DDP MANAGEMENT STRATEGIES: KEYS TO SUCCESS OR FAILURE

By D. S. Appleton

Since the dawn of data processing, man has sought in vain for a solution to the dilemma: Should DP be centralized or decentralized. Some of the best minds in business have groped with this conundrum, searching for the answers. Happily, answers do exist, to be discovered only after the proper amount of time, money and executive brainpower have been expended in the search. Most of the answers, like those from the Delphic Oracle, tend to obscure the question—to all but those trained in the Mysteries (of data processing). The answers are: (1) *It depends,* (2) *the opposite of what you are doing* and (3) *distributed data processing*.

There are plenty of seers who can treat with the first two answers—for a price. Therefore, we won't deal with them here. Instead, we will concentrate on the third answer because it is the only one which actually makes the question go away. By definition, distributed data processing (DDP) is a hybrid concept, taking the best from the centralists and the best from the decentralists and welding them into one DP management strategy.

DDP is the synthesis deriving from the dialectic between centralization and decentralization. Over the past decade or so, this dialectic has been encouraged by a DP community which has taken special pains to identify and separate needs of centralists and decentralists, justifying the development of two seemingly distinct technologies. The results are minicomputers and turnkey systems for the decentralists and maxicomputers and data base systems for the centralists.

It became increasingly obvious that the technologies being developed ostensibly for one camp were actually complementary to technologies in the other. In fact, the market actually began demanding the best of both. Thus came requirements for sophisticated software on minicomputers and low-cost, modular growth ities on maxicomputers.

Since there was no time to develop a third set of technologies (although there were some feints in the direction of midicomputer technology), the obvious answer was that the two technologies should be linked together. The bridge between the two camps has been made via communications technology. This "third world" is the fulcrum of DDP. Whoever controls it controls both centralized and decentralized technology. Thus, at the foundation of DDP is the scramble to establish and control what has come to be known as the "network architecture."

The scenario just described is the context within which we have all come to understand DDP. It is a technological scenario, filled with discussions of star and ring structures, mininodes and maxihosts, SNA and DECNET and jillions of other fundamentally technical issues. But what is missing is the basic strategy needed to manage one of those technological monsters. The wrong strategy can hopelessly bog down the best technology in needless cost and low productivity. On the other hand, the correct strategy can make even the best technologies better and more effective.

DDP Management Strategies

In the business of data processing, there is no metaphysical right or wrong. The right way is the way that works, and the righter ways are those that work longer. Ignorance of this latter truth has been the main chink in data processing's armor. Successes in DP have been plentiful, but sustained successes have been notable by their rarity.

A basic reason for this eventuality is the fact that most DP managers—encouraged by their bosses, subordinates and user groups—manage by using project-oriented strategies. These strategies are great for development programs; but they are terrible for sustaining a responsive, cost-effective business, which is, afterall, the real DP *management* problem.

The author is director of management information systems for Byron Jackson Pump Division of Borg Warner in Los Angeles,

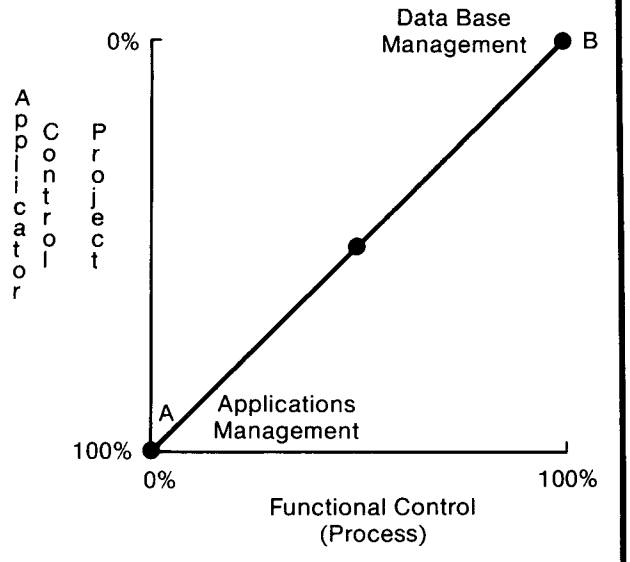

Figure 1 Data Processing Management Matrix

The DP management matrix shown in Figure 1 provides the needed perspective to understand the problem. The matrix juxtaposes project and process management strategies. At Point A, project management strategies are dominant. The project strategy, which we will dub the applications strategy, derives from a tradition which treats data processing as a customized service. It calls for a DDP approach which propagates complete computer applications within the network, defining "distribution" primarily in terms of computer hardware, or processing power. In effect, it "distributes" computer resources and centralizes or decentralizes application systems.

Process management (point B on the matrix) traces its roots to data base concepts originally developed in the latter 1960s. It could be called DDB (distributed data base), but it is better defined as a functional strategy. This strategy seeks to distribute capabilities and responsibilities by DP function rather than concentrating on individual computer applications, such as payroll, A/P, purchasing and so on; it establishes a DDP perspective based on the processes needed to effectively manage distributed input, output and data bases.

It is apparent that few DDP environments are pure in their project or process orientation. Most are hybrid, with one or the other strategy dominating. Thus, we are really confronted with two fundamentally different management strategies. On the one hand, we have the applications DDP network, dominated by the project management strategy. On the other hand, we have the functional DDP network, dominated by the process management strategy.

It is my experience that the functional strategy has significant advantages over the other, and that, accordingly, it should be dominant in most cases.

Applications DDP Networks

A computer application system is a set of computer programs which, taken together, input, store, process and output information required to support specific management situations. These situations are defined in terms of the needs of managers for reports at points in time. Based on these output requirements, DP constructs input procedures and programs, internal processing routines and computer master files, organizing time into self-contained, purposefully oriented computer application systems such as payroll, accounts payable or purchasing. Although initially applications may be developed independently, eventually they become integrated with one another as additional applications.

The traditional DP network, be it a single, batch minicomputer or a complicated on-line, time-sharing, multiple CPU international network, evolves through the continual addition of application systems. Each application creates additional need for network facilities, technologies and capabilities, while at the same time providing the justification and logic for their installation. The network is a byproduct of independent decisions made for specific computer applications over a long period of time. Networks developed in this traditional way we may refer to as applications networks.

It is the nature of the applications network that it be symbiotic with the organization structure and the management personalities present during the construction of each application. This is because the whole structure (including hardware and software) is established to service predefined information needs. Since it is necessary that complete applications be used to satisfy specific information needs, complete applications must be distributed in the network. Thus, we find a migration of computer applications out to the various network nodes and an attendant demand for distributed hardware and software overhead. Each distributed application is unique because the customized information demands it is intended to service are unique, because each manager is unique.

This management strategy eventually leads to DDP networks characterized by the following:

- Technology is introduced into the network for use by individual applications, leaving earlier applications under older, often less efficient and more expensive technologies.
- Since each application has its own special input, output and master files, each generates its own unique overhead.
- Each application at each node must be managed separately from the others.
- Investment decisions for the network are treated as expense items and are made on the basis of the requirement for specific applications, not on the overall needs of the network or the business.
- Capabilities developed for use on one application, capabilities which effect cost or efficiency, are difficult to implement for other applications and even distributed versions of the same applications.
- When newer applications are integrated with older applications, the older technology tends to reduce the impact of the newer; thus older applications impose a natural drag on the whole network.
- The distribution of responsibilities and capabilities is made application-by-application, creating redundancies and additional costs which tend to increase marginal computer costs and offset gains from economies of scale.
- After many applications have been developed, implemented and distributed, each being unique unto itself

and each drawing on the same network resources, the problem of maintaining control becomes extremely difficult. The result is chaos in the network, and chaos generates additional costs and overhead, while dramatically reducing productivity and flexibility.

Notwithstanding, applications networks have broad appeal. They are easy to understand, politically simple and, with the introduction of the minicomputer, easy to justify. More and more they are becoming turnkey, which obscures the DP problem, making their introduction into business a deceptively simple affair. Another part of their appeal is their perception as a sort of "DP pill:" one a day keeps the DP people away. Management is intrigued by the idea that users alone can control applications, keeping them customized to business needs and preserving certain inalienable managerial rights such as privacy and invention.

At the root of this whole applications network philosophy is the idea that each set of information demands must be dealt with in terms of an engineered product, such as a customized application system. Since it is specially built, it must be specially serviced and maintained. Indeed, it must be separately managed, having its own documentation, staff resources, machine resources, data entry expenses, storage and so on. In short, it must be treated as a special project—forever.

With even this limited insight into the distributed applications network, the basic problem is clear. The environment has an inherent self-destruct feature. The environment is built and managed on a foundation of applications which have themselves been built primarily to satisfy output needs. If output needs change, only one of three things can happen: The application dies, the application is changed or a new application is built. Simply put, changes in output generate a lot of work which must be managed. And yet, the need for information is the most dynamic need in business.

Functional DDP Networks

The self-destructive tendency in the applications approach was apparent long before distributed data processing burst onto the DP scene. In fact, the problems inherent in applications management are what gave rise to the concept of data base management. In the final anaylsis, "data base" was, and is, a DP management philosophy intended to counteract the problems generated by its alter-ego, computer applications. Essentially, it is a functional management philosophy which attempts to ignore the individualities of computer applications in favor of their similarities. These similarities occur at the functional level, not the service level, specifically in the areas of input technology, output technology and storage and processing technology.

The data base approach, as it is called, attempts to standardize *processes* used to perform functions common to all applications. The data base approach assumes that the quality, survivability, cost-effectiveness, responsiveness and utility of DP service derive specifically from the ability to competently manage the three DP functions; input, output and storage and processing (the data base). These functions must be performed regardless of the demands for any specific service (application). Each of these functions has distinct characteristics and requires special skills and technologies to perform well. Each consumes its own share of resources and provides its own unique opportunities, liabilities and management challenges.

Resources	*Storage Control*	*Input Control*	*Output Control*
Quantity	Medium	Medium	Medium
Quality	High	Medium	Low
Dynamics	Low	Medium	High
Risk	High	Medium	Low
Responsiveness	Low	Medium	High
Difficulty	High	Medium	Low

Figure 2 Data Base Strategy Resource Requirements

To better understand the functional DP strategy, we need to look more closely at each of the three basic functions. Figure 2 will be an aid in this endeavor because it describes the relative demand of each function on DP resources.

The storage and processing function could be called the prime function of DP. Basically, it involves identifying data, structuring data and managing processing routines that create data which is, in turn, stored. This function defines what *can* be input and what the *potentials* are for output. In a sense, it is the function which controls both the dictionary and the rules of grammar composing the level of "management technology" in a business.

Once the storage and processing function is working properly, it is subject to only minor changes. These changes result from management tuning of the business, upgrading the basic level of management technology. The only time it will undergo major change is when the business mission changes radically.

From a data processing standpoint, the storage and processing function is the most difficult to perform, requires the highest level of technology, absorbs the greatest amount in internal DP overhead and carries the highest risk if mistakes are made. As a result, it requires the greatest amount of attention and the highest investment. (See Figure 2.)

The storage and processing function defines the available levels of management technology. To be more than just potential, however, it must be implemented through the input and output functions. These are more clearly tied to the current demands and levels of sophistication of the business than is the storage and processing function. They define how, when and to what degree the storage and processing function is implemented in various business products, organizations or locations.

The input function is habitually the biggest problem area for DP. The reasons are that input, by definition, generates overhead expenses in areas and it requires a high degree of discipline and control and DP. Currently, the input function is characterized by a wide variety of unimaginative technology. It has been grossly over-simplified, misunderstood and underestimated in its importance, and it is probably the weakest of the three DP functions; however, it is of no less importance than either of the other two.

Properly used, input is a basic mechanism for distribution of the management technology capabilities of the storage and processing function. Through input control, capabilities of the storage and processing function can be selectively implemented, and this implementation can be closely coordinated with the ability of the business to

absorb the cost (overhead) of various levels of management technology. Reduced input costs, through improved DP management, can make higher levels of management technology available to smaller portions of the business.

Whereas the storage and processing function varies in accordance with overall business strategies, the input function tends to respond to variations in product mix and levels of activity. This is because timing, accuracy and volumes are the basic determinants of input control. Input controls must be distributed and adjusted based on changes in these areas, rather than changes in the demands for output. The latter are independent of the former, and must be kept that way.

The third DP function is that of output. The need for computer reports is directly relatable to management's need to make decisions. Obviously, this need is not only dynamic; it is almost unpredictable, being, as it is, a result of internal and external influences, economics, politics, personalities and so on.

Traditionally, the output function has been the dominant DP function. It has been so dominant that literally everything DP did was keyed to the need to deliver output. Applications were constructed on the basis of output needs; input was defined on the basis of output needs, and methods for storing and processing data were designed on the basis of specific statements of requirements for computer reports.

The fallacy in this approach is that output is so dynamic, it has to be artificially frozen long enough to set up the input and master file structures necessary to support it. Freezing output is self-defeating not only to the business, but to DP as well.

The real objectives are to be able to generate information quickly and economically and to deliver that information in the appropriate form to the requester. This responsibility is important in and of itself, regardless of what the information content is. Properly approached, the output function will be performed effectively, regardless of what attributes of the storage and processing function are available to it. The tools and technologies of output are independent of storage and processing, and will respond whatever their level of implementation.

Referring to Figure 2, again, it is apparent that, relative to either the input or storage and processing function, the output function is the most dynamic, lowest cost, lowest risk and lowest overhead DP function, and it requires the lowest level of technical skill. However, it is the most visible, and it requires the highest responsiveness. In all of these areas, it is exactly the reverse of the storage and processing function.

The overall objectives of the functional network are to distribute capabilities and responsibilities based on current management needs and to provide the flexibility needed to respond effectively to changes in those needs.

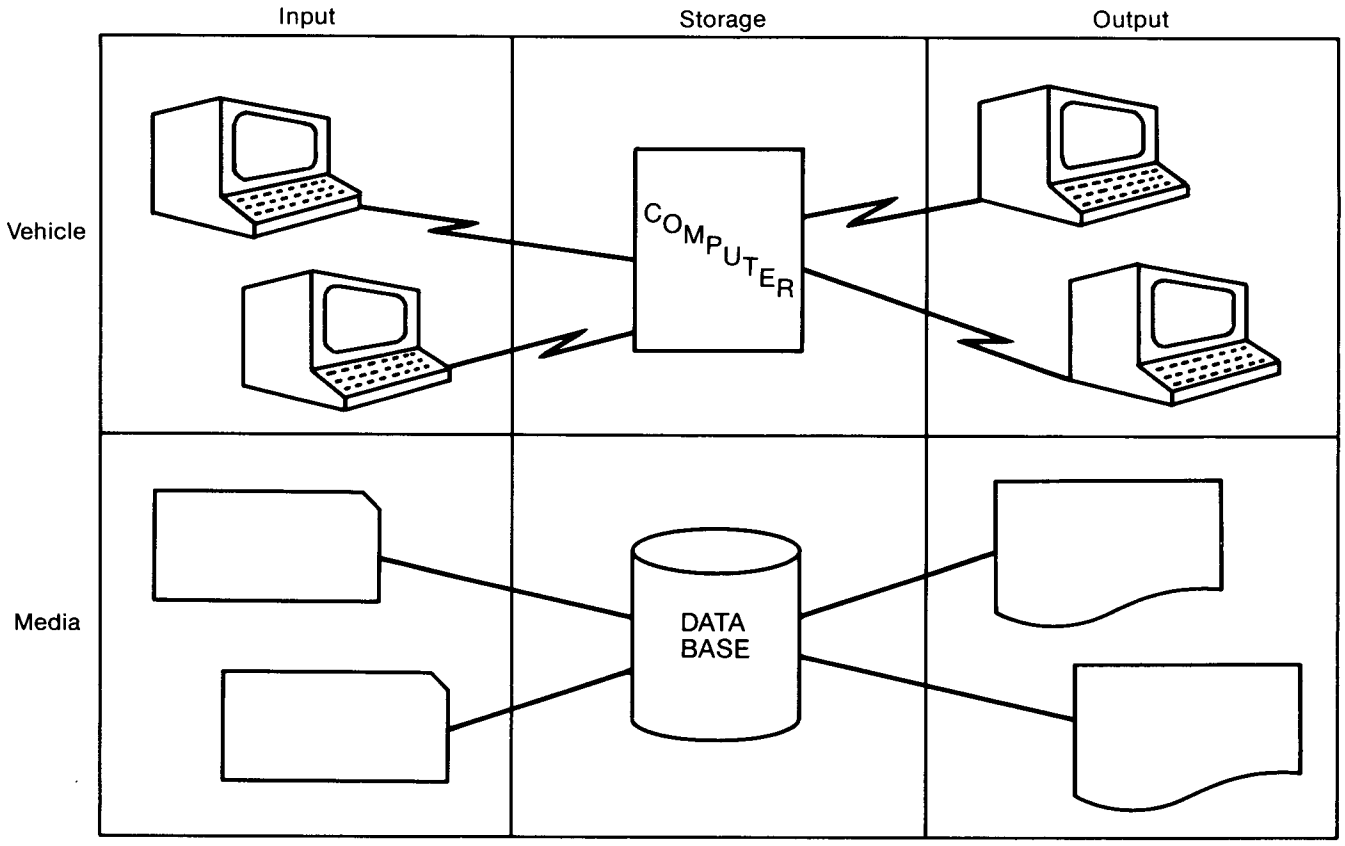

Figure 3 Functional Network Architecture

To fully understand how these objectives are realized in the functional environment, it is necessary to introduce two new network concepts: the network vehicle, the electromotive parts of the network, including CPUs, telecommunications lines, programming languages, operating systems and so on, and the network media, computer data bases and special programs which take input, store and process data or generate output.

Figure 3 shows the relationships between the physical and functional structures of a functional DDP network. This perspective allows us to think about, say, the input vehicle or the output media, and to use these concepts for our DDP management strategy. We can thus establish procedural controls for each individual area, as well as the interfaces between them. At the bottom line, we have established a foundation for network management which allows for a great deal of management flexibility. Just separating the vehicle and media as management problems is a tremendous step forward. But, adding to that separation—which is not just logical; it can also be done physically—the functional separation provides tremendous management advantages.

The overall objective of the functional DDP network is to distribute capabilities and responsibilities to various network nodes based on the requirements to perform specific DP functions at a node. Figure 4 describes this concept in terms of main nodes and service nodes. The main node could be the central node in a star structure or any designated node in a ring structure. It could be the main node for the vehicle or the media, and it would be responsible for managing the storage and processing function.

The effect is to create a DP environment wherein the main node is responsible for the high technology, high overhead and long lead-time functions, while the service nodes (they might be called DP service centers) are responsible for quick reaction to user output requirements and for maintaining the highest level possible for input quality. The latter are short lead-time, low technology, low overhead functions, requiring a good mix of quick response and efficient control.

Functions	*Storage Control*	*Input Control*	*Output Control*
Administration	MN	MN	MN
Cost Control	MN	SN	SN
Programming	MN	SN	SN
Analysis	MN	SN	SN
Training	MN	SN	SN
Development	MN	SN	SN
Maintenance	MN	SN	SN
Security	MN	SN	SN
Implementation	MN	SN	SN
Documentation	MN	SN	SN
Processing	MN	SN	SN
Storage	MN	SN	SN

MN—Main Node
SN—Service Nodes

Figure 4 Data Base Strategy with MIS Service Center Concept

Process Vs. Project Management

It is apparent from the above discussion that the functional network has potential advantages far beyond those of the application network. These potentials stem primarily from the process versus project management approach and can manifest themselves in many important ways, such as the following:

Lower DP Marginal Costs. DP marginal costs are incurred when new service capabilities are added or current services are expanded or changed. In applications networks, this is done application by application, requiring either the addition of complete new applications or the duplication or modification of existing ones. Functional networks treat these problems in terms of process improvements, such as modifications to the ICS, OCS or DBCS. Since each of these areas is standardized, not only are the marginal costs of the change reduced, but so are those future costs associated with maintaining the change.

To put it another way, the functional approach provides enough stability so that the DP department can deal in standard costs, while applications networks are controlled by job costing. The comparison can be taken further. The expense profile (indeed the whole management philosophy) of applications network is identical with that of a manufacturing job shop, while the functional network closely parallels the manufacturing process shop.

More Compatibility With Business Growth And Dynamics The applications strategy, as described above, creates an inflexible DP environment because it is constructed to service specifically defined needs. Changes to the business which require systems changes, such as changes to the organization structure, activity level, product mix or management philosophy, occur very frequently because of this rigidity. The effects are increased costs and long lead-times needed to change inflexible systems, constraints on business changes because of the inability to change systems and an increased rate of atrophy of the systems because of the constant changes needed to keep them in tune with business dynamics. On the other hand, functional networks are built with change management as a design criteria. Most changes can be accommodated in the ICS or OCS without ever affecting the DBCS. Changes which do affect the DBCS are made by technically competent people in a controlled environment.

More Efficient Use of DP Overhead DP, like all other functions, supports a tremendous amount of internal overhead. In DP, this overhead is most obvious in computer operations; however, a critical overhead item, which many companies don't recognize, is the utilization of computer capacity for overhead functions such as accessing data in discs or tapes, printing reports, compiling computer programs and so on. Both departmental and equipment overhead must be used efficiently. The objective, as in any business, is to gain maximum quality at minimum cost and try to capitalize on economies of scale.

In the applications network, achieving these ends is difficult because each application generates its own overhead demands. In the functional network, overhead demands are created by the control systems. Because of the structure, more overhead can be centralized in a functional network than in an applications network. As a result, the opportunities for quality, low-cost overhead with wider applicability and greater economy of scale are better in a functional network than in an applications network. Fur-

ther, functional networks require more standardized overhead and provide better overhead control.

More Effective Use of Direct DP Expenses Direct DP expenses are those specifically associated with providing a needed service at a specific point in time. These expenses should be incurred as close as possible to the point of consumption. Accordingly, they should be decentralized as much as possible. In an applications network, these expenses are difficult to isolate from overhead expenses because they, too, are incurred application by application. Further, they must be incurred at the level (centralized or decentralized) of application control. Most direct expenses are incurred by input and output requirements. The functional strategy not only keeps these expenses down because of standardization and control in the I/O areas, but it assures that they are incurred where the service is consumed. Further, these expenses are more clearly defined and more controllable.

Better Control Over the Infusion of Technology Technology is introduced into applications networks based on the specific or accumulated needs of individual applications. Again, this approach is similar to that used in a manufacturing job shop. In a functional network, however, technology is introduced based on the requirements of the individual control systems (ICS, OCS, DBCS). Management technology (that is, the ability to use certain management techniques) is introduced through the DBCS by adding data elements, modifying data structures and adding or changing internal processing routines.

Reduced Implementation Times In a functional network, development and implementation are accomplished by modifications to the three control systems. In the applications network, these same activities require, in most cases, building up from scratch. In many cases, implementation in the functional network requires only that the input and output control be defined, not that new master files be constructed and new processing routines be developed. These latter activities are extremely time-consuming and expensive. This is especially true if the development methodology ties them closely to the output requirements, and the output requirements often change before implementation is completed, requiring file and processing changes and the appropriate additional time.

Easier Customization to Varying Requirements It is difficult to take a complete application built for one environment and modify it to work in another environment. This is because of the effect the design methodology has on the final product. In essence, the whole application, including input, output, file content and structure and processing routines, is customized. In the functional network, the DBCS is standardized, and customization is accomplished by modifying the I/O controls. This is the same strategy used by companies which sell "proprietary software," such as payroll systems, accounts payable systems and so on. The product structure is controlled at the data base level, and various customized implementations are made by changing input and output. Most of this "proprietary software" is designed and built using functional development techniques, such as prototype development, rather than the standard, in-house applications development approach.

Higher Quality Systems System quality is not just a function of compatibility with user requirements. Of equal importance are flexibility and efficiency. If the user environment were static, these latter factors would be of less importance, but it is not. In fact, it is so dynamic that compatibility is more of an ideal than a practical goal. The user environment is a moving target, and high-quality systems are able to keep that target in focus, while lower-quality systems are constantly trying to catch up. The functional strategy is more conducive to high-quality systems because of its perspective. It forces a logical separation among input, output and the data base, thereby creating a higher degree of flexibility. It introduces technology function by function, thereby improving efficiency through use of the I/O control systems. Functional networks can make the adjustments necessary to maintain better compatibility with a dynamic user environment.

Easier Maintenance DP maintenance is the ability to sustain a certain service level. Computer systems, like all other systems, are subject to the Second Law of Thermodynamics: They naturally tend to become less and less efficient as entropy increases. There is a need for a constant infusion of new energy through corrective and preventative maintenance. In the applications network, maintenance is performed application by application. To perform this maintenance, individual applications expertise is necessary. The problem is complicated when individually developed applications are integrated with one another, changes in one requiring changes in others. What might normally be a minor change to one individual application can turn out to be a major change to a maze of computer programs supporting a number of systems. Functional networks, again, overcome this problem by redefining it in terms of the three functional control systems. Expertise is developed, regardless of specific applications, in each area. Because each is standardized, the maintenance problem is simplified.

The lessons of process and project management strategies discussed above are not new by any stretch of the imagination. They are fundamental management strategies for any type of business. The management theorists teach us that project management techniques (such as matrix management, job shop management) is a very workable, appropriate strategy for highly dynamic, creative environments. Process management, in contrast, is appropriate for mechanistic, high-volume production environments. Ironically, the shift in DP toward data base concepts is a tacit acceptance that business data processing is passing from the highly dynamic, creative stage to being a more routine, mundane management problem, requiring a more appropriate management strategy. The problem is no longer in trying to find new things to do, new projects to undertake; it is in doing what we do better.

It is the contention of this article that, given the complicated interlinkage and interdependencies of DDP networks, the appropriate management strategies are not applications-oriented. Indeed, to use an applications approach is to be counterproductive to the basic objective, which is to construct systems that can survive the natural dynamics of the business. Actually, the ultimate objective of distributed data processing is not just to survive, but to help the business respond more quickly and cost-effectively to natural forces for change. Applications strategies have difficulty just surviving. Process management concepts, what have been called here the functional strategy, give us a goal to shoot for, and, as Dr. Peter has said, "Without a goal, you will end up somewhere else."