EDP ANALYZER

© 1977 by Canning Publications, Inc.

JANUARY, 1977
VOL. 15, NO. 1

THE ARRIVAL OF COMMON SYSTEMS

"Do your own thing" has been the slogan of many of today's young people—and it might also apply to many of today's system analysts, programmers, and even DP managers. The proliferation of application systems that comes from "doing your own thing" is essential in a rapidly developing field such as the computer field; it is at the heart of the trial-and-error learning period. But the time arrives when proliferation must be controlled, to reduce duplication of development costs, high maintenance costs, and hard-to-compare system outputs. Common application systems offer one means of controlling proliferation. These are generalized, flexible systems that are designed to meet the needs of a number of users. There have been some unhappy experiences among the pioneers of common systems, such as when those systems turned out to be too inflexible and were removed by the users. But now some notable successes have occurred. If your organization has multiple computer sites—or even a single computer site, for that matter—here is a development you should consider.

For years, data processing management has been searching for the elusive "common application system"—almost, it seems, as Ponce de Leon searched for the Fountain of Youth. Yes, there have been a few successes such as the generalized tax routines that are being used in payroll systems. Turnkey systems using mini-computers are widely used, as are some service bureau application packages. These relatively few successes have kept the hopes alive for high quality generalized application systems.

But common application systems have encountered their share of difficulties. Originally designed to meet the needs of one or a few user organizations, they have been found to be too inflexible when other organizations tried to use them. Trying to modify this type of "common system" so as to make it meet the needs of a particular user often proved to be even more difficult than building a system for that user from scratch.

Common application systems are (or should be) of interest to most data processing installations. For the single site user, it means being able to purchase suitable application packages—resulting in faster implementation, reduced development costs, reduced maintenance costs, and so on. For multi-site organizations, these same advantages accrue as well as others such as consistency of reports from the several sites.

Common application systems are of particular interest to multi-national organizations. The problems of managing computer sites in multiple countries are similar to the problems of managing multiple sites within one country, except that the problems are more extreme. If common systems can be developed and run effectively in a multinational environment, then they should hold promise for both single site and domestic multiple site users as well.

The good news is that some multi-national organizations have developed and are using common application systems successfully. We are not saying that all of the problems associated with common systems have been solved; there are still some important constraints on where they can be used successfully and failures are still occurring. But there are now enough successes, we believe, to indicate that common application systems are becoming feasible.

This is the first of several reports on experiences with common application systems. Our next report on this subject will appear four months from now. In our research on this subject, we found several different approaches to the use of common systems. These might be classified as follows.

In-house development. Common application systems may be developed in-house, to be operated in two different environments. In one environment, the systems are developed to run on small to medium size computers that are located at the company's remote sites. In the other environment, the systems run on a central computer and serve the (rather different) business requirements of the remote sites via data communications and remote terminals.

Purchased software/services. The other main approach is to purchase common application software or services from outside suppliers. Within this general approach, several variations have been encountered. The software can operate on small to medium size computers located at the remote sites. But there is still one lower level within this approach—truly generalized application software (such as IBM's System 32 Industrial Application Packages) versus skeleton generalized systems that must be customized for each particular user. Another approach is to use application software available from service bureaus. And still another approach is to obtain common application services from a time-sharing service.

In this report, we will deal with user experiences with in-house development. Our next report, four months hence, will describe user experiences with purchased software/services.

We begin our discussion of the in-house development with The Coca-Cola Company's experience using multiple small computers.

The Coca-Cola Company

The Coca-Cola Company, Inc., with headquarters in Atlanta, Georgia, is a leading manufacturer and marketer of soft drink beverages and other food materials. The company operates in more than 130 countries of the world. Annual corporate sales are in the order of $2.9 billion and the company employs over 29,000 people. In general, the company franchises and sells syrup to local bottling companies in the various countries; the company typically does not own bottling plants.

Interest in common applications systems began at The Coca-Cola Co. in the late 1960s. In 1967, the director of the management information system division for Europe initiated a project to develop application systems for a bottler in The Netherlands. These systems were developed to meet the needs of this particular bottler and were not generally applicable to other bottlers. But from this experience, the director saw that such application systems probably could be developed for bottlers in general.

At about the same time, there was a growing interest among bottlers in Belgium and Italy for common application systems. IBM helped The Coca-Cola Co. develop the "European Standard Package" to operate on an IBM 360/20 card system. This package went into operation in late 1968 and was used by all bottlers in Belgium and partially by some in Italy.

Also, at about this same time, bottlers in West Germany were planning a centralized system for serving all bottlers in the country, via data communications. But with the success of the above-mentioned package, it was decided to investigate a small computer (System 3) approach for West Germany.

Because of all of this interest, a task group was established in late 1971 to study and design an international common application package for European bottling plants. The main part of the task group was located in Vienna, Austria, but in addition implementation teams were set up in several countries. These teams were the contact points between the central task group and the bottlers in the various countries.

The common system that has emerged has been called BASIS. It was first installed in Cologne, West Germany, in 1973. In 1974, 16 more locations were added, and in 1975, 17 more. By the middle of this year, The Coca-Cola Co. expects to have BASIS installed at 72 locations in 12 countries—in Western Europe, Africa, and Australia. It is now being considered by bottlers in many other countries.

There is some joint use of BASIS by bottlers. The maximum joint use currently is four bottlers, limited more by the need to physically transport documents than by system capability. Basis has been designed for use by medium and larger size bottling operations which serve at least 2000 "outlets" (retail establishments where Coca-Cola is sold).

Basis is designed to operate on the IBM System 3, models 8, 10, 12 and 15, with a minimum of 32 kbytes of memory, 2.4 mbytes of disk capacity, a printer, and card or diskette input-output. Basis is now being adapted for use on a System 32, for bottlers with up to 5000 outlets. To achieve efficient usage of both main memory and disk storage, BASIS has been programmed in extended RPG II; the extensions have been accomplished by adding functions programmed in assembly language—for compressing records and fields, for using bits within bytes for specific purposes, etc.

The System 3 is not well suited for serving a number of outlying sites via data communications, but this is not a significant limitation, the company feels. Since the bottlers are relatively independent of each other and each serves a specific geographic area, there is no need for centralized data files or the central control of the business. Hence, stand-alone System 3s or System 32s are satisfactory. However, The Coca-Cola Co. has provided a means for remote entry and output using the IBM 3741 model 3 terminal, which they call Tele-BASIS. Each week the System 3 prepares diskettes with master data for the outlying sites and these are physically transported to the sites—say, 50 miles away or so. During the week the 3741s are used to prepare individual invoices, route settlements, stock reports, and so on. Transaction data is captured on diskettes, which are sent in to the System 3 at the end of the week, where the remainder of the processing is done.

Basis covers four groups of applications: (1) general administration, (2) sales and marketing, (3) production and warehouses, and (4) finance. By mid-1976, 27 specific applications within these four areas had been developed, involving some 300 computer programs. Of these 27 applications, 20 deal with routine data processing and the remainder are designed to aid in management decision making.

Basis was developed using generalized design. The various implementation teams throughout Western Europe determined system requirements from interviews with the various bottlers. The task group in Vienna assembled these requirements and then—*ignoring the names that the various bottlers used for the categories of activities*—the team determined just what was being done in each activity. The team sought to find the structure of each activity—what was constant from bottler to bottler and what varied among the bottlers.

To illustrate by a simple example, different tax situations (such as value added tax and others) could be reduced to a 9 x 3 table of tax rates. The tax rate was determined to be a function of the product and the outlet. Analysis showed that there were never more than three outlet classes nor more than nine product categories. Hence the simple 9 x 3 table would cover all the tax situations. The product category code is stored in the product master file and the outlet class code is stored in the outlet master file. The actual meaning of these codes can vary from country to country. The actual local meanings can be printed on reports by means of inserted constants.

The Coca-Cola Co. has purchased some components of BASIS from outside suppliers, such as the generalized general ledger package obtained from Software International.

Installing BASIS

How easy is it to install BASIS? How well do the common application systems meet local needs? To find out, we visited Jarlsberg Mineralvann A/S, in Oslo, Norway, the largest Coca-Cola bottler in that country. Like many Coca-Cola bottlers, Jarlsberg also handles other soft drink beverages. The company has about 5000 soft drink outlets in the Oslo area. In addition, Jarlsberg owns a brewery in southern Norway, distributes beers, and owns another soft drink bottling plant some distance from Oslo.

Jarlsberg began considering the use of BASIS in mid-1973 and signed a contract for it in January 1974. At that time, no one in the Jarlsberg company had any experience with data processing. An EDP manager was selected from the staff and sent to Vienna for five months. He took with him sufficient data for building a mini-bottler system in Vienna. At the same time, another person was selected to be the computer operator. He had been in the sales department and was familiar with the customer records. He began collecting the data for the master files. Finally, the route sales people were trained in the new procedures, forms, and so on.

An IBM System 3 model 10 was installed in November 1974 and cutover to the new system occurred in January 1975. One person from Vienna came to Oslo to help with the conversion. The various options built into BASIS meet their needs except for payroll, the people at Jarlsberg told us; payroll is not included within BASIS.

The computer processing has given Jarlsberg faster, better control of its business operations. Figures are more accurate. Changes are implemented much more quickly and effectively, such as price changes, tax changes, and production changes. There has been no reduction in direct costs but the business has increased by about 10% without the need to add office staff. Based on the success with BASIS, Jarlsberg is considering how it can best provide the same services for its brewery and its other bottling plant.

The Coca-Cola Co. is pleased with the acceptance of BASIS by its franchised bottlers. The company is continuing to enhance the system. In addition, an executive position has been set up at corporate headquarters for participating in the overall BASIS program—such as the other locations in the world where it might be used, what new features should be added, and when serious revisions should be undertaken.

We believe that BASIS demonstrates that a common application system can be developed which will operate in a wide variety of business environments.

Eastman Kodak Company

The Eastman Kodak Company, with headquarters in Rochester, New York, is a leading manufacturer and marketer of photographic film, cameras, photographic services, and chemicals. Annual sales exceed $4.5 billion and the company employs some 124,000 people.

Kodak operates in 42 countries of the world. The company recognized that the Kodak units in these countries had similar but not identical problems. For instance, computers could help these companies with their billing and inventory control activities. At the same time, not many of the companies had experience in designing and building such systems. Also, substantial duplication of effort would occur if each company did try to design and build its own systems.

So the first effort toward international common systems occurred in Scandinavia in 1968. An international computer center was set up in Gothenberg, Sweden, to serve the Kodak companies in Sweden, Norway, Denmark, Finland, Belgium, and The Netherlands. It was called the Northern Information Center (NIC). It is still in operation and we will give more details about it shortly.

Then in 1970, a common system was developed by a project team in the Far East, composed of EDP supervisors from the Philippines, Hong Kong, Singapore, and Thailand. The system was installed in those same companies.

Using the experience gained with NIC and the Far East systems, the International Common Computer System (ICOM) was developed in 1970-71. ICOM was first installed in Puerto Rico in 1971. It is now installed at 20 locations around the world, at most of the computerized Kodak companies.

What explains Kodak's success with common systems when so many other organizations have encountered difficulties with them? The success may well have been due to the approach taken at the outset by the NIC project.

Northern Information Center

The idea of an international computer center, using common application systems, was raised in 1967 in Scandinavia. So a team was formed of the EDP supervisors from each of six countries—Sweden, Norway, Denmark, Finland, Belgium, and The Netherlands. This was a real team effort for developing the system specifications, we were told. The team members had good esprit de corps, knew how data processing was serving the business functions in their companies, and had direct access to company management.

The policies under which the project was conducted are important. First of all, the users were heavily involved. When issues arose, the rule was to perform a detailed, objective analysis to clearly define the issue, and then propose a course of action. By getting all of the facts before everyone, issues generally were settled easily. The project team sought to determine the real needs of the companies, not the apparent ones. For instance, it was found that different rules were in effect in the different companies, reflecting, for example, the varying characteristics of customers who were buying X-ray film. But the analysis of what was actually occurring in each of the companies on this point showed that common rules could be used.

So the application systems were designed with the needs of all six companies in mind. When country-to-country variations were needed, they were built in to the systems. But frequently it was found that the variations were illusory and that common procedures could in fact be used.

A common computer center was set up, at Gothenberg, Sweden. The application systems were first installed in Kodak Finland in 1968 and were installed in the other five countries as rapidly as possible thereafter. Transaction data is transmitted to Gothenberg each afternoon, with invoices and other output information transmitted back the next morning.

The differences in languages among the six countries has been no problem, we were told, either during the development of NIC or in its operation. The common language is English for system documentation.

As mentioned above, the almost immediate success of NIC led to the development of the Far East system in 1970, and then to ICOM in 1970-71.

International Common Computer System

ICOM was developed to facilitate the corporate management reporting by non-U.S. units of Kodak. It includes 19 application systems such as sales analysis, inventory control and marketing information analysis. It does not include payroll. ICOM has almost 600 computer programs in these 19 systems.

ICOM has been aimed at Kodak units with annual sales in the $5-100 million range. It was designed to operate on small to medium size equipment, specifically on the IBM System 3, System 32, or the 370 DOS configurations. It can operate either via an international data center serving several countries or on a single computer within one Kodak company.

In developing ICOM, the designs of NIC and the Far East systems served as the starting point. In addition, the needs of Kodak companies in other parts of the world were studied in depth. To accommodate the real variations in needs (as differentiated from apparent variations), the system design included parameterized programs and optional routines in modular form. The 600 programs involve over 120,000 lines of code. Using these generalized features, a user company can adapt ICOM to meet its specific needs.

In fact, we were told, so comprehensive was the coverage of needs that ICOM code has met almost all of the local needs in the 20 installations to date. In 10 of the 19 application systems, all code is common and no customizing has been needed. In another 7 of the systems, over 95% of the code is common. The merchandise invoicing application has needed the most local customizing, with about 65% of the code being common.

Kodak companies have found ICOM to be so comprehensive that when they install a new procedure (perhaps in support of a new product or a new marketing method), chances are the procedure is already in ICOM.

Pros and cons

Common systems are not an unmixed blessing, Kodak has found. Users tend to resist common systems, we were told, believing that such systems cannot meet their needs. The design of common systems is more expensive than designing a similar system for one company. One reason for the additional expense is the need to determine when a customer request is really a "must" and when it is a "nice to have" feature. The costs of interfacing common systems with the other existing local systems can be high; the smaller the percentage of the total work-load done by the common systems, the higher this interfacing cost will be. The generality of the common system means that users pay an overhead price for the options they have chosen not to use.

But the net result of the common systems has been very much on the positive side, Kodak feels. The biggest advantage to the Eastman Kodak Company is that standard data elements and formats give greater compatibility of reporting to corporate management. Common systems have tended to be more flexible than locally designed systems, making it easier for the Kodak companies to add new products and services. Common systems tend to be better documented and tend to use later technology than do local systems.

The Coca-Cola Company's BASIS and Kodak's ICOM represent the multi-site approach to common systems. Let us now consider a case where the common systems are being run from a central computer facility.

Shell Scandinavia

The oil refining and marketing operations of Shell International in Scandinavia are conducted by four subsidiaries—the Shell companies in Sweden, Norway, Denmark, and Finland. Svenska (Swedish) Shell has the largest annual sales of the four, amounting to some SKr 2.1 billion (about $500 million).

In late 1965, these four companies established the Nordic Data Centre in Stockholm. The goal of the NDC was to take over all computing activities for the four companies and to develop standard application systems. But the effort ended in failure and the center was disbanded in 1968.

Why the failure? The people at Svenska Shell see several possible reasons. The project was initiated more from central office management than from interest by the four companies; the companies had not yet reached the point in their use of computers where they were thinking about common systems. The inter-company steering group for the project was not well developed, so that most of the decisions ended up being made by Svenska Shell. The resulting systems were almost completely "standardized," requiring the same input and delivering the same output for each of the companies. No options for meeting local needs were provided. The project occurred in the relatively early days of computers, when the computer staffs were finding it hard to communicate with users within their own companies, much less between companies. The computer technology was not well developed for supporting common systems. Finally, there was no overall agreement among the users as to what the application systems should do.

But even with the failure of the NDC, the concept of common systems did not die. In 1971, discussions on the subject started again among the Swedish, Norwegian, and Danish companies. This time, though, the key word for the project was to be "harmonization" to indicate that the companies would have to agree on every step if a successful effort were to occur.

First step. Between 1971 and 1974, the IBM 360/20s in Norway and Denmark were replaced with remote entry terminals connected to the center in Stockholm. During this same period, the 360/50 in Stockholm was replaced with a 370/145. These steps provided the Norwegian and Danish companies with much more computing power than they otherwise would have had. Yearly time allocations and joint scheduling was used to avoid peak loads on the computer. Harmonization of standards, techniques, programming languages, and the use of software occurred. Close contacts were established among the computer staffs in the three countries, and joint training efforts were set up.

Next step. After the central processing proved successful, the general managers and the finance managers of the three companies met in late 1973 to consider further cooperative steps. The Nordic computer steering committee was organized to guide the overall implementation of policies set up by general management. This committee consisted of the finance managers of the three companies and the head of the computer center in Stockholm (who was designated as the Scandinavian computer manager). It was decided to develop common application systems jointly among the three companies, but on a different basis from before. A generalized design methodology would be used, so as to leave as much freedom as possible to the individual companies for operating the systems. (This generalization proved to be a challenge to the staffs, resulting in good motivation for the project.) It was also decided to provide all high quality computer expertise from Stockholm, while at the same exchanging qualified computer and finance personnel among the three companies. Also it was decided that where no plans existed for a common system, the three companies would be free to develop their own systems. In practice, this has meant that big systems have been developed as common systems while many smaller systems have been developed locally.

The first application system undertaken as a common system was accounting and cost reporting, done during 1974-75. This system proved to be successful, meeting the needs of the companies and costing only 15% to 20% over the cost of a system for Sweden alone.

The pattern of system design that has emerged is as follows. The requirements of each company, for a given application, are documented and then combined for all of the companies. Where variations are found to exist, they are explored. If agreement can be reached on a common routine, it is used for all companies. Where variations are required, they are provided by optional routines. We were shown a chart that looks very much like a program module chart. Each module was marked as to whether it was a common module or whether it was specific to one or two countries. The input in general varies by country, the updating routines are common, and some of the output is common. For the remainder of the output, each country extracts what it wants from the common data files using parameter cards to specify the content and format of the requirements. Extensive use is made of external tables, which are grouped together in a stand-alone system, for steering the system and for overcoming incompatibilities between different countries' data.

Further progress. The general managers and the finance managers of the companies met in the fall of 1975 to review progress and plan further steps. It was decided to incorporate Finland into the program. The needs and possibilities of a computer network were to be studied by the four companies. A joint computer staff planning and job classification system was to be initiated, and a standard project management system installed. Also, target dates were discussed for converting all major existing systems into common systems.

As of mid-1976, seven common systems had been implemented. Two were operating in all four companies, three in three companies, and two in just two companies. Four more common systems were under development, all involving joint efforts of the companies.

The Scandinavian Shell companies have found that it is practical to develop a system jointly at several different locations. To accomplish this, they use the MARK IV system and DL/1 data base management, as well as a common project management and development process obtained from IBM.

Project organization. Much of the success of the common systems, we were told, can be attributed to the organization of the efforts. Overall direction is provided by the steering group, consisting of the four finance managers plus the Scandinavian computer manager. This group suggests ideas for investigation, selects projects, oversees the performance of the computer center, and appoints the next lower level group. This steering group meets three times a year. The next lower level is made up of several decision groups, each consisting of the appropriate department heads from the four companies plus the head of systems in Stockholm. These groups make design decisions for the respective companies, monitor project progress, then appoint the next lower level groups. At the third level are the project groups, each consisting of four main user representatives (from the four companies) and one system designer who is qualified to work on a Scandinavia-wide basis. One company acts as the coordinating user for the project, making sure that requirements are defined and that specifications and design are approved. The terms of reference for the group must be carefully defined. The project leader is a user selected from the country in which most of the development work will occur; the success of the project is very dependent upon the caliber of this project leader. Finally, at the fourth level are the programming groups, which construct the system.

The Scandinavian Shell companies have found something interesting from this organization. The business-oriented system analysts who know little or no computer programming are playing a smaller and smaller role in the projects. In the past, the system analysts have been the key figures in application system development—the intermediaries between users and technical staffs. Now with the users working directly with technically-oriented system designers, the need for the system analysts is disappearing.

Another point of interest is that these Shell companies have found it much easier to install a common system in applications that have already been mechanized than in those being done manually.

Problems. Common systems are not without their difficulties, we were told. The main one at the Scandinavian Shell companies relates to timing. It is seldom the case that each of the four companies wants the same common system at the same time. To resolve the issue, the policy has been adopted that if two companies agree on the timing for a common system, it will be undertaken on the time scale they desire. Another problem, mentioned earlier in this report, involves the need to interface the common systems with the other existing systems at each site. Senior management people and senior staff members must spend more time on Scandinavian problems, leaving less time for them to work on their own company problems. A smaller company may be pushed into a solution that it does not really want or need. Local initiative can be killed. And the inter-company project management system is costly and can become bureaucratic. However, as long as these are recognized as problems, they can be kept in check, we were told.

Benefits. The Scandinavian Shell companies find that the development costs for common systems are not much higher than the cost of development for one company. System quality is higher with common systems and maintenance costs are lower. It is easier to make enhancements to common systems, due to the generalized design. And by the pooling of expertise, it is easier to keep pace with technical advances in the field.

This second attempt at common systems by the Scandinavian Shell companies has been a successful one.

Are common systems feasible for you?

A seminar on multi-national information systems was held in conjunction with the 1976 National Computer Conference in New York City, under the auspices of AFIPS (the sponsor of the NCC) and IAG (the IFIP Applied Information Processing Group). The seminar was chaired by Paul Strassmann, Director of Administration and Information Services for the Xerox Corporation. It was attended by information service executives from the U.S., Canada, France, and the U.K.

The main subject of discussion at this seminar was the possible factors that inhibit the sharing of application software across national boundaries. Is there anything unique about multi-national information processing that inhibits this sharing, the attendees were asked. Strassmann issued a report summarizing the consensus of the attendees.

Differences in laws, taxation, and currency are probably not real obstacles to the sharing of application software and methods, the group concluded. This was the most unexpected result of the seminar, Strassmann observed; very few instances could be cited by the group on how these factors had inhibited sharing.

Vendor support for multi-site systems was not an obstacle, nor was the question of technology sharing among the sites, according to this group. It might be necessary to make special efforts to provide capable staff members at specific locations, but the same problem can arise at domestic sites.

But are not the multi-national common systems more complex than domestic common systems, the group was asked. The consensus view was that the problems in these two situations were very similar. The problems that were brought up were found not to be unique to the information systems function nor to multi-national operations.

Well, then, how about data communications? Isn't that much more of a problem area in multi-national information systems? "No," concluded the group. Multi-national data communications problems are mainly matters of cost and planning lead time, rather than an inherent obstacle. There are enough success stories to indicate that these problems can be overcome.

But the group did identify two problem areas that make difficult the sharing of application software across multiple locations. One is *internal resistance to central control.* For successful sharing, the parent organization must have previously established long term, coordinated management goals and controls. Information systems should follow the sharing of management know-how, not lead it. The other problem area is *standards.* A major inhibiting factor to the sharing of application software and methods is the relative weakness of industrial standards and protocols, the group felt. Organizations need strong standards and protocol programs.

Internal resistance and the lack of standards, then, were singled out as the main inhibitors to the sharing of application systems. It appears to us that these might well be the main inhibitors to the use of common software in almost any computer environment, not just on the multi-national scene.

From our discussions with many information processing executives, it appears to us that the "internal resistance" problem is the one that must be overcome first. One of the mechanisms for helping to overcome it is to install a corporate-wide standards program. A standards program, while necessary, is not sufficient for overcoming internal resistance.

Three factors have stood out, in our view, as contributing to this internal resistance problem. These are: the nature of the business, the nature of the motivation, and economic justification.

The nature of the business

As the case examples at the beginning of this report have indicated, the country-to-country variations that exist in a multi-national environment are really not obstacles to the use of common application systems. So one might conclude that variations in the *business environment* are not inhibitors to sharing.

Much more fundamental is the nature of the businesses at the various sites. Two extremes in the nature of the business might be (1) the multi-business conglomerate and (2) the single product business. Closely related to the single product businesses would be the common product businesses—such as Coca-Cola, Kodak, and Shell—where essentially the same products are sold in each of the sites (countries).

With single product and common product companies, the manufacturing and marketing functions are basically the same at each of the sites. The common application systems must adapt to the various local business environments. As we have seen, this has proved to be feasible to do.

With the multi-business conglomerate, the situation is quite different. Even if the company is basically in one type of business—say, manufacturing—the nature of the manufacturing function in the several subsidiaries may be so variable that it may be very difficult to develop a common application system. Some subsidiaries may manufacture on a continuous flow basis, others on a small batch basis, and still others on a large batch basis. The shop control function in large metal fabrication job shops is particularly troublesome; more success has been encountered in common application systems for assembly operations. We will have more to say on this subject in our next report on common systems.

If the subsidiaries of the conglomerate operate in very different lines of business—say, manufacturing, publishing, hotel management, and transportation—then the problem of common application systems becomes even more difficult. It still might be possible to develop some common application systems for the financial function in such subsidiaries, however.

Still another factor, related to the nature of the business, is the size of the business at each site. A site may be so small and so isolated that it is not economically feasible to install either a small computer or a remote terminal there. Minicomputer systems and improved data communications facilities continue to lower the minimum size site that can be served with mechanized data processing.

The nature of the motivation

The question to be answered here is: where does the impetus for common systems come from?

If the push for common systems comes mainly from corporate headquarters, that usually will not be sufficient to assure success—or so we gather from our discussions. If the individual sites already have their own computers and are doing—and want to continue to do—"their own things," then corporate headquarters is going to find it very hard to install common systems with any sort of rapidity.

We talked to one multi-national information systems executive who described three different cases for us that bear on this subject of motivation. In all three companies, the various sites within each company were performing basically the same type of business; none of these was the multi-business conglomerate situation. So in each case, common application systems were a desirable goal, as far as corporate headquarters was concerned.

In one case, the various sites resisted the idea of common systems and wanted to continue to develop their own systems. Instead of trying to force through the idea of common systems, the corporate systems office decided to define a master, or ideal, system for each application area. The inputs, outputs, files, and processing functions were defined so as to provide a capability beyond what any of the sites were then providing. Each site was then asked to compare its existing application systems with these master systems and list the enhancements that it might expect to make in the next several years. These master systems proved to be helpful in getting the various sites to begin thinking along the same lines, as a step toward common systems.

In another case, the various sites resisted any thought of common systems, each maintaining that its needs were "different." In this instance, corporate headquarters instituted a two-fold program. One aspect called for the consolidation of computer sites, with the idea in mind that the use of common routines could be encouraged when computer facilities were shared. The other aspect was the introduction of a strong corporate standards program at all of the sites. As the various sites began to see that common systems might be workable, and as budget restraints forced them to look for more efficient ways of doing things, it was hoped that common systems might be accepted. But corporate management recognized that it might well be five years or so from the start of this effort before the first common systems were installed.

In the third case, the company was setting up brand new marketing units in a number of European countries. The corporate systems department, on a rush basis, designed a common system that could be used in each of these new units. The time factor was critical because the system had to be ready by the time the first unit was organized. Since each new unit began using the common system at the outset, no resistance was encountered. However, two previously existing units continued to use their old systems and resisted the installation of the common system.

The desire to want to "do our own thing" is a strong one. Until it can be overcome, the installation of common systems may not be feasible.

Economic justification

The economic justification for common systems is obviously important but is a difficult subject to analyze in a general fashion. By that we mean that we do not know of any reliable ground rules that will tell how much a common system will cost or how much it will save.

How much does it cost to develop a high quality, generalized application system, as opposed to one designed for one particular user? The answer clearly depends on the amount of generality—how much logical variation the system is capable of handling. If the system is being designed to meet the needs of only three or four sites, then an extra cost factor of perhaps 15% to 20% might be reasonable; only those variations that occur in the three or four sites need to be considered. But if the system is to be installed at tens of sites, then the needs of all of those sites must be considered. The requirements study becomes longer and the number of variations in the programs becomes greater. The overall cost might be at least two or three times what it could cost to develop the system for one site only.

A well designed, generalized system probably will be less costly to maintain and to enhance than is true of the garden variety of application systems. Maintenance costs are less because maintenance is done only once, and updated copies are then supplied to each site. Enhancement costs are less because some of the desired enhancements are designed into the system in the first place.

But a generalized system often will be more costly to operate than a tailor-made system. Program sizes normally will be larger. More condition tests will be made. Processing steps will be taken that meet the need of all users but some of which may not be pertinent for any particular user.

As we say, we do not know of any general rules pertaining to the economic justification of common systems.

If common systems look feasible

Assume, now, that the nature of the business, the nature of the motivation and the economics support the introduction of common systems. What happens next?

As we see it, the main two steps are: set up an organization and policies to support common systems, and use a generalized design approach.

Organization and policies

In most of the successful uses of common systems that we have encountered, users have played strong roles in project leadership and in project participation.

So, if common systems appear feasible for your situation, we cannot urge too strongly that you establish policies and set up the project organization so as to obtain a high degree of user participation.

Further, the user participation should represent all (if possible) sites that will be using the common systems.

The common systems may be developed at one central point, or may be assigned to the several sites for development, or may be developed jointly by two or more sites. The main factor in this decision is the computer expertise that is required and that is available at each of the sites.

The role of the users is to determine the requirements for the common systems, approve the specifications, monitor the progress and the testing of the systems, and to oversee the installation of the systems.

With multiple sites involved, disagreements are sure to arise. Some form of "harmonization" (to use Shell's term) will be needed, and should be established by policy. An effective way of reconciling differences must be found and used. Again, it should be the users who play the key role in reconciling differences.

Computer technologists, of course, should perform the system design and system building. And it is the computer technologists who may have to identify the true variables in the system, as contrasted with the "this is the way we prefer to do it" variables.

You might keep in mind Shell's point about the reduced need for business-oriented system analysts in such a project organization. We have not come across another organization that has made that point to us. If valid, and it might well be, it could influence how you organize your common systems projects.

Generalized design

The goal in designing common systems is to meet local needs as much as possible and to require local customizing of the common system as little as possible.

The cumbersome way of accomplishing this goal is to gather all of the users' requirements, put them all together, and design a system that incorporates all of them. The resultant system will probably be huge, unwieldy, and hard to maintain.

A preferred approach is to identify and separate the common procedures from the procedures that must have logical variations in order to meet local needs.

We can illustrate this approach by recounting an experience we had a number of years ago. In this instance, we were participating in a project that was studying the structure of a product—a fractional horsepower electrical motor. The company manufactured many models of this motor, each of which had its own set of engineering drawings, parts lists, manufacturing instructions, and so on.

In the study, all of the engineering drawings—for all parts and for all models—were gathered together. Then for any given part (say, the motor shaft), all of the drawings for that part were compared. Each model of the motor had a shaft, and the different models had somewhat different shafts. By comparing the drawings, some interesting things came to light.

Common characteristics. Some of the characteristics were common, from drawing to drawing. For instance, the diameter of all of the shafts for this type of motor were the same; the company had standardized on that shaft diameter.

Logical variations. Some of the other characteristics varied among the drawings, in order to meet the needs of the application for that particular model of the motor. For instance, the lengths of the shafts varied, to meet the needs of the mechanisms in which the motors would be mounted.

"Personal preference" variations. A surprising number of variations were found that had no logical basis; they were just based on the personal preferences of the engineers who had designed the parts. An example might be two shafts that were otherwise identical but one had a rounded end and the other had a square cut end, and where the rounding had no functional purpose.

This particular project came up with a very dramatic result—a master set of engineering drawings, parts lists, and manufacturing instructions from which *all* models of the motor could be manufactured; only variable data was omitted from these documents. In addition, the input to a computer program was a set of values of parameters that defined what the customer wanted; the output was all of the variable measurements, parts list quantities, amounts of materials, etc. When combined with the standard documents, one had all of the information needed to build the motors. It was the epitome of a common system, in our opinion.

The point we are making is that these same types of variations will occur in the requirements study for a common application system. Each of the sites studied will have its own set of requirements, the same as each model of motor had its set of engineering drawings, etc. By comparing the requirements, some will be found to be common and some will vary. The study of the variations will be the challenge, to separate the logical differences from the personal preference differences—the self-interest and "nationalistic" differences. In addition, there probably will be still another cause of differences—wrong assumptions about needs—which will raise problems that will have to be resolved. Such wrong assumptions might show up in statements by one site representative like, "We need this procedure in order to guard against such-and-such happening"—when, in fact, none of the sites can remember cases of that such-and-such actually happening.

So a good part of the generalized design procedure will be devoted to a study of the variations in the requirements. Logical differences, to meet the needs of the businesses, will have to be accommodated. Personal preference differences should be eliminated by selecting one of the procedures as the common one. Wrong assumption differences will be the hardest to reconcile because each one might require a study to determine whether the assumption is valid or not. Again, when differences are due to wrong assumptions, such differences should be eliminated by selecting a common procedure to be used by all sites.

When the logical differences have been identified and isolated, they should be accommodated in as efficient a manner as possible. For instance, in the case examples discussed earlier, The Coca-Cola Company used a standard 9 x 3 matrix for handling all possible categories of value added tax rates. Each user could assign local names and codes to the categories and enter the local tax rates, and the system would handle the value added tax calculations. Similarly, the Scandinavian Shell companies handled a common program design much like modular program design by developing a hierarchical chart of routines. Each routine was identified to indicate whether it was a common routine or was one of several alternative routines and for which sites it was applicable. By developing such a program in modular fashion, individual routines might be changed, added, or deleted with minimal impact on the rest of the program.

So, if common systems appear feasible for you, organize the effort and set up policies which will get all user sites heavily involved, and use a generalized design approach for the systems.

Before leaving this subject, we might mention that we have heard of difficulties occurring with both of these steps. Users hesitate to get too deeply involved with a big information system project, for a variety of reasons; top management support may well be needed in order to get the desired participation. Also, generalized design takes longer than does design to meet one site's needs. If a few of the sites urgently want the new common system, they might argue strenuously against the generalized design. It is quite likely, however, that once they have obtained the generalized design, they will prefer it over a custom design.

If common systems are not yet feasible for you

If common systems look interesting to you but the situation is not yet right for installing them, there are some preparatory steps that you could take. These steps can help pave the way for the later introduction of common systems. We will briefly cover the ones that were described to us, in our research for this issue.

Corporate standards. As pointed out at the NCC seminar and by several organizations we talked to, a strong program of corporate standards can be a big asset. These standards could include design, documentation, and operation. We discussed a corporate standards program in our August 1975 issue.

Application system catalog. One company we talked to publishes a catalog of all internally available software, including application systems. In the past, use of available software has been optional. In the near future, we were told, a much tougher stance will be taken. Users will be *expected* to use available software, instead of developing new software, unless they have good reasons not to.

Master system design. We discussed this approach earlier in this report. Corporate headquarters developed master designs for all of the major application systems used at the company's multiple sites. The sites were asked to compare their existing systems with the master plans and identify the future enhancements that they would make to their existing systems. This approach paved the way for the introduction of common systems.

Develop long range plans. If each site wants to go its own way, have all of the sites develop one, two, and five year plans for new systems and enhancements. We talked to one executive who is using this approach. In addition to the plans, he said he was asking the sites to indicate what they thought the application systems would look like in five years. "We will develop a cohesive plan on a country-by-country basis," he said. "By 1980, I expect that we will be able to pull these together until we have a coordinated set of plans. That, together with corporate standards in development, documentation, and operations, will lead us to common systems."

Buy generalized application packages. Take a good look at the generalized packages that are commercially available. Where they look attractive, you might be able to install them on a site-by-site basis, thus setting the stage for more use of common systems. The Coca-Cola Company, for instance, acquired a generalized general ledger system for use within BASIS—indicating that commercial generalized packages might both help pave the way for common systems as well as play an important role in the common systems themselves.

Common systems *are* within the state of the art for many multi-site organizations. There are now enough success stories to support that view. Even single site users should consider (if they have not already done so) the possibility of buying commercial generalized packages, which are examples of common systems.

The benefits of generalized common systems are many. Be sure to look into them.

Industry has attempted to increase the productivity of its blue collar employees through such steps as the use of capital equipment, automation of certain processes, and methods-time measurement studies. However, it has done comparatively little to achieve similar productivity advances for its white collar workers. The office looks pretty much as it did 60 years ago when the typewriter was introduced. Now, a revolution is occurring in the office. It is known as word processing and it portends to have a dramatic effect, not only upon secretaries and their managers but also upon all white collar workers. Next month, we begin a series of three reports—two on word processing and one on the related subject of computer message systems—to indicate the broadening responsibilities of the emerging "top information *exeuctives."*

EDP ANALYZER published monthly and Copyright© 1977 by Canning Publications, Inc., 925 Anza Avenue, Vista, Calif. 92083. All rights reserved. While the contents of each report are based on the best information available to us, we cannot guarantee them. This report may not be reproduced in whole or in part, including photocopy reproduction, without the written permission of the publisher. Richard G. Canning, Editor and Publisher. Subscription rates and back issue prices on last page. Please report non-receipt of an issue within one month of normal receiving date. Missing issues requested after this time will be supplied at regular rate.

SUBJECTS COVERED BY EDP ANALYZER IN PRIOR YEARS

1974 (Volume 12)

Number

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

1975 (Volume 13)

Number

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

1976 (Volume 14)

Number

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

1977 (Volume 15)

Number

1. The Arrival of Common Systems

(List of subjects prior to 1974 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____________________