. Savings have amounted to many millions of dollars per year.

What seems to be happening?

At first glance, the above two cases seem to be 180 degrees out of phase. However, we do not think that is the case.

First of all, Citibank's new operating policies could have been supported by remote batch terminals located in the different segments of the bank and connected to one or more large, central computers. Each segment manager would have had control over the data processing work of his segment. But Citibank people make two points against this approach—the difficulty of managing a huge, centralized data processing center and their concern for data security.

At the manufacturing company, about 95% of the processing and data storage is currently done at the central site, with results delivered to the outlying locations via the network. The other 5% of the processing and data storage are done at network nodes. Perhaps the main reason for this allocation of load is that the minis being used are not the latest technology and do not have the cost/performance benefits of today's minis. A vice president of the company predicts that within ten years, as the existing minis are replaced with newer ones, the bulk of the processing and data storage will shift outward. Only about 20% will be done at the central site, he thinks, and the other 80% will be at or close to the end user sites. So the *structure* of a distributed system has been established.

What *is* common between the two cases is the fact that the tools of data processing have been put in the hands of the end users in both cases. The managers of the components of each of these companies now control the resources for getting their data processing done.

But, you ask, isn't something wrong here? One company has put in minis at its headquarters site and the other has put in an intercontinental centralized system—and both are claiming benefits compared with their previous operating methods. One approach or the other ought to have greater benefits.

Our opinion is that if the manufacturing company had used the latest technology in mini-computers (and it was precluded from doing so, for reasons we will not discuss), then much more of the computing power would have been distributed. In that case, the two approaches would have looked much more alike. Moreover, the vice president of the manufacturing company predicts that not too many years hence, they *will* look more alike. In the meantime, the centralized system has provided savings over the former policy of stand-alone, decentralized systems.

Wait a minute, you say, I'm not ready yet to accept the argument that economy of scale does not work any longer. Give more basis for this argument.

In the mid-1950s, Dr. H. R. J. Grosch proposed his well-known "law"—to wit, computing power increases as the square of the price. If one computer costs twice as much as another, it will have four times the computing power, says Grosch. In the intervening 20 years, there has been considerable debate on the validity of this "law." Here are some of the factors that distort the "law."

Amendment needed. At the least, the "law" should be amended to read: *in a given state of the art,* computing power increases as the square of the price. Mini-computers represent a new state of the art as compared to today's conventional main frames. Micro-computers may have a similar relationship to minis. Our field is changing very rapidly and the state of the art does not stay fixed for very long.

Cumbersome operating systems. Today's big operating systems use up much of the raw power of the larger computers. True, the larger operating systems do more for the user. But their generality has made them very complex, and complexity usually slows things down.

Input-output limitations. Much commercial data processing tends to be input-output limited. Internal computer speeds are increasing faster than input-output speeds. So commercial data processing may not really exploit the higher internal speeds of the larger machines.

Change in operating conditions. Mini-computer systems may have very few computer operator functions—all files stored on disks, only an occasional journal tape to be changed and printer paper to be loaded. Contrast this case with the large computer center that still does mainly magnetic tape processing. Many operators are needed. A tape library and librarian function are needed. An expensive site with environmental controls is needed. In such a situation, the operating costs would distort the economy of scale picture.

This argument works the other way, too. If one compares multiple free-standing, third generation computer sites with one centralized, third generation site, the centralized site probably will have less operating costs than the total of the multiple sites. This is why the manufacturing company discussed above achieved savings with its centralized system.

So the economy of scale picture looks as follows. Within the third generation (or third-and-a-half generation) of computers, economy of scale of processing costs for commercial work is not readily apparent. Users move to larger computers to get more capacity but processing costs do not drop much. However, by consolidating many free-standing centers into one or a few larger centers, operating costs may be reduced. This appears to be a cost reduction due to economy of scale in the computer—but it should be recognized for what it really is.

When comparing mini-computer systems with third (or 3½) generation computers, the minis have the advantage of a later state of the art. Mini systems save on processing costs, operator costs, and space costs. Data storage costs may be higher, on a per-byte basis, but are not great enough to offset the other savings.

What, then, seems to be happening? It looks to us as though the economic pressures exerted by mini-computers will gradually take the workload away from large, central computers. Processing and data storage will shift in the direction of the end user.

As this shift occurs, one may ask: what will happen to system development? To programming? To data entry? To computer operations? To get a clue as to what might happen, let us now consider the experiences of some selected organizations.

What may happen to system development?

A fairly common problem in the commercial applications of computers has been the users' dissatisfaction with the systems created for them by the data processing department. The systems may be close to what the users want—but they are not quite right.

In an attempt to solve this problem, many computer using organizations have tried to get user department management more involved in the system development process. The idea here is, if the end users help develop a new system, they are more likely to be satisfied with the outcome. But there have been problems with this solution. User department managers may resist getting involved deeply over an extended period of time, or may fear that they will be blamed for a failure, and end up giving only token support. Also, data processing people, who may be eager to use the latest technology, might try to manipulate the situation.

Here is the experience of one organization where the efforts to bring end users into the picture have been working reasonably well.

Public Service Board of N.S.W.

The Public Service Board of New South Wales, with headquarters in Sydney, Australia, provides the computing services for the NSW state government. It has been providing centralized computer services for NSW state government departments since 1962.

Originally, for the support of new application systems, the PSB assigned system development teams to work in the various departments. By the early 1970s, however, PSB management was convinced that a change was needed. User department management was too often dissatisfied with the systems being developed for them. What often emerged from the development efforts was the data processing people's idea of the application system, not departmental management's idea.

So in 1973, a reorganization and a redirection of the system development effort occurred. There were two main policy points established:

Responsibility. The responsibility for the development of an application system was clearly assigned to the management of the requesting department. This responsibility is to exist from the initial request on through the complete life cycle of the system.

Technical help. When the computer is to be used in a new application system, the technical design and construction of the computerized part of the system are to be done by well qualified people. To perform this function, a central specialist group was set up by bringing together most of the people who had previously been assigned to the departments.

Under this new policy, the initiative for the request for a new system (or a change to an existing system) *must* come from user department management. Department people can talk to anyone they like, including the central specialist group, or vendors, or others, to help them visualize a solution. The department then prepares a proposal for the new system, including the ideas on a solution.

The proposal then goes to three review groups. One is the consultant and review division, staffed with consultant-type people who are *not* data processing specialists. These people review the proposal for justification. They probe such questions as: Is the problem a real one? Does the solution look reasonable? Should a completely different approach to solving the problem be considered?

The second review group is the technical evaluation group. This is a small staff of hardware and software specialists. They look at the proposal from a technical standpoint, and probe such questions as: Does the proposed solution look technically feasible? Is another approach better? Can a solution be implemented without acquiring new equipment?

Both of these groups prepare their comments and submit them, together with the proposal, to an executive review panel. This panel consists of three executives, one of whom is the head of the department involved, another is the head of another department, and the third is a PSB board member. This panel normally has at best constrained enthusiasm (often approaching pessimism) about expected results of new computer systems, we were told. This is the group that makes the Go/No Go decision on the project. In making that decision, the panel may require that a large project be divided into shorter subprojects.

If the decision is to proceed with the project, a project is organized. The first two top people on the project come from the user department. The project leader is a very senior manager in the department. He does not become deeply involved in the project but is expected to make sure that things are going smoothly. The systems director is a manager who is experienced in the application area.

This person is responsible for developing the detailed system specs. The project is a full time assignment for him (or her) until the new system is running, after which he goes back to his old job.

The next two people on the project represent the technical side. One is the technical director—a computer specialist who is responsible for developing the computer system design within the specifications made by the systems director. The fourth person is the project coordinator who spends part time on the project on project management functions.

PSB executives stress that the criteria for success now are based on user satisfaction and not on the use of latest technology. But it was not an easy matter to get this attitude accepted by the data processing people, we were told.

About a year after the new system has been installed, a post implementation review is conducted. It is conducted by a team of three people, including one DP representative who was not involved in the implementation. This review probes to see how effective the new system is and how satisfied the user department is with it. The review team may give its blessings to the system, or may recommend modifications, or may recommend taking the system off the computer. The last is no idle threat. When these reviews were first started, all existing systems were reviewed—and the teams recommended that about 12 systems be taken off the computer. In fact, six of them were removed. The threat of removal caused user departments to clean up the other six.

This new approach to system development has been in use for about three years. There was resistance to the new ideas at first, but the difficulties are being smoothed out and the approach is working quite well now, we were told. More importantly, user satisfaction with new application systems is increasing.

So, in the state government of New South Wales, the end users are playing a much more active role in the development of application systems than they previously did. They initiate the projects, develop the specifications, make sure the project is running smoothly, and have final responsibility for the success or failure of the projects. The data processing specialists are brought in for the design and construction of the computerized portions of the new systems.

It seems to us that as the processing capability and data storage capability shift outward toward the end users, as distributed systems are installed, so will the responsibility for system development. This does not mean that every end user department can go its own way; proposal review procedures such as those used by PSB could insure compliance with policies and objectives. The benefits of this shift in responsibility, we believe, will be greater user satisfaction with the application systems and a steady stream of positive accomplishments.

There still will be a role for a central group of data processing specialists, of course, as in the case of PSB. But even this role will be changing somewhat, as part of the programming function is taken over by end users.

What may happen to programming?

Typically, there has been someone—programmers or trained staff persons, for instance—between the end users and the computers. The reason, of course, is that the end users have not been able to "speak the language" of the computers. Someone has been needed to translate the end user's desires into computer programs.

Now that picture is changing somewhat. We will discuss user experiences with two end-user-oriented systems.

REALITY system

Gene J. Goldberg, CPA, is a certified public accountant with offices in Northfield, Illinois, just north of Chicago. For the past four years, he has been using a computer to help provide accounting services for clients. In 1972, he installed an IBM System 3 card oriented system. By the end of 1973, the volume of work had grown to the point where he needed more capacity, so he began looking at a number of alternative systems.

What he finally selected, in mid-1974, was the REALITY system developed by Microdata Corporation (17481 Red Hill Avenue, Irvine, California 92705). The REALITY system includes a Microdata 1600 mini-computer, with 16K to 64K bytes of memory; 1 to 32 CRT terminals; 10 to 40 million bytes of disk storage; a 165 character per second to 300 lines per minute printer; and a 9-track magnetic tape unit. The system has a virtual memory operating system implemented in firmware. It has multi-programming capability, so that different terminals can be working on different jobs, and it has a spooling capability for printed output. Programming languages available include an assembler, RPG II, Data/Basic, and ENGLISH (a high level retrieval and report writer language).

In addition to the power of the system, an aspect that attracted Goldberg was Microdata's arrangement for marketing the system. It is sold through local dealers. Goldberg felt it important that someone be nearby to help him when he needed it. The dealer for the Chicago area is Systems Management Inc. (SMI), in Des Plaines, Illinois.

Goldberg contracted with SMI to write the programs for his accounting applications—balance sheet and income and expense statements, general ledgers, cash flows, payroll record keeping and quarterly tax returns, accounts receivable, accounts payable, and so on. The System 3 programs provided the starting point, and Goldberg specified what he wanted that would take advantage of the features of the REALITY system. In addition, he requested and obtained a 24-month contract with SMI to provide him with 16 man-hours programming time per month, to use as he chooses—maintenance, enhancements, or new applications.

Goldberg's system includes the processor (upgraded from 16K to 32K bytes of memory), disk storage (upgraded from 10M bytes to 20M bytes), CRT terminals (upgraded from two to four), a 300 lpm printer, and a magnetic tape unit. The upgrading has come about as transaction volume and data storage needs have grown. For a coupon redemption accounting application, severe peak loads occur; he has contracted with a service bureau to do the data entry function and provide him with a magnetic tape of the transactions which are then read into his computer.

But the point we wish to emphasize is that Goldberg, using the ENGLISH retrieval language, now does the bulk of the output "programming." For special reports, he can easily retrieve data from the disk files, display it on a CRT terminal, sort it as he wants it, perform arithmetic calculations on it, arrange it in report format, and so on—all with a relatively easy-to-use, free-form, English-like language. When he desires, he can direct that the output be printed on the printer. He can create ad hoc reports, as well as programs for repetitive reports, in this manner. Just as important, he and his office staff of three people can do these things "right now" as the need arises. They do not have to wait on the availability of a programmer.

Goldberg has also installed a data communications capability for his system—and the reason is somewhat surprising. When questions come up or difficulties arise with the programs, he just phones SMI. After describing the problem, he puts his computer on-line to their computer. An SMI specialist, operating his local computer, diagnoses and corrects the problem. Only when the system was first installed did SMI need to send anyone out to his office, Goldberg told us. Since then, all problems have been corrected via data communications.

The computer, disk storage, and tape unit all are in one small cabinet; this cabinet and the printer sit in one corner of one room of the office. The CRT terminals are located on four desks. There is no need for a raised floor or special air conditioning. Further, there is no need for a professional computer operator; Goldberg and his staff do all of the operating. In short, the computer is very much like another piece of office equipment.

INQUIRE system

E. R. Squibb & Sons, Inc., with headquarters in Princeton, New Jersey, is the pharmaceutical segment of Squibb Corporation. As with most pharmaceutical firms, Squibb has an extensive drug research program. The need for more efficient and effective literature searches in connection with this drug research has led Squibb to the use of a powerful information retrieval and data base management system, INQUIRE.

It was in 1969 that Squibb decided to mechanize the literature searching function, in support of their research and marketing activities. After investigating a number of packages and systems, they selected INQUIRE, developed and marketed by Infodata Systems Inc. (5205 Leesburg Pike, Falls Church, Virginia 22041). It is designed to run on IBM 360/370 equipment under any version of OS or VS. Initially, Squibb ran a much more limited version of INQUIRE than they now use on an IBM 360/40 with a 128K byte memory. Currently it is run on an IBM 370/155 located at the corporate computing center, some 15 miles away. INQUIRE is operated in any of three modes—on-line, remote batch, or batch. Batch applications are handled on an overnight basis.

After installing INQUIRE for the literature search application, Squibb found that the system had a wide spectrum of potential applications. Squibb now uses INQUIRE for such things as searching a personnel skills data base and searching a comprehensive data base of chemical and biological information on all Squibb chemical compounds.

We asked how the personnel department uses INQUIRE. The requirement for the personnel skills system arose because it was found that some job openings within Squibb could be filled by qualified people within the company—that is, if there were some convenient way to locate those individuals. With manual records, any such searches would be largely hit-or-miss.

To initially set up the personnel skills file, the automated payroll file was used to produce basic records for all employees. A questionnaire was then sent to all employees, requesting information on skills, education, foreign language capability, and such. COBOL programs were developed by the systems staff for editing and merging the data into the skills data base.

The systems department provides training in the use of INQUIRE for staff members in user departments. Typically one to three training sessions of about two hours each are needed. For relatively straight-forward searches of a single file, such as the personnel skills file, one training session may be sufficient. For multiple file searches of more complex data bases, more training is required.

When a job opening occurs at Squibb, the manager involved contacts the personnel department and discusses the job requirements. A staff member in personnel fills out a search request, using the English-like, free-form INQUIRE command language. Usually these requests are not so urgent that the remote batch terminals must be used, so the search requests are sent to the computer center by company mail. The output is returned the next morning; it lists the people already employed by Squibb who meet the criteria, together with summary data. These people are contacted through their department managers, to see if they are available and are interested in the opening.

A more complex type of search occurs in the chemical/biological data base application. Typically, research personnel want to search through all appropriate chemical and biological data about a particular class of compound. The output desired is a report showing the chemical properties of the compound and the results of all biological tests performed on it, so multiple files are involved in such searches. INQUIRE can perform a search with as many as 32 files, although Squibb has not found it necessary to search more than six files at a time.

At Squibb, the preprocessor programs that produce clean, updated files have been written in COBOL or PL/1 by company programmers. But from that point on, end users use INQUIRE to create the searches that retrieve and report information in the desired formats.

What about programmers?

What the above discussion indicates is that, with the use of high-level retrieval languages, the creation of retrieval/display/reporting programs is shifting to the end users. In this manner, the users are more likely to get what they want when they want it than if they had to use the services of programmers.

But a number of these high-level languages have facilities for data validation and file update—which means that they *could* be used for programming complete (perhaps simple) application systems. So the question might be asked: from what we know today, what is the likelihood that end users will take over the whole programming function?

Two factors seem to stand out in answer to this question:

One. A good programming language, by itself, will not allow "clerks" or "staff members" or even "managers" to write effective programs. Programs may involve substantial complexity—many branches, or complex sequences of events. No programming language that we have seen makes it easy to handle complexity; that is still up to the mental ability of the programmer. A good language perhaps does no more than not creating false obstacles when building the program.

Two. A good language, however, may allow a business oriented person who has an analytical mind to create at least some of the programs. So business system analysts, assistant managers, and managers, for instance, may find it practical to create programs for setting up their own files and maintaining those files.

In addition, there have been developments in what one would normally consider programming languages, that are (or could be) used by non-programmers. We discussed some of these in our September 1975 report. Here we present user experiences with two others, to illustrate what is happening.

Basic Business Language

We talked to a large international financial organization, with headquarters in New York City. The company has found that an end user oriented system is helping in the company's financial planning.

In recent years, a number of groups within the company have been using a variety of financial modelling packages and services. The groups use these modelling packages to help them analyze such things as budgets, the effects of changes in foreign exchange rates, sales projections, and so on. Generally these groups build and operate the models via commercial time sharing services.

Early in 1975, one group at corporate headquarters decided to look for a better package or language. What they finally selected was Basic Business Language (BBL), developed by Core & Code, Inc. (7 Trinity Court, Wellesley, Mass. 02181), and offered via the First Data Corporation time sharing network. BBL is also offered for in-house time sharing systems.

BBL is a financial planning, analysis, and reporting system that uses the English language syntax. Models can be created from terminals and the user has complete control over the analysis and reporting process. The people at company headquarters consider BBL to be quite a bit simpler to use than either FORTRAN or BASIC. At the same time, BBL can be used for programming reasonably complex routines.

As the company uses BBL, capable user department staff people are given training in the language. These staff people are not programmers but have the capability of being programmers. However, their interests lie in the business and not in programming computers. Using BBL, these people develop a variety of models for analyzing the company's business. They must program the models to validate and format input data, update files, perform the calculations involved in the model, and produce the output reports. We asked whether these models generally used data extracted from other application systems. Not generally, we were told; raw data is generally gathered for each model. In fact, in the case of budgeting models, it is model data that flows to other application systems such as the general ledger system.

So BBL has allowed the company to shift some of the programming function for financial modelling from the programming staff to business-oriented analysts.

ROBOT

Glaxo Holdings Ltd., with headquarters in London, is the largest pharmaceutical manufacturer in the U.K. The corporation controls some 60 operating companies located throughout the world. Annual sales are over £300 million, and total employment is some 30,000 people.

Corporate headquarters is concerned mainly with planning and analysis activities. An ICL 1902T is used to provide management information services for the board of directors and other head office departments. The data processing function has nine programmers and five system analysts. The environment in which Glaxo companies operate changes rapidly and there is need to respond quickly to change. The computer is used as a tool to help meet this need.

In 1973, the data processing staff started looking for a packaged system that would help them set up new data files more quickly and produce analysis reports more quickly. At that point in time, all programming was being done in COBOL, and that language was not serving the relatively fast response that was desired.

After some study, the company selected ROBOT, developed and marketed by a subsidiary of Software Sciences Group located in Macclesfield, Cheshire, U.K. ROBOT is a generalized information processing system incorporating its own problem-oriented language and a data base management facility. It was designed to be a portable, machine-independent system, first implemented to run on the ICL 1900 series of computers.

At the same time as they installed ROBOT, Glaxo decided to convert to a data base environment, operating under ROBOT. They have a variety of data types for serving multiple uses. These include summarized data on budgets and actuals from the operating companies and purchased survey data on medical statistics.

We asked whether staff members or system analysts used ROBOT for performing analyses and producing reports. No, we were told; even though Glaxo finds ROBOT much easier to use than COBOL, they use it as a conventional programming language. If an executive or manager wants a new report, he talks to a system analyst. The analyst draws up the report format and identifies the data that will be required. If a new type of data will be needed, authorization must first be obtained from the data base administrator. A programmer must then write a program for validating the new data, followed by a program for preparing the desired report. We got the impression, however, that non-programmers could use ROBOT for directly producing output reports, if they desired to do so.

Glaxo encountered some resistance from the programming staff to the use of ROBOT at the outset, primarily because it was considered to be a simpler, less professional language. After a number of months experience with ROBOT, however, none of the programmers wanted to switch back. Because programming is faster, a programmer can now do a complete job where before he might work on just one program in a set of programs. COBOL is still used for maintaining older programs but all new systems are being programmed in ROBOT.

It is possible, of course, that in time the system analysts will begin to take over some of the programming duties through the use of ROBOT. Should that happen, the programming function would have moved a step in the direction of the end user.

Research in end user languages

Research on the subject of end user oriented languages is being conducted at many places. Some particularly interesting research is going on at the IBM laboratories at San Jose, California.

Reisner et al. (Reference 1) report on a human factors evaluation study for two data base query languages, SQUARE and SEQUEL. One question addressed by the study was: how well can non-programmers learn to use a query language? Both of the subject languages are based on a relational model of data, which in concept is less demanding of the user than other more common models of data. SQUARE uses a mathematical notation while SEQUEL uses an English-like syntax. The study was conducted at a local university.

For the study, 61 undergraduate students and 3 graduate students were selected, representing 29 different majors (accounting, mathematics, fine arts, nursing, political science, etc.). The students were divided into four groups: two groups of "programmers" (those who had had at least one course in programming), one to learn SQUARE and the other to learn SEQUEL and two other groups of non-programmers (no previous training in programming), one for SQUARE and one for SEQUEL. The four groups were kept separate. Each was given classroom training in the assigned language, 12 to 14 hours in length. Five review quizzes were given during the instruction, followed by a final exam at the end of the training—and then a retention-of-knowledge test one week later. The quizzes and exams stated queries as English sentences, such as "Find the names of all employees who make more than their managers." The same queries were posed to all four groups. The students had to formulate the queries in the assigned language.

Some of the results were the following. All four groups were able to learn, use, and remember the assigned languages effectively. The people with previous programming training outperformed the non-programmers, often by a non-trivial amount. (This difference might well disappear after a period of actual use.) And the non-programmers did better with the English-like SEQUEL than they did with the math-oriented SQUARE.

Note that the college student population of this study was chosen to represent professional users—professional in the sense of accountants, lawyers, and managers. It did not attempt to study how well the languages might be learned by, say, clerical people.

Carlson et al. (Reference 2) discuss research in computer-assisted management problem solving and decision making, operating in an interactive mode. The system under development was GADS—Geo-data Analysis & Display System. Geographic and numeric data are presented on a CRT screen in the form of maps, overlays to maps, and graphs. Users can call up maps, create overlays, change boundaries within maps, and so on.

The system was tested by a case study at the San Jose Police Department. The problem being used in the case study: assigning policemen to beats. The problem is complex for a number of reasons, such as a varying number of police officers, respecting the natural and neighborhood boundaries, the need to level the workloads, and so on.

The study, conducted over a four-month period, involved eleven police personnel. None had prior programming knowledge. Five of these were the principal users of the system and were given eight hours of training. They then formed four teams of two to three people each. In total, the four teams spent over 200 hours constructing proposed beats.

In brief, these people were able to use the system effectively after eight hours of training. The solutions to the problem took longer than the manual methods then in use, but much more data was analyzed. The participants unanimously agreed that the system had improved their understanding of the problem and had given better solutions than the manual system.

In both of these studies, the systems were not just retrieval languages but rather could do quite a bit more, such as manipulate data, perform arithmetic calculations, sort data, and so on.

These studies would seem to reinforce what we have been saying earlier in this report. Capable end users *can* take on some of the "programming" functions.

What may happen to operations?

Data entry is another computer-related function that is moving toward the end user. In our June 1976 report, we discussed how several organizations, including the Firemans Fund Insurance Company and the Kennington Motor Group, have moved their data entry out close to the point of transaction.

We have talked to several service bureaus recently. All have been encouraging customers to install terminals (usually intelligent terminals) for performing the data entry function. Transactions may be stored on cassette for later transmittal to the service bureau—by data communications or by mailing the cassettes.

There are several advantages in having the end user perform the data entry. For one thing, the end user becomes responsible for the timeliness and accuracy of the input data. Intelligent terminals can perform the validation function during data entry, so that errors can be detected and corrected on the spot. Also, the data entry function may be performed by the regular clerical staff, who understand what the data means and who perform the function as a part of their duties. Full-time data entry is a hard function to staff and usually has a high turnover. Distributed data entry can help alleviate the problems. Finally, distributed data entry spreads out the workload both geographically and in time. The problem of peak loads is reduced.

So data entry, as one part of computer operations, is shifting in the direction of the end user.

Computer operations

Fisher & Paykel Ltd., with headquarters in Auckland, New Zealand, manufactures a wide range of home appliances, mainly for the New Zealand and Australian markets. The company employs about 2500 people.

In 1973, the company decided to install a comprehensive on-line order entry, stock control, invoicing, and accounting system for its spare parts department. Proposals were solicited from computer manufacturers and four were received. Of these, Fisher & Paykel selected the DEC PDP 11/40 system, operating under the MUMPS-11 operating system.

The configuration selected includes a 96K byte memory, 9.6M bytes of disk storage, six CRT terminals, two typewriter terminals, high speed printer, paper tape reader, and magnetic tape unit.

Since the original system was installed, Fisher & Paykel has added other applications. These include accounts payable, general ledger, payroll data entry, and accounting analysis. The system has been operating 12 to 16 hours a day, six days a week.

We visited a number of installations in Australia and New Zealand, with computers from most of the major computer manufacturers. Every installation except one that had pioneered something—the first to use a new computer or a new software package, in the Australasia area—said that they had gone through a very painful experience while the supplier's representatives learned at their expense. The one exception was Fisher & Paykel. The PDP 11/40 went in smoothly, the documentation was complete and correct, and the on-line software worked as it should. Only minor bugs were found, and only one of those caused any inconvenience. (The batch system software did not come up to expectations, and MUMPS has some shortcomings as far as commercial data processing is concerned, but in general the company is well satisfied.)

Until recently, the 11/40 used only a relatively modest amount of floor space. It was installed in one corner of the room housing the mechanical accounting machines. In a sense, it was treated much like any other office machine. The operations manager, who also supervises the mechanical accounting machine section, has one computer operator per shift. Since the system is primarily on-line, with the files stored on disk, the operator functions are not demanding.

With the growth of applications and volumes, more capacity was needed. So Fisher & Paykel has added a second 11/40, to be installed in a new administration building due for completion at the end of this year. Also, in March, they installed an ICL 2903 for conventional batch processing, taking over work previously performed at a service bureau. A computer room was built for the 2903, so the first 11/40 was transferred to that location.

The changing operating environment

Fisher & Paykel is another illustration of the fact that powerful mini-computer systems can be installed on a departmental basis, even in a medium size company. Special environmental requirements, such as air conditioning, humidity control, and dust control, are not particularly demanding. With on-line data entry and with files stored on disk storage, operator functions are reduced. Professional operators may be needed, of course, to get the system in operation again after a failure occurs. But it might be quite feasible for one trained operator to handle multiple departmental mini systems.

We also mentioned another approach—that of putting remote batch or keyboard terminals in the various departments, each tied to a central computer. If all data files were stored on disk or mass storage at the central site, most of the operator functions would shift to the terminals. Operators would be needed for getting the system running again after a failure. We discussed just such a system as this in our September 1974 report, concerning Copley Computer Services, Inc., in San Diego, California.

So with on-line terminals, either tied to a departmental mini or to a central system, and with most files on disk storage, the operating environment is changing. The operator functions are becoming less manual—no feeding of input cards, no changing of tapes, no adjusting of the workload. Instead, the main functions become the diagnosing of failures and getting the system back into operation after a failure occurs.

We do not want to give the impression that just anyone can do the computer operations function. But where a full time, professional operator just is not justified, the functions of fault diagnosis and recovery might be performed by a manager, assistant manager, or a sharp staff person who is interested in computers.

The shift toward the end user

We see a number of things occurring that are changing the conventional data processing environment.

The economics are changing. It is becoming economically feasible to give each manager the responsibility and the resources for his own data processing operations. It is no longer really *necessary* to have a "large" central computer to which batches of work are carried. Departmental minicomputer systems are here, are practical, and are in every-day use. An alternative, which may be preferred by some, is to place terminals in the departments, tied to a central system.

The users want more control. We continue to hear that "management is unhappy" about data processing's performance. Users would like to be able to control when their work is done and how it is done. This includes not only data entry and data processing production but also system development, programming of production programs, and the programming of special analyses and reports.

The end users would like to be able to do more of these things themselves, calling in the data processing specialists only when needed.

As data processing operations move out toward the end user, by way of departmental mini systems or by departmental terminals, so will more and more aspects of system development and programming move in this same direction.

Some aspects of programming are getting easier. It is now quite feasible for end users to perform data retrieval and reporting via the type of new languages and systems that we have been discussing. There really is no need for end users to call on the services of programmers for these functions any longer—that is, as soon as organizations are able to install such systems.

Similarly, capable staff people will be able to take over some of the remaining functions of system design and programming. Application systems for departmental minis will tend to be smaller and may well be easier to set up and easier to change, as compared with systems set up for central computers.

We do not see the end of the road in sight for conventional system analysis and programming. But that road might be getting narrower.

Computer operating needs are changing. The data entry function already has moved, in a significant amount, toward the end user. As departmental minis or departmental terminals are installed, the computer operations function will move toward the end user.

This shift toward the end user of most of the functions of data processing is an aspect of distributed systems we have not heard discussed much. We think it is an important shift, one that deserves more discussion and consideration.

Of course, all of this is not going to happen overnight. The "data processing department" is too deeply enmeshed in the fabric of most organizations for the shift to the end user to occur suddenly. Even if distributed systems catch on like wildfire, we suspect that data processing will not change much at most organizations during the next five to seven years at least.

But if we are seeing this shift correctly, then the concept of the "data processing department" *is* going to change—and perhaps not too far hence for some organizations. And if the concept of the department is going to change, so will the role and functions of data processing management.

We will try to address that subject in a future issue.

REFERENCES AND FURTHER INFORMATION

1. Reisner, P., R. F. Boyce, and D. D. Chamberlin, "Human factors evaluation of two data base query languages—Square and Sequel," *Proceedings of the 1975 National Computer Conference* (AFIPS Press, 210 Summit Avenue, Montvale, New Jersey 07645), p. 447-452; price $50 paper, $15 microfiche.
2. Carlson, E. D., J. L. Bennett, G. M. Giddings, and P. E. Mantey, "The design and evaluation of an interactive geo-data analysis and display system," *Information Processing 74* (in U.S., American Elsevier Publishing Co., 52 Vanderbilt Avenue, New York, N.Y. 10017, price $80; in Europe, North-Holland Publishing Co., P.O. Box 211, Amsterdam, The Netherlands, price Dfl 200), Sec. 5, p. 1057-1061.
3. For more on research on end-user oriented query languages, see "English as a query language, a debate," in *Proceedings of the 1972 ACM Annual Conference* (ACM, 1133 Avenue of the Americas, New York, N.Y. 10036), p. 1073-1078.
4. A conference on decision support systems, some of which include end user orientation, will be held at the IBM Research Laboratory, San Jose, California, on January 24-26, 1977. It is being sponsored by IBM, M.I.T., University of Pennsylvania, and ACM SIGBDP. For more information, write E. D. Carlson, K/54 282, IBM Research Laboratory, 5600 Cottle Road, San Jose, Calif. 95193.

With batch-type application systems using files on magnetic tape, recovery from failures has not been conceptually difficult. While in practice some recoveries have been complex (such as when the only backup copy of a file was destroyed), in theory recovery just meant going to a prior generation file and rerunning the work. However, with the arrival of data base systems—and particularly on-line data base systems—this rerun concept is far from adequate. New methodology has had to be developed for rapidly recovering from failures in data base systems. Next month we will describe some user experiences with these new methodologies.

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

(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 ____________________