EDP ANALYZER
Editor and Publisher: RICHARD G. CANNING

EDP ANALYZER

MAY, 1981
VOL. 19, NO. 5

'PROGRAMMING' BY END USERS

It is becoming increasingly clear that there are not enough system analysts and programmers to meet today's needs. One solution that we found both large and small companies taking is to install one of the new 'data management systems' that enable end users to perform some of their own programming—handling query and report requests and even some complete applications. These companies are already receiving impressive benefits from this type of use. To data processing management, this prospect may appear both tantalizing and threatening, yet proponents of end user programming see it as the way of the future. From what we found, they could be right.

The American Society of Corporate Secretaries, with headquarters in New York City, has a membership of some 2500 individuals who are, or who recently were, engaged as corporate secretaries. The Society employs 13 people in its headquarters office.

The Society's members are corporate officers who, in the main, work for large corporations. Their responsibilities include the issuance of stock certificates to new stockholders, and the maintenance of stockholder records.

The Society also maintains a list of approved 'nominees'—a rather complex function perhaps best defined by an example. When people have stock shares in a trust or account that is managed by a bank, that particular bank may register and hold that stock in the bank's name as nominee. When the customer or the trustee wants to sell the stock, the bank already has the certificate in its custody; the customer or trustee need not sign it and mail it in.

In the past, the Society had maintained these two lists at a service bureau. But costs were rising and the Society really wanted a faster turnaround than they were getting from the service bureau.

At about this time, the administrator of member services for the Society came across the CREATE turnkey computer system, developed by Complete Computer Systems of Horsham, Pennsylvania. CREATE runs on a Data General Eclipse or Nova computer with at least 64k of memory and the RDOS operating system.

ISSN 0012-7523. Multiple copy prices listed on last page.

CREATE includes software for defining files and records, for defining input screen formats and programs, for defining output reports, for handling most file maintenance activities, and other data management functions. The administrator liked what he saw, checked out the system with advisors from some member firms, and then recommended procurement of the system. The system was installed in January 1980.

The system that the Society obtained includes the Nova III with a 64k memory, 10m bytes of disk storage (5 fixed and 5 removable), two CRT terminals, and a character printer—plus the CREATE software.

There really was nothing to programming the system, the administrator told us; he did it all, and very quickly, for the membership and nominee files. Neither file requires complex application logic for its updating; most updating and maintenance can be handled by the CREATE functions. In fact, he demonstrated to us how, in a few minutes, he could define a completely new file, enter some records, and create a report from the file.

The Society continues to add new applications to the system—a file of participants at seminars, a file of Society regional meetings, a file of data obtained in a survey, and so on. For the files used in the daily operations of the Society, CREATE has made it possible to set these up quickly and to do the processing that the Society has required.

Dart Industries

Dart Industries, Inc. is a multi-national holding company with some 160 subsidiary operating companies. These include such well-known brand names as Tupperware, West Bend, Duracell, and Mallory. Dart recently merged with Kraft Foods and formed a new parent company, Dart and Kraft, with consolidated sales over $9 billion in 1980. Dart is in the process of moving its headquarters from Los Angeles to Northbrook, Illinois, a suburb of Chicago.

Dart Industries has a corporate information systems staff of sixty people. The corporate computer center has an IBM 370/158 and a 4341, which share some disk files; these computers operate under MVS and have TSO/SPF and CICS for in-house time-sharing and data communications. In addition, each operating company has its own data processing department. In this environment, the corporate group concentrates on developing the larger, more complex on-line applications for the holding company as well as for some of the operating groups.

In early 1980, the information systems department began a project to enhance their existing financial database with a more inclusive decision support system. A prime objective of the new system was to provide rapid response to users' requests for information.

The existing database, operating under IBM's IMS database management system, contains ten years worth of actual performance data for all operating companies, by month and by account (of which there are about 80). The database also contains planned performance data.

Up to that time, two people in the financial analysis department had been updating the information in the database by submitting batch updates. They also handled many report requests via already-written batch-type report programs that had been developed by the information systems staff. But a growing number of user requests were for reports that had not been preprogrammed. Information systems simply could not respond to these ad hoc requests fast enough.

So information systems investigated a number of data management systems and finally selected FOCUS, from Information Builders, Inc. FOCUS works under MVS and TSO (as well as other IBM operating systems), and it works with IMS files. It contains a non-procedural language for interactive query and report generation. Also, it has a dialog manager which allows pre-written procedures, such as requests for certain types of reports, to be easily run by end users; the users simply supply the necessary variable information at run time. The product is aimed at end users, and Dart wanted to shift the report generation work away from its information systems staff.

FOCUS was installed in mid-1980. Three Dart people were given extensive training on FOCUS because they were to become the in-house FOCUS consultants. One of these employees was the manager of financial information systems. The

other two were the manager and one of her staff from financial analysis—the ones who had previously submitted the database updates and report requests.

These three people then: (1) developed an in-house FOCUS training course for end users, (2) interfaced FOCUS to the existing database, (3) developed twenty standard report programs, and (4) had a number of IBM 3270 CRT terminals installed in user departments throughout Dart headquarters.

They then began training selected end users who had experience with computers and thus were most likely to pick up the use of FOCUS easily. The training course ran three consecutive mornings and one afternoon. The final afternoon was spent helping the attendees actually use the pre-programmed report dialogs as well as create one new report program for their own use. The three support people also provide any needed follow-on assistance to these users.

Once this group of end users was working with FOCUS on their own, the support group invited other users—superiors, peers and subordinates of the original group—to take the class. Dart has now trained about 25 users.

FOCUS is also used in information systems. An interesting use occurred because of the Dart and Kraft merger. The people at Dart and Kraft headquarters in Chicago wanted a number of financial statements from Dart. Previously a financial manager manually computed some figures, filled out some forms, and had the forms transmitted to Chicago by facsimile. The calculations were time consuming and the transmitted facsimile copy was often hard to read. So this procedure was not very satisfactory. Using FOCUS, the procedure was automated.

One programmer, along with the financial manager, designed and created the necessary FOCUS files, calculation algorithms, and report dialogs in less than one week. The financial manager now runs these report dialogs on the IBM 370/158 in Los Angeles in order to create the desired financial reports. These are then transmitted to Chicago via dial-up lines, where they are printed out on an IBM 6670 remote printing system.

The people at Dart are pleased with their use of FOCUS, both within information systems and within a growing number of headquarters departments. They feel that they can now respond more quickly to user requests by moving report generation, and some programming and maintenance, out to the corporate users. This allows them more time to concentrate on more complex systems development in information systems.

Scholastic, Inc.

Scholastic, Inc., with headquarters in New York City, is the largest publisher of supplementary classroom reading materials in the United States; the company produces about 200 million copies a year of their bi-weekly and monthly magazines. They also distribute about 100 million books a year, mostly bookclub books. Annual sales are about $135 million, and the company employs some 1700 people.

Scholastic's data processing is done on an IBM 370/158 located in northern New Jersey, across the Hudson River from New York City. While this computer operates mainly in a batch mode, over 100 CRT terminals are on-line for query purposes (mainly, to look up customer account information).

We talked to the vice president of corporate market research about his computer uses. He used to administer the data processing function and so has some computer background and has gained some perspectives on the (normally very lengthy) system development process. In his present position, though, he is on the end user side of the fence. Like most users, he wants to get new applications up and running quickly. Further, some of his applications involve not-well-defined needs. So he suspected that he would have to make use of an interactive computer facility, not a batch service, to get the results he wanted on the time scale he desired.

He looked around for a suitable facility, and selected NOMAD, offered by National CSS, a nation-wide time sharing service. NOMAD couples a very powerful data management system together with a very flexible and easy-to-use reporting system. So he signed up for and started using NOMAD in late 1976.

In his use of NOMAD, he has found two main types of applications where the usefulness of the system is outstanding.

Prototyping. In this use, he said, there generally are well-defined input and data specifications, but at the outset, output requirements are not well understood. The users are not quite sure what output information they want or how they want it formatted.

He cited an example of this situation—a sales call analysis system that was requested by the sales manager. Scholastic sales people make sales calls at schools to present one or more Scholastic products; at the end of each call, they fill out a sales call report log. This is the input information—well-defined, discreet, and prespecified. The purpose of the new application was to simply help sales management gain better insight, so as to assist these field sales people in their efforts.

Since NOMAD is so user-friendly, the vice president himself could define the data file structure for this new application in a few hours. Then data was entered, simply by keyboarding the data from all available sales call reports for the current year; this function was performed by Scholastic's data entry section.

With the data in the computer, the vice president used NOMAD to create the types of prototype reports that the sales manager thought might be useful. In fact, this initial report formatting took only a relatively few minutes of time of both the vice president and the sales manager. These reports were then critiqued by the sales manager and his staff.

As is usually the case, the sales department saw needed changes and improvements in the reports. Having these first versions to look at, the sales staff had something concrete to analyze. Working at a terminal, the *sales manager* requested some changes and was immediately able to obtain revised reports. After a few such sessions, usable and valuable operating reports were developed.

When the sales manager saw the need for additional information on a report that was not contained in the sales call reports or in the NOMAD database—for example, some information about the school districts where the sales calls were made—no problem was posed. A file of relevant school district information was set up by extracting the desired information from an available data processing file and batch loaded into a supplementary NOMAD file.

The elapsed time for getting these operations reports defined and the system set up to produce them was less than one week. Even if the output reports had been precisely defined at the outset, doing it by conventional programming methods probably would have taken months, said the vice president. They continue to keep this application on NOMAD because of the system's superb ad hoc analysis capability, he added.

"In fact," he continued, "in such cases where the data is fixed and specified, where the processing logic is straight-forward, and where you have a pretty good idea of the kinds of output reports desired (even if you do not yet know the exact formats), with NOMAD you can get such a system up and running in a week. There is nothing to it."

Developing system design specifications. The second area where NOMAD has performed well, according to the vice president, has been in situations where it has been impossible to specify the system at the outset—where the user cannot determine what the data input should be, or what the output data or formats should be, and so on. As an example of this situation, the vice president has been working on the design of a new marketing information system for Scholastic, to help management decide where Scholastic's marketing efforts should be directed.

The problem is, he said, the magnitude of the data and the difficulty of defining which elements of the data are the key elements. With the sales call analysis system, just discussed, there was a definite starting point—the data on the sales call logs. In contrast, with a marketing information system, he said, generally one does not know where to begin.

To illustrate, consider the magnitude of Scholastic's database. There are some 2 million teachers in the U.S., about 1 million of whom are active Scholastic customers. These teachers are employed by 16,000 school districts, with 125,000 schools. Further, there are about 1,000 different and available characteristics about each

school district that could affect or relate to the demand for a given education product. During the peak ordering season (in September, after schools open), the company can receive up to 30,000 orders per day, primarily for the 30 classroom magazines and for the five book clubs. Finding patterns in such a mass of data, for optimizing direct selling coverage, is like looking for the speck of gold in a steamshovel-full (not a pan) of dust.

For this application, all potentially relevant data was summarized from Scholastic's extensive data processing master files, and batch loaded into NOMAD. Using NOMAD's data management system and statistical routines, they started looking at the data in many ways.

The underlying concept of the system is based on 40 mathematically derived, homogeneous market segments for the 16,000 school districts in the U.S. These market segments were developed several years ago under a contract with a consulting firm that specializes in social areal analysis. The firm performed factor and cluster analysis on the previously mentioned 1,000 school district characteristics to obtain the 40 'natural' groupings.

Since Scholastic's sales and marketing data was loaded into and maintained by NOMAD in a sales/school/district/market-segment hierarchy, NOMAD could be and was used to generate (for example) literally hundreds of sales penetration indexes by the 40 market segments—such as sales per student, or orders per teacher. NOMAD was then used to select the richest ones, the ones that contain the most information about the demand for or use of a product.

With these optimized sales penetration indexes at hand, NOMAD was then used to associate these indexes with a few key district product demand characteristics (from the 1,000 characteristics available). This analysis provided the system logic for automatically identifying key target markets and explaining the nature of the demand for a given product within these target markets. As these results crystallized, the vice president stayed very much in touch with Scholastic's line marketing people; he used NOMAD to try different ways of reformatting the test results into easy-to-use marketing reports.

Such iterative procedures allowed the vice president to identify the key input, processing, and output elements of a viable marketing information system.

Because of the research nature of his function, the vice president makes extensive personal use of NOMAD. But this is no big deal, he says; with a few hours of training, any conscientious marketer can be taught how to use NOMAD to select data from a file, format it, and print out a report that highlights and aggregates the 'specks of gold.' Learning to use the statistical analysis routines takes a bit longer, but still is simple enough to do with NOMAD's assistance. He sees this kind of tool as a help in designing systems and making decisions where the 'rules' and processing algorithms just do not exist.

An emerging new world

As we discussed last month, 'end users' in some organizations are being encouraged to help redesign their jobs and to view computers as tools to help them in these redesigned jobs. By this, we are not referring to large or complex application systems that are developed by a programming staff; rather, we are referring to ways that employees of departments other than data processing seek to use the computer. The idea of job redesign is sure to spread—redesigning jobs to make them more enriched, more human—in order to achieve the productivity increases that are so sorely needed. In the course of this redesign, the employees will be seeing more and more ways in which the computer can help them.

The question then becomes: how are these new needs to be met? The negative-aspect answer is: almost surely *not* by today's conventional application development methods.

The three company case studies just presented give one solution—end user programming. They illustrate what we believe is a significant trend—computing power is being put directly into the hands of end users for the purpose of helping them perform their daily work. These end users may or may not be drawing on the large corporate data files, but they *are* creating and using their own files to keep track of the status of their own work. We see this trend happening in

small as well as large organizations, as our case studies illustrate.

Since the beginning use of computers, programming has remained pretty much the work of specialists. These professionals were needed to translate natural language to computer language. But now, this 'typical' way of getting applications developed is causing innumerable problems.

First, the development process is too long. Users get their custom-made systems months, or even years, after they request them. Even seemingly simple changes take a long time. And this time lag is lengthening, because users are asking for more and more applications, which are becoming increasingly complex. These eventually will require more maintenance, which could increase the backlogs of work even more.

Second, there is a shortage of specialists—programmers, system analysts, database administrators, and data communications experts. Demand has surpassed supply, and there is no foreseeable lessening of demand or dramatic increase in supply. This shortage of professionals has caused both salaries and turnover to skyrocket.

Because (1) our industry relies on expensive professionals, (2) they are supported with only a few automated aids, and (3) most applications are still custom programmed, the cost of in-house software is becoming exorbitant. Two possible solutions to these problems are: (a) increase the productivity of the professionals, and (b) make other employees part-time programmers.

Interestingly, there are some relatively new products, which we call data management systems (DMS), that facilitate both of these solutions. End users can use them to write many of their own programs, and programmers can use them to both create and maintain application systems faster.

In the past, programming has required extensive training and very precise logical reasoning. Certainly, few end users fit into this mold. But these attributes apply to programming using procedural languages, where the programmer must tell the computer *how* to perform the work step by step. With the emergence of non-procedural languages, the 'programmer' mainly tells the computer *what* to do. The system is already programmed on how to perform these functions. So with non-procedural languages, non-technical users can direct the computer on their own, without a programmer as a middleman.

We see end user programming coming; it promises just too many benefits not to catch on. It probably will be offered as another component of office automation, along with word processing, electronic mail, electronic calendars, etc. If the capability is not offered through data processing, users are likely to acquire their own machines to do it, with or without the blessing or help of the data processing department.

In order to not let the trend get out of control, we believe data processing should not only support it, but actively encourage it, by offering end user programming tools as well as training and assistance.

In some companies, end user programming is already here. In these organizations, simple programming is becoming a part of office workers' jobs. It is a new way to accomplish their work. Pens, pencils, calculators, typewriters, and paper are being (partially) replaced by computers.

End user programming is spreading fast in these organizations—more rapidly than anyone foresaw. From the employees' view, this growth is understandable. They discovered they can accomplish tasks more quickly than was possible manually. They can now easily retrieve information from their files. And they can perform analyses that were previously impractical. Most importantly, they can use the computer directly, *on their own.*

From data processing's viewpoint, this rapid growth could be appalling, unless some thoughtful planning has been done beforehand.

In this report we discuss the capabilities of the new data management systems, systems that these end users are most likely to work with. And we point out what data processing management can do right now to respond to this trend. Next month we look at how some companies are developing end user programming support groups, and how this trend may affect data processing organizations in the future.

What is end user programming?

One can take a rather broad view of the term 'end user programming.' In its simplest form it involves a straight-forward query to a file, such as "Find record X". The point is, this query statement need not have been previously programmed by a programmer; it is an ad hoc request specified directly to the computer by the user.

At the other extreme, end user programming is akin to conventional programming—but programming that uses a non-procedural language. Such programs contain both operational and control statements, and the application logic may be quite complex.

In between these two extremes users can (1) create and update files, (2) find all records that meet certain criteria, (3) use statistical routines to analyze these records, (4) define input forms and input validation rules, and (5) generate graphs and reports. In total, end users can obtain much of the ad hoc information they need by doing their own programming.

And who are these end user 'programmers' likely to be? They are potentially all employees who need ad hoc information—vice presidents, financial analysts, secretaries, librarians, and so on. They all could conceivably interact directly with a computer in some manner, and not just by following procedures programmed by someone else.

Most of these end users cannot be expected to remember complex syntax or procedures in order to write programs. However, a few could be called 'amateur programmers.' These latter often use mathematical models in their work, so programming in APL, for example, does not scare them. But most end users are not like that. Computers *do* scare them, so the language they use must be very natural, and the system must be very friendly and forgiving. The tool most employees would probably be willing to use is a data management system with a non-procedural language.

You could say the trend toward end user programming began many years ago with the arrival of the file management systems. Both MARK IV and ASI-ST, for example, allowed users to select, sort, and print data stored in tape files; today, they work with data stored on disk and interface with leading DBMS. These products include a report language with which users can enter queries and obtain reports. Typically, users have been programmers, but some end users have used them.

In the mid 1960s, database management came onto the scene. Systems such as TOTAL, IMS, IDS, Codasyl-type DBMS, System 2000, ADABAS, and others have provided powerful ways to access data. With them, users could retrieve information more easily. However, users were required to know about the structure of the files, and they had to enter their queries in pre-specified ways—so these users were programmers, in almost all cases.

The next step in the progression came with data management systems (DMS). These couple a DBMS or file management system with peripheral capabilities—input format design, data dictionary, interactive query, report generation, and so on. Vendors began pulling together the various functions end users would like to perform into one or several linked products.

The latest step has been the addition of non-procedural languages to data management systems. These allow users to (largely) specify *what* is to be done rather than how it is to be performed. These systems contain one set of commands through which users can interact with any part of the system. These DMS support on-line, interactive queries, and some provide additional new functions, such as statistical routines and graphics generators; also most include some type of programming capability. End users can create programs which can be given a name, stored, and run at anytime by just referring to the name. Such programs or dialogs can even ask the user to enter variable information at run time.

Some DMS work in conjunction with popular mainframe DBMS. These DMS can be installed in-house or used through various time sharing services. In addition, there are a growing number of DMS that work on departmental mini-computers; some of these work with DBMS, while others work with file management systems.

Interestingly, we are seeing a growing number of DMS-like systems developing from other types

of products. For instance, a number of program generators—especially those offered with minicomputers—have been enhanced so that they can be used by end users. Even some application development systems, which are tools to aid programmers, are being enhanced so that some end users can use them.

It seems that *many* suppliers are getting into the act—enhancing their products to offer DMS capabilities.

The new data management systems

Following are the major characteristics of the data management systems we found being offered today.

Non-procedural language. The front-end, or the language end users work with, is non-procedural. Such programming is much easier to learn than procedural language programming. These languages consist of commands, each of which performs one function—SORT file X on field Y, SELECT all records with value Y in field Z, or DELETE record A, and so on. Some languages contain control commands as well, such as IF...THEN and DO, so that more complex logic can be specified.

Interactive query facilities permit users to sit at terminals and key in commands to retrieve ad hoc information from files or a database. Because most systems today provide interactive facilities, they can help guide users by prompting them with menus or dialogs. They may also help users out of confusing or 'dangerous' situations—for example, by requiring them to confirm a command to delete a file. Interactive systems are conducive to ad hoc use because they do not impede the user's train of thought. And they allow users to consider more information than was possible manually. Some DMS allow both interactive and batch queries.

Report generation. Whereas query facilities concentrate on helping users retrieve desired ad hoc information, report generators help users format information in a suitable output form. Some systems lump these two categories together under one name. Through a report generation facility, users can use default values that produce standard report formats. Or they can move columns, create calculated rows and columns, etc. to design their own formats. They can even design reports that are based on variable command (parameter) information. These can be stored under a name and then run over and over again, each time with the user supplying different parameters. Often the report generator can perform simple statistical functions, such as averaging, calculating percentages, finding maximums and minimums, and so on. Some products provide the capability to generate other forms of output, such as mailing labels, for example.

Screen formatter. Although report generators have been around for some time, screen formatters are relatively new. They allow users to interactively design data entry and/or query screen formats. Such a formatter (sometimes called a 'screen painter') can allow a user to (1) design a screen format by simply typing in the various data input field names and the locations where they are desired to appear; these field locations can even be moved, by use of an editor. (2) The user may be able to specify input field validation rules; usually, these are only the basic checks, such as numeric-only, range checks, checks against tables of valid values, and so on. (3) The system may help the user specify just what it's (the system's) response should be to each user entry. (4) The system may alert the user to the fact that some necessary logic has not been specified. And (5) the formatter might help the user to design menus and dialogs for the application.

Graphics. More and more vendors are adding graphics capabilities to their DMS. They provide standard routines for creating bar charts, histograms, connected point plots, and scatter diagrams. These can often be drawn on CRT terminals, high resolution graphics display terminals, character and dot matrix printers, and plotters. Some even allow limited color graphics.

Supplementary tools. Since users will use these DMS to replace manual office tasks, we are seeing diversified packages linked to these systems. For example, one system has a module for creating computer letters.

Several DMS include statistical packages with which users can calculate time series, averages, standard deviations, correlation coefficients, and

so on. One product includes a financial analysis package, with which users can develop budgets, profit and loss statements, and so on.

The number of supplementary tools that vendors are adding to their DMS is growing, because they increase product versatility.

Library of programs. It is very common for users to develop a sequence of commands that they will want to repeat in the future. Many DMS allow users to name these programs, and store them in a catalog of programs. Others depend upon the host's operating system to perform this storage function. Once stored, these procedures can be used by any user or by only certain users, if protected by passwords. This storage and retrieval of user-written programs enhances the usefulness of the DMS immeasurably.

Programming interface. In most cases the non-procedural language contains its own control commands, for iterations and conditions. With these iterative and conditional statements, professional programmers (and some end users) can write very complex program logic using the non-procedural language alone. In addition, many DMS provide a link to a procedural language, such as BASIC, COBOL, Assembler, or PL/1—so programmers can access the DMS files with a conventional programming language.

One system we came across allows users to define their own macro commands. One possible use for this is to define a command (say, GOFETCH) that brings in a named subroutine that has been written in any procedural language that the company computer supports. Programmers can thereby intermix non-procedural and procedural modules to create full-blown application programs. Further, they only need to code those portions which cannot be performed by the non-procedural language.

Backup and recovery. A full-blown DBMS usually provides automatic storage backup and recovery procedures; the DBMS used in a DMS may or may not have these facilities. At the very least, though, the DMS should provide commands whereby users can dump selected files to tape or other forms of backup storage, as well as read data from these peripherals. With this feature, users can provide their own backup copies, and can perform recovery. A DMS should also provide some means for restoring information that has been unintentionally deleted, by, say, allowing users to reload a recent generation of a file.

Data dictionary. Most of the systems we have seen provide an austere data dictionary function, for defining data items. We see data dictionaries becoming increasingly important for controlling data definitions, not only for large transaction-type applications but also for departmental applications that use operational data. If a DMS does not have a data dictionary, it is much harder to uncover the data definitions that are used in the programs. Even DMS with file management systems should provide a dictionary. Data integrity provisions should also be provided within the data dictionary; for instance, limits and other validation checks can be specified by users as a part of the data definitions. And the updating of data definitions ideally should be protected by passwords, so that only authorized employees can change them.

Security and privacy. Some systems that we have seen require users to use passwords, to log on and to protect their files. With some, users can designate who can see and use their files. Files may even be placed under the control of a database administrator, and users can be assigned read and write capabilities. At least one system provides file encryption capabilities. Some users we talked with were quite lax about using such capabilities. Many felt their files were not sensitive enough to worry about file passwords. However, a few users who had developed sensitive files, such as pay scale files, did use the security features.

We found data processing management to be concerned about security, and they often have had their own security features created, which they added to the DMS. One such feature requires users to change their passwords regularly, or else they cannot log onto the system.

Links to other DBMS. All DMS have file management capabilities. That is, users can create new files with the DMS and can maintain existing files that use the DMS-acceptable formats. In addition, if you have a DBMS installed already, some DMS may work with your existing files,

without converting them to the DMS structure. Most of the mainframe products provide optional interfaces to such DBMS as IMS, TOTAL, IDMS, ADABAS, and others.

Records maintenance. Maintenance of files is achieved through the use of the non-procedural language. The language contains commands for opening up new files, adding, deleting, or changing records, and so on. Update transactions may be entered interactively at a terminal or, in some systems, stored on disk or tape for batch updating—for those user companies that restrict which files users can update interactively. Or for efficiency, some companies that do not require up-to-the-minute information perform the actual database updating with night-time batch runs. DMS should provide the important supplementary functions for file maintenance as well, such as transaction validation, logging, and so on.

Portability. A future consideration with DMS is portability: Can the product be used on different hardware? Some now can, others cannot. In a distributed environment, or with a change of mainframes, this portability question could become important.

Control executive. One main feature of data management systems is that they not only provide users with the above list of capabilities, they also act as an application controller when DMS programs are run. However, some products with DMS-like features but without this capability, such as program generators, create source code which must be compiled before it can be run.

Some available systems. As an example of the available products, here are some DMS which are available on mainframe computers, through time-sharing services, and on some mini-computers.

FOCUS, from Information Builders, runs on IBM 370s, under VM/CMS, TSO, or CICS. It can interface with IMS, IDMS, and TOTAL files, and it is also available on the Tymnet network.

RAMIS II, from Mathematica Products Group, runs on IBM equipment under OS, VS, TSO, VM/CMS, and other IBM operating systems. It works with IMS, ADABAS, and TOTAL files. RAMIS II is available on National CSS, Litton Mellonics, Sun Oil, and other time-sharing services.

NOMAD, from National CSS, is available over the company's time-sharing network. It also runs on the company's 3200 series computers.

For smaller machines, in the mini-computer range, Four-Phase, Hewlett-Packard, Microdata, Prime, Wang, and others offer DMS products for their systems. And numerous software vendors now have such products. USER-11, from North County Computer Services, works on DEC equipment from the PDP 11/34 to the 11/70, under the RSTS operating system. INFO, from Henco, Inc., works on Prime, Honeywell Level 6, Amdahl, and IBM 370, 303X, and 4300 series computers. CREATE, from Complete Computer Systems, works on Data General equipment.

For addresses of the above suppliers, and for a free listing of DMS products, see References 1 and 2.

Implications for data processing

The emerging trend of end user programming presents some new responsibilities for data processing. Because of this, next month we will discuss some proposed approaches for dealing with these challenges. Here we look at the more immediate implications that we uncovered in our research.

The current implications for data processing involve supporting end user programming—with education and assistance, software products, machine resources, and management.

Education and assistance. End users will need basic education to use these new tools. And they will also need on-going assistance. This support is best supplied by technically competent employees who like to work with people—who become in-house consultants for end users. Most data processing departments do not have many of these types of people, since most programmers prefer to work alone solving technical problems. Therefore, companies will need to search out these people. A few companies have found them in user departments. When end user programming is introduced in departments, some employees become more interested in this new challenge than in their former work, and they ask to be transferred to the support group.

Finding a few in-house consultants who like to work with end users is one immediate implication of end user programming.

Software products. Supporting end user programming requires the right software. The most likely products that we have found are the DMS. In addition, someone (possibly the data processing department) may need to create an end user interface through which employees access the end user system. The DMS may be only one part of an office automation system, for example; other parts could include computer messages, calendars, word processing, etc. The interface would tie all of the parts together, perhaps through a menu of the options offered on the system. The interface might also contain an on-line tutorial, a help facility, and some security features.

Machine resources. From what we have seen, end user programming definitely will increase the amount of machine resources that a company uses. The reason is that the DMS makes it possible for end users and programmers alike to write applications more quickly. So more applications will be running and more resources will be consumed.

Currently there are three sources for providing this increased computing power. The first is the company's mainframe computer. Some DMS are designed to use facilities of mainframe operating systems and DBMS.

A second option is to use the DMS via a commercial time-sharing service. This alternative is often chosen when companies want to experiment with end user programming on a trial basis. This option may not be appropriate if you want to interface the DMS with your existing DBMS files.

The third option is departmental computers. There are a growing number of DMS for minicomputers. Some are based on a DBMS, others are based on file management systems. This alternative may perhaps present the most benefits in the long run, such as system backup, installation versatility, and lower cost. Some companies have found these products meet their needs; others prefer the more powerful DMS that run on their mainframes.

In addition, we are beginning to see ads for DMS on micro-computers. So this may be a fourth option in the not-distant future.

Management. End user programming will require managing these new support functions—supervising the in-house consultants, choosing software, estimating resource requirements, marketing the service, and so on. It looks as if a whole new function might be needed, possibly within data processing.

End user programming will probably also have other, perhaps more long term, implications for data processing. One is the need for expanded data communications. If departmental computers are installed, users will eventually want to access not only the corporate computer but also other departmental or regional systems. They will want to move files between systems, send electronic mail, use software offered on other systems, and even access outside services.

Another implication is that end user programming will probably affect the work being done in the data processing department. Some people predict that application development backlogs will shrink; however, backlogs could easily grow, because end users may adopt the attitude, "the more you get, the more you want." Some say that programmers will concentrate on system software work rather than application software work. And some expect application maintenance to decrease. Next month, we will discuss some of these points in more detail.

What can you do today?

To bring this discussion back to the present, we ask: What can you do *today* about this new trend?

Search for suitable DMS. First, we suggest that you find out which of your users are the most anxious for more computer services. We suspect that these will be the most dynamic users who want to put new applications onto the computer. But their budgets probably are not large enough for all of these applications, so they feel frustrated.

Once you have singled out a few of these frustrated users, we suggest you present them with the idea of letting them develop some of their

own applications. Tell them about the new DMS products. Perhaps one of these products can be installed on your in-house computer. If not, then maybe these users would be willing to experiment with a time-sharing service. Or they might even want to install a departmental computer which supports a DMS package.

Determine support needed. The next consideration should be what kind of support these users will need from data processing, in order to get started (and keep going). The data processing departments we have talked with—those that are encouraging end user programming—started out with small support groups. Typically, such a group has a manager, a few in-house consultants, one or two software products, and some machine time. Some used the DMS as is; others wrote front-end modules to provide a help facility and to tie the package to other facilities, such as electronic mail. Then they began educating a few users at a time on the use of the system. If you want to encourage end user programming, these steps will get the ball rolling.

Next month we will look at supporting end user programming in more depth, and will discuss some long term effects that this emerging trend might have on data processing organizations. We believe that data processing should start soon to plan for what users will probably ask for in the future in the way of end user programming.

End user programming, like office automation, is coming. It really cannot be ignored. And like office automation, it will most benefit those companies that guide its growth. By putting new management policies into place, establishing a small support group, and installing some end user programming tools, data processing can guide and encourage this new trend—rather than ignore it or watch it spread uncontrollably. End user programming is really a new challenge to data processing management.

REFERENCES

1. For a free listing of data management systems that we have come across, write EDP ANALYZER, 925 Anza Avenue, Vista, California 92083.
2. Addresses of the suppliers of products mentioned in this report are: CREATE (Complete Computer Systems, 159 Gibralter Road, Horsham, Pennsylvania 19044); ENGLISH (Microdata Corp., 17481 Red Hill Avenue, Irvine, California 92705); FOCUS (Information Builders, Inc., 1250 Broadway, New York, N.Y. 10001); INFO (Henco, Inc., 35 Walnut Street, Wellesley, Massachusetts 02181); INFORMATION (Prime Computer, Inc., 145 Pennsylvania Avenue, Framingham, Massachusetts 02146); NOMAD (National CSS, Inc., 644 Danbury Road, Wilton, Connecticut 06897); RAMIS II (Mathematica Products Group, P.O. Box 2392, Princeton, New Jersey 08540); USER-11 (North County Computer Services, 2335 Meyers Avenue, Escondido, California 92025); VISION (Four-Phase Systems, Inc., 10700 North De Anza Boulevard, Cupertino, California 95014).

Prepared by:
Barbara C. McNurlin
Associate Editor

SUBJECTS COVERED BY EDP ANALYZER IN PRIOR YEARS

1978 (Volume 16)

Number

1. Installing a Data Dictionary
2. Progress in Software Engineering: Part 1
3. Progress in Software Engineering: Part 2
4. The Debate on Trans-border Data Flows
5. Planning for DBMS Conversions
6. "Personal" Computers in Business
7. Planning to Use Public Packet Networks
8. The Challenges of Distributed Systems
9. The Automated Office: Part 1
10. The Automated Office: Part 2
11. Get Ready for Major Changes
12. Data Encryption: Is It for You?

1979 (Volume 17)

Number

1. The Analysis of User Needs
2. The Production of Better Software
3. Program Design Techniques
4. How to Prepare for the Coming Changes
5. Computer Support for Managers
6. What Information Do Managers Need?
7. The Security of Managers' Information
8. Tools for Building an EIS
9. How to Use Advanced Technology
10. Programming Work-Stations
11. Stand-alone Programming Work-Stations
12. Progress Toward System Integrity

1980 (Volume 18)

Number

1. Managing the Computer Workload
2. How Companies are Preparing for Change
3. Introducing Advanced Technology
4. Risk Assessment for Distributed Systems
5. An Update on Corporate EFT
6. In Your Future: Local Computer Networks
7. Quantitative Methods for Capacity Planning
8. Finding Qualified EDP Personnel
9. Various Paths to Electronic Mail
10. Tools for Building Distributed Systems
11. Educating Executives on New Technology
12. Get Ready for Managerial Work-stations

1981 (Volume 19)

Number

1. The Coming Impact of New Technology
2. Energy Management Systems
3. DBMS for Mini-computers
4. The Challenge of "Increased Productivity"
5. "Programming" by End Users

(List of subjects prior to 1978 sent upon request)

PRICE SCHEDULE (all prices in U.S. dollars)

	U.S., Canada, Mexico (surface delivery)	Other countries (via air mail)
Subscriptions (see notes 1,2,4,5)		
1 year	$48	$60
2 years	88	112
3 years	120	156
Back issues (see notes 1,2,3,5,)		
First copy	$6	$7
Additional copies	5	6

NOTES

1. Reduced prices are in effect for multiple copy subscriptions and for larger quantities of a back issue. Write for details.
2. Subscription agency orders are limited to single copy subscriptions for one-, two-, and three-years only.
3. Because of the continuing demand for back issues, all previous reports are available. All back issues, at above prices, are sent air mail.
4. Optional air mail delivery is available for Canada and Mexico.
5. We strongly recommend AIR MAIL delivery to "other countries" of the world, and have included the added cost in these prices.

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_____

THE IMPORTANCE OF DATA MANAGEMENT SYSTEMS

Recently we went through issues of EDP ANALYZER for the past two years. We looked for important new uses of computers that we had written up—uses that either involved a DMS or clearly could be implemented more quickly if a DMS had been used. The list impressed us. In addition to end user programming, discussed in this issue, and the role that DMS can play in job re-design, which we discussed last month, here are some of the other uses.

Distributed systems. We suspect that departmental computers and personal computers will be installed in ever-increasing numbers. They will be able to draw on services from, and exchange data with, central data processing as the need arises. A DMS can help each such computer better serve its users, as we discussed two months ago.

Office automation. The boundary between office automation systems and the kind of distributed system just mentioned will become very fuzzy indeed. You will begin to see DMS offered along with office automation systems and as part of managerial work-stations, which we discussed last December.

System development. You will be hearing more and more about the prototyping approach to information system development—where DMS has an important role to play. We'll discuss this in our September issue.

Computer support for managers. We have had a number of recent issues dealing with this subject—and a DMS certainly fits this type of computer use. And a DMS should make decision support systems much easier to implement and operate.

Programmer work-benches. We have been observing at close hand how useful a DMS is for professional programmers. With one available, they can concentrate on the special requirements of an application; all of the routine functions are handled by the DMS.

Yes, keep your eye on the DMS area. It can play a big role in your future computer use.

Editorial: Richard G. Canning, Editor and Publisher; Barbara McNurlin, Associate Editor. While the contents of this report are based on the best information available to us, we cannot guarantee them.

Missing Issues: Please report the non-receipt of an issue within one month of normal receiving date; missing issues requested after this time will be supplied at the regular back-issue price.

Microform: EDP ANALYZER is available in microform, from University Microfilms International, Dept. P.R., (1) 300 North Zeeb Road, Ann Arbor, Mich. 48106, or (2) 30-32 Mortimer Street, London W1N 7RA, U.K.

Prices: Prices of subscriptions and back issues on page 13.