ESTABLISHING MANAGEMENT OBJECTIVES

The pace of the assimilation of computer technology into the data processing organization is represented by the S-shaped "Data Processing Learning Curve" (see Figure 1). The curve is approximated by the growth of the data processing budget and reflects the staged evolution of the data processing environment along four growth processes:

1. *The portfolio of computer applications.* The programs and procedures which are used by the organization in its business activities. The applications portfolio represents the cumulative end product of the data processing organization.
2. *The data processing organization and technical capabilities.* The organization structures and technical capabilities found within the data processing department which are required to develop and operate application systems. These include data processing management structure, hardware and software resources, systems development and operations organizations.
3. *Data processing planning and management control systems.* The set of organization practices used to direct, coordinate and control those involved in the development and operation of application systems, including data processing planning, project management, top management steering committees, chargeout and performance measurement.
4. *The user.* The members of line and staff departments who must use the applications systems to perform their jobs.

The nature of data base management systems (DBMS) dictates that they interface with each of the four growth areas. First, the DBMS acts as the data manager for all types of application systems. Second, the DBMS introduces a new level of technology to be assimilated by the data processing organization and it calls for the introduction of a new data processing organization structure: the data base administrator. Third, the DBMS requires that application system planning be more comprehensive and that control through "chargeout" be restructured to reflect shared resource usage. Fourth, the DBMS may impact the user by providing new functional capabilities and mechanisms for data retrieval.

Because of the integral position occupied by the DBMS in the systems environment, the conversion to data base technology and its use should be carefully managed. In managing the conversion to data base technology, data processing managers should have a well-articulated set of objectives regarding each of the four data processing growth processes. These objectives should form the foundation for goals against which data base conversion activities are measured.

The initial objective of the Establishing Management Objectives panel was to analyze the impact of the data base conversion effort on each of the four data processing growth processes. Based on this analysis, the panel then determined the management considerations associated with data base conversions. The management considerations identified by the panel can be summarized into four key concepts:

1. *DBMS conversions are a matter of "when?" not "whether?"* Conversion from a nondata base to a data

---

The panel chairman was Richard L. Nolan of Nolan, Norton & Company, Inc. Other participants were Marty Aronoff, National Bureau of Standards; Richard G. Canning, Canning Publications, Inc.; Larry Espe, Nolan, Norton & Company; Gordon Everest, University of Minnesota; Robert Gerritsen, University of Pennsylvania; Richard Godlove, Monsanto Company; Samuel C. Kahn, E. I. duPont de Nemours & Co.; H. Eugene Lockhart, Nolan, Norton & Company; John Lyon, Colonial Penn Group Data Corporation; Thomas E. Murray, Delmonte Corp.; Jack Newcomb, Tennessee Department of Finance & Administration; T. William Olle, consultant; Michael J. Samek, Celanese Corporation; Steven Schindler, University of Michigan; Richard D. Secrest, Standard Oil Company (Indiana), and Edgar Sibley, University of Maryland.

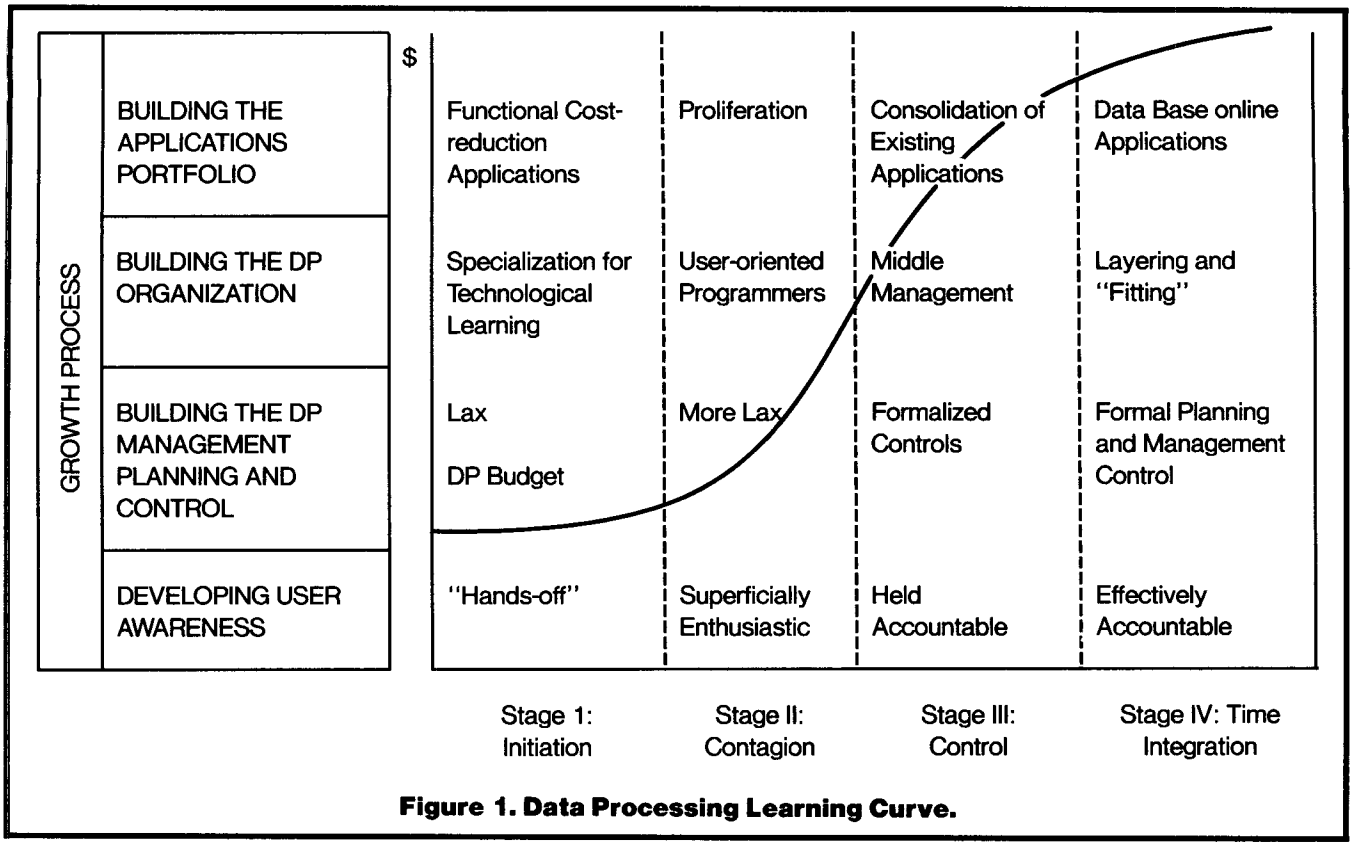

Figure 1. Data Processing Learning Curve.

base environment is a part of the natural evolution of data processing within an organization. A data processing department which has matured and progressed to a Stage III environment is typically faced with high maintenance costs and an inability to respond to ad hoc inquiries and requests from user management for integrated reports. This situation, in effect, forces the data processing department to employ data base technology to restructure the applications portfolio. In other words, conversion to a DBMS is primarily a question of how soon the data base environment should begin to be constructed, not whether a data base environment should be implemented.

2. *Choose the DBMS conversion application carefully.* The initial application used for conversion to data base technology represents an important learning experience for the entire organization. As such, the initial application should be a nontrival application, should demonstrate the "power" of the DBMS facilities and should be simple to avoid overextension caused by attempting to do too much too fast.
3. *Treat the initial and subsequent DBMS conversions similar to other systems projects.* Although a data base conversion introduces a new technology to the organization and requires the involvement of all areas in the systems environment, the risk exposure of this conversion effort can best be minimized by managing the conversion as any other large project would be managed. The same planning and justification procedures should be used. The same project management mechanisms should be exercised throughout the project life cycle. Because of the major impact caused by a data base conversion, special efforts should be made to coordinate conversion activities with steering committees, senior management and user areas.
4. *Plan and structure for future DBMS conversions now; DBMS conversions will be a way of life.* Certainly, once a DBMS is successfully installed, the conversion of applications to that DBMS will continue. However, the mature organization should also plan on converting to *another* DBMS at some point in time. The second DBMS may be just an enhanced version of the first DBMS, or it may be a totally new software package. In either case, it is certain that a mature data processing organization will want to take advantage of new DBMS facilities and efficiencies; therefore, DBMS conversions will become a way of life.

To prepare for these continued conversions, the data processing organization can take several steps to minimize their impact. Minimize the application system processing logic, program code and data base design dependencies on the features of a particular DBMS. Institutionalize the data administration function. Fully document all business

system functions on an integrated dictionary.

The overall DBMS conversion philosophy developed by the panel can be summarized as follows: *Appreciate the technology, but recognize that DBMS conversion is a management problem.*

The panel approached the topic of data base conversion in a chronological manner. As such, the following sections are organized to reflect management considerations during the life-cycle data base conversion efforts.

A certain level of maturity is necessary before conversion to a data base environment is feasible. In general, data base technology is not appropriate for data processing departments in Stage I or early Stage II environments. With these exceptions, conversion to a DBMS should be initiated as soon as possible. However, the Stage III environment is most compatible with the initial conversion.

Two basic approaches exist for conversion of the application portfolio to a data base environment: revolutionary and evolutionary. Actually, these two approaches represent opposite ends of a continuum of approaches. No single approach is universally best. In fact, more than one approach may be operable within a given conversion; certain application systems may be converted on a revolutionary basis, while other systems are converted in an evolutionary manner.

In the revolutionary approach to conversion, sometimes called resystemization, one rewrites and restructures existing application systems as necessary to operate under the new DBMS. Generally, one should avoid exclusive use of this approach for the following reasons:

- Risks overextension caused by attempting too much too fast.
- Delays in one subproject may impact others.
- Insufficient resources may be available for developing new systems during the conversion.

At the opposite extreme of the revolutionary approach is the evolutionary approach in which all new systems are developed under the new environment. Existing systems are not converted but rather are replaced at the end of their normal life cycle. This approach reduces the risk of overextension and the impact of delays in subprojects. However, there are disadvantages to an evolutionary approach:

- Complex interfaces with existing, conventional systems are generally entailed.
- Local inefficiencies and redundancy typically result.
- Current organizational deficiencies and constraints may be perpetuated.

Just as the conversion of the entire applications portfolio may be approached in an evolutionary or revolutionary manner, so may the conversion of a single, existing application system. In other words, the entire application system may be converted to the new environment at one time, or the conversion may take place in phases. The latter approach, in which the reporting and update functions are converted gradually, using bridges, has the advantage of early availability of both cross-functional data and the new features of the DBMS. Furthermore, greater flexibility in scheduling the conversion is provided. However, the gradual approach has the disadvantage of redundant development and data storage and requires increased management to provide and control the conversion bridges.

One temporary measure that may be employed to avoid or postpone conversion is to extract data from existing master files to build a transient, integrated data base. This data base is not maintained, but instead is recreated on a cyclic basis. The data base is used for cross-functional reporting and analysis. This approach provides early availability of cross-functional data and lends itself to a specialized interrogation language. At the same time, there are certain disadvantages to this approach:

- Availability of data is achieved at the expense of redundancy and reloading.
- Problems of timeliness and consistency may be created.
- Basic application limitations are perpetuated.

There are never sufficient resources, nor is it appropriate, to permit continued maintenance and enhancements of application systems during the conversion to the data base environment. Moreover, conversion requires a relatively stationary target. Thus, a moratorium on maintenance (or more accurately on enhancement) may be declared during conversion.

The declaration of a maintenance moratorium must be the result of agreement among user, senior and data processing management. Senior and user management support is necessary for the conversion. However, if senior management is the primary motivator behind the conversion, there will be some degree of user resistance to a maintenance moratorium. On the other hand, if the conversion is driven by user management, senior management will tolerate a moratorium on maintenance only so long as it does not interfere with normal business functions. In either case, user and data processing management must jointly determine the scope and duration of the moratorium and agree to the circumstances under which it may be modified or cancelled.

A common device for invoking moratoriums is a steering or priorities committee. Composed of data processing and user management, the steering committee is responsible for approving projects and establishing priorities. The steering committee does not manage, nor does it relieve management of its business responsibilities. Rather, it provides a forum for discussion and has power derived from its membership and sponsorship.

Certain application systems indicate better opportunities

for conversion than others. The following types of applications represent good opportunities for conversion:

- An application system using many different master files and/or many internal sorts, indicating the need to represent complex data structures and to support multiple paths between data.
- An application with a requirement for online inquiry and/or update of interrelated data. A DBMS would still be applicable, although not required, if the data were not interrelated.
- An application system with chronically heavy maintenance backlogs, suggesting redundant data and/or inflexibility with respect to its data structures.
- An application system requiring a broader view of data (either more detail or greater cross-functional breadth).
- An application which crosses functional or organizational boundaries (such as project control).
- An application which cannot support basic business needs.
- An application which provides data used by other systems.

Certain types of applications represent poor opportunities for conversion. For example:

- A purchased application which is maintained by a third-party supplier.
- An application which uses historical data and which is processed infrequently.
- A recently installed application system which is effective in satisfying user needs.

An analysis of the characteristics of the existing applications based on the above considerations will yield a preliminary ordering for conversion of the application portfolio. As the conversion is planned in more depth, the preliminary ordering will be revised and refined to reflect such factors as precedence relationships regarding conversion, level of effort required and the availability of resources.

In converting to a data base environment, a key decision is selecting the entry-level application. In an ideal world, the initial application would be selected as the vehicle for making mistakes and learning how to convert and how to manage the conversion. It would have a low profile and not present any risk to the business. However, the realities of the world will force initial conversion of a system which has visibility, contains some element of risk and which must be completed quickly. The factors listed below should be considered in identifying the best opportunity for developing technical competence while simultaneously reducing risk and visibility:

- The application should be representative and nontrivial.
- It should be a good DBMS application (although not necessarily the best).
- It represents a relatively low risk to the business.
- It provides sufficient opportunity for learning.
- It is either an old system or technically obsolete.
- It provides eventual visibility as a vehicle for management controls.
- It is "owned" by a vocal, important, but neglected (by data processing) segment of the business.

Converting to a data base environment generally entails reorganization of the data processing function to provide the technical and administrative means for managing data as a resource. A key organizational consideration is the need to establish a data base administration (DBA) function within data processing. Conversion will also impact the applications development and computer operations functions within data processing.

The DBA function is responsible for defining, controlling and administering the data resources of an organization. The many responsibilities of the DBA function are not discussed in detail here since they are covered extensively in the literature. However, some of the major responsibilities include the following:

- DBA must have primary responsibility for defining the logical and physical structure of the data base, not merely consulting responsibility.
- DBA is responsible for protecting the physical existence of the data base and for preventing unauthorized or accidental access to the data base.
- DBA must monitor usage of the data base and collect statistics to determine the efficiency and effectiveness of the data base in satisfying the needs of the user community.
- DBA must mediate the conflicting needs and preference of diverse user groups that arise because of data sharing.

Many alternatives exist for locating the DBA function within the overall corporate structure. Three such alternatives include the following:

- To avoid an application orientation or an emphasis on computer efficiency, DBA should, in general, not report to applications development or computer operations. Rather, DBA should report to the highest full-time data processing executive.
- When located at the highest corporate level, DBA can take a broad view of data as a corporate resource. Furthermore, DBA is in a position to resolve conflicts between user areas. When DBA resides at this location, some of the more technical aspects of the DBA function are typically performed within the data processing organization.
- The matrix structure is patterned after the aerospace industry where a given project draws upon all functional areas. In this case, the DBA staff would report functionally to DBA, but would also report directly to a project manager. This organizational strategy has the

advantages of recognizing the integration required for a data base, puts DBA at an equal level with other functional areas and serves to increase communication during application development.

Within the DBA function, the two basic organizational strategies are functional specialization versus application area specialization:

- Functional specialization organizes DBA according to functions performed, such as data base design, performance monitoring, data dictionary and so on. This approach has the disadvantage of insuring that no one person is knowledgeable about all aspects of DBA support for a particular application system.
- Application area specialization has one person within DBA responsible for performing all DBA functions for a particular application area, including both application development and operation. This approach has the disadvantage of developing expertise within functional areas of DBA more slowly. Furthermore, unless controlled, activities within DBA may become fragmented. However, this approach results in an interesting and challenging job and facilitates attracting and keeping capable personnel.

Conversion to a data base environment will affect the applications development function within the data processing organization in several ways. The most fundamental impact will be the change from an applications orientation to a data orientation. Conversion to a data base environment should also broaden the scope of the application developers. Specifically, the developers need to understand the basic business processes and to develop application systems that cross organizational boundaries.

The application development methodology will have to be modified by delimiting the relative responsibilities of both DBA and applications development. Moreover, the basic approach to application development may be revolutionized as a result of conversion to a DBMS. Specifically, instead of a rigorous approach to application development, the DBMS may permit an iterative or convergence approach. With this approach, user requirements are not defined in detail before developing the application system. Rather, user requirements are defined at a more general level, and a system is quickly built using the DBMS. When presented with the system outputs, the user specifies any required changes, which are then incorporated into the system. This process is repeated until the application system satisfies user needs. Note that this approach to application development requires a DBMS in which data base definition, creation and redefinition and report writing are quickly and easily accomplished.

Conversion to a data base environment will impact the computer operations function in two ways. First, many of the responsibilities of computer operations will be transferred to the newly established DBA function. Second, the characteristics of the application systems may change. Specifically, the DBMS may facilitate the development and operation of online applications as opposed to the more traditional batch systems. Consequently, the computer operations function may have to reorganize to operate within this more dynamic environment.

Because the subject of DBMS selection has been covered adequately in the literature, it was not addressed by this panel. However, the following are some of the tools and methodologies typically required in making effective use of the DBMS after its installation:

- Data dictionary/directory is a tool for organizing, documenting, inventorying and controlling data. It provides for a more comprehensive definition of data than is possible in the DDL facility of most commercial DBMS. As such, it is essential for management of data as a resource.
- Data base design and validation tools are used to facilitate the design process and to validate the resultant design prior to programming. Included in this category are such tools as hashing algorithm analyzers and data base simulation techniques.
- Performance monitoring tools are useful in analyzing and tuning the physical data base structure. These tools provide statistics on data base usage and operation.
- Application development tools are used to facilitate the development of application systems, including such tools as terminal simulators which operate in batch mode and test data base generators.
- Data base storage structure validation utilities are used to verify that a stored data base conforms to its definition or to assess the extent of damage of a damaged data base. Examples include a "chain walker" utility.
- A query/report writer facility enables users to access the data base and extract data without having to write a procedural program in a conventional programming language.
- Data base design methodology is needed to standardize the approach to data base design and to provide guidance in using the data base design, modeling and monitoring tools.
- Application development methodology specifies the standardized approach to developing application systems, such as the activities to be performed during the development process and the corresponding roles and responsibilities of each of the various project participants. Of particular importance is the need to define the points in the development process at which DBA and applications development functions must interface and the relative responsibilities of each with respect to application development.
- Documentation methodologies are needed by DBA to

document data definitions uniformly and to document data base design decisions.

The following types of personnel skills are needed in a data base environment:

- Data base administration should be staffed with individuals who are strong technically, interface well with people and are knowledgeable about the DBMS itself, the tools necessary to support it, the application development process and the corporation and its data.
- Within DBA there is a need for individuals possessing the ability to recognize and catalog data elements, to group related data elements, identify relationships between groups and to use the data description language.
- Within DBA there is a need for individuals knowledgeable with respect to organization techniques, data compression, tradeoffs in data base design, simulation and modeling techniques.
- DBA should include individuals with knowledge of the DML and its associated host language, data base navigation and the currency concept.

Obviously, the required skills may be developed internally or acquired externally. Hiring the required personnel has the advantage of bringing experience and new ideas into the data processing organization. However, individuals knowledgeable with respect to DBMS are scarce and hence expensive. Moreover, individuals brought in from the outside typically have little, if any, knowledge of the business.

Developing skills internally has the advantage of building DBMS skills on top of knowledge of the business. Furthermore, control can be exercised over what is learned and when. Finally, it is generally less expensive and disruptive than hiring.

When skills are developed internally, there are several possible approaches to training. With an in-house approach, staff personnel possessing the necessary skills teach these skills to others by means of courses or joint projects. This approach may fit well with initial application development and has no cash cost. Moreover, the mere act of having to teach their skills to others enhances the knowledge and understanding of the teachers themselves.

There are several disadvantages to this approach. It requires the time of the most capable personnel when they may be more effectively used elsewhere; it cannot be used where the required skills do not exist internally; as a closed system, it excludes differing points of view.

The vendor approach utilizes the courses offered by DBMS and support software vendors. Vendor courses may be a relatively inexpensive approach, particularly when courses are bundled as part of the purchase/lease price. Furthermore, the internal staff is likely to benefit from the expertise of the vendor. However, the courses may be only a thinly disguised sales pitch.

Additional approaches to training include independent educational organizations, colleges or universities and videotape/cassette courses.

Conversion to a data base environment may result in employee turnover. The new DBMS may be perceived by the staff as being threatening and, hence, may be resisted. This resistance to change may be overcome somewhat by involving the staff in the series of decisions leading to the acquisition of a DBMS. If required skills are obtained through hiring, the existing employees are likely to resent the high salaries paid to the new employees. Finally, as the skills of the staff increase, so does the market value, and it becomes increasingly expensive to retain the staff. These three factors—resistance to change, resentment of new hires and increased employee market value—tend to increase turnover following conversion to a DBMS environment.

On the other hand, certain factors tend to decrease turnover. Conversion to a data base environment involves new opportunities for individual growth and excitement such as new technology, new hardware and software and major development efforts. Properly exploited, these factors can increase job satisfaction and correspondingly decrease turnover.

With the exception of the chargeout mechanism, conversion to a data base environment will not affect the basic mechanisms for planning and control. However, recognize that conversion is itself a process to be managed. This entails applying justification procedures for conversion, planning the conversion, establishing review and approval checkpoints and monitoring progress.

Planning for the conversion requires the involvement of senior, user and data processing management. Attempts to convert to a data base environment without senior management support runs a high risk of failure. If senior management has not formally authorized DBMS studies or incorporated DBMS planning into corporate plans, the probability of successful conversion is remote.

Conversion will have a significant impact on user departments in the form of disruption of normal data processing services, restructuring of application systems and a change in orientation on the part of users from ownership to sharing of data. Consequently, user involvement in planning the conversion is critical.

Conversion to a DBMS generally affects the structure, system development methodology, personnel skill requirements and hardware/software configuration of the data processing function. The lead time necessary to develop the appropriate infrastructure for operating in a data base environment must be appreciated and planned for accordingly.

Given senior management support for the conversion, one strategy for obtaining the required involvement in the planning process is to establish a steering committee for the

data base as mentioned in the earlier section on maintenance moratorium. This steering committee contains representatives from both user departments and from data processing and is responsible for controlling the evolution of the data base. As such, the data base steering committee is subordinate to the data processing strategic steering committee, which is concerned with the evolution of the entire data processing function within the enterprise.

Given the appropriate participation, a necessary first step in converting from a nondata base to a data base environment is the development of an architectural plan for the data base. This plan describes the intended structure of the target data base. Conceptually, a data base represents a model or image of the organization which it serves. For the data base to represent an accurate image of the organization, it is necessary for the data base structure to reflect the fundamental business processes performed in the organization. Consequently, the designers of the data base must first understand the key decisions and activities required to manage and administer the resources and operations of the enterprise. This typically entails a cross-functional study of the enterprise to identify the business processes and information needs of the various user departments.

The architectural plan permits planning and scheduling the migration of application programs, manual procedures and people to a data base environment. This implementation plan must incorporate review and approval checkpoints that enable management to control and monitor the conversion process.

The actual conversion to a data base environment is effected by a project team composed of representatives from user departments, applications development and data base administration. At formally established checkpoints during the conversion, the data base steering committee reviews the progress of the project. Items reviewed and analyzed include the following:

Projected benefits versus actual benefits; data quality such as completeness, timeliness and availability; projected operating and development costs versus actual costs; actual costs of collecting, maintaining and storing data versus benefits realized, and performance of the project team against the conversion schedule and budget.

Based on the review, the data base steering committee takes the appropriate approval action (go/no go) with respect to the conversion activity.

The costs of operating in a data base (shared data) environment are extremely difficult to charge back to individual users in an equitable manner. At best, complex job accounting systems can only approximate actual resource usage. Moreover, the chargeback algorithm must not be dysfunctional with respect to its impact on the various user departments.

Conversion to a data base environment frequently requires that a user department supply data which it does not itself use. The chargeback algorithm must reward, not penalize, such behavior on the part of the user department.

Some considerations in developing an appropriate chargeback algorithm include the following:

- Consider capitalization of the costs of conversion instead of treating such costs as current expense to avoid inhibiting user departments from undergoing the conversion.
- User departments typically have little control over the costs of conversion. Consequently, consider treating such costs as unallocated overhead, since allocation will have little effect on the decisions or efficiency of the user departments.
- Because ongoing costs of collecting, maintaining and storing data are difficult to associate with individual users, consider developing percentage allocation factors for these costs based on periodic reviews of data base usage. Alternatively, consider treating these costs as overhead.
- Resource usage for retrieval and processing purposes are easier to approximate, and such costs should be charged directly to the users.
- Consider incorporating a reverse charging mechanism into the chargeback algorithm to compensate users who supply data which they do not use.

The most fundamental impact of conversion to a data base environment is the required change in orientation on the part of users. No longer are files and applications "owned" by a particular user department. Rather, data must be viewed as a corporate resource to be shared by all user departments. The requirement for sharing constrains the freedom of a user to change arbitrarily and unilaterally the definition of the data.

Sharing of data will impact users in a second way. Following conversion to a data base environment, users may be required to supply data which they themselves do not use. As already discussed, the chargeback algorithm must be structured to reward such behavior. Furthermore, suppliers of data in general must be infused with a sense of responsibility (not ownership) for the data to maintain data quality.

Conversion to a data base environment may impact the user community in other ways:

- Planning the conversion. User participation in developing both the architectural plan and implementation plan is necessary to obtain user commitment and to insure that the resultant data base satisfies user needs.
- Disruption of operations. Normal data processing services are likely to be severely disrupted as a result of such factors as limited availability of personnel and maintenance moratoriums. Furthermore, the conversion may disrupt and strain user operations as new and old applications are operated in parallel.
- Resolution of inconsistencies. Creation of the data base

typically entails merging of application-oriented files. During this process, inconsistencies in both data definitions and data values are identified. These inconsistencies must then be resolved by the relevant user departments.

- Structure of user department. The structure of a user department may no longer be effective following conversion; the user department may be designed around a particular application system. Restructuring application systems during conversion may precipitate user reorganization.
- New organizational roles. Conversion may cause new organizational roles to evolve in user departments. For example, to provide coordination between data base administration and the user departments, a "user data administrator" may evolve. The user data administrator serves as the focal point for participation and involvement on the part of the user department both during and subsequent to the conversion.
- Systems analysis. By providing such tools as a high-level query language and/or a generalized report writer, the DBMS enables nontechnical users to access the data base directly; users are less dependent upon programmers to satisfy simple requests for information. This increased availability of data may result in the migration of the systems analysis function from data processing to user departments.
- Personnel skill requirements. Conversion may impact the skill requirements of user personnel. For example, conversion to a data base environment may also result in operating certain application systems online, requiring that the user department acquire or develop terminal operations skills.

The initial conversion to a data base environment is not likely to be the only data base-related conversion that an enterprise will undergo. Rather, conversions of one form or another are likely to be a way of life. However, there are certain measures that the data processing organization can take to minimize the impact of future conversions, including institutionalization of the data base administration function, insulation of programs from a particular DBMS and DBMS-independent data base design.

A well-established DBA function will minimize the impact of future data base conversions. Specifically, the DBA function should take action as follows:

- Maintain data definitions and relationships in an up-to-date data dictionary.
- Document the structure and contents of all data bases independently of the data description language of the DBMS.
- Develop methodologies and standards for data base design which are independent of any particular DBMS.

The recent work of the ANSI/X3/SPARC Study Group on DBMS introduced the idea of a conceptual data model. The topic has also been pursued extensively in research papers. In practical terms it amounts to design and documentation of the data base in a form independent of any particular DBMS.

In translating the conceptual model to the data description language of the DBMS selected for implementation, the design decisions which are predicated on the characteristics of the DBMS are more clearly distinguishable from the natural structure of the data. Should the DBMS be changed subsequent to implementation, it is possible to focus more clearly on the structural conversions required of the data base and the applications accessing the data base. As a point of interest, use of the conceptual data model is appropriate whether a DBMS or conventional files are to be used.

Two sets of circumstances may motivate an organization to attempt to insulate its application programs from any one DBMS. On the one hand, an organization may be unwilling to commit completely to the use of a particular DBMS. Rather, it may desire to keep its options open with respect to converting to a different DBMS at a later date. Alternatively, a large multidivision corporation may desire to develop common application systems for the divisions; yet it may find that the data processing organizations within autonomous divisions have installed different DBMS.

It is possible to build an interface between application programs and the DBMS to isolate the programs from the DBMS. That is, the programs do not interact directly with the DBMS. Rather, standard program requests for DBMS services are translated into the required DML statements either at compilation time or at execution time. Thus, application programs are insulated from changes in the DBMS as long as an interface module can be developed to translate program requests into the DML statements of the new DBMS. Similarly, a single application will execute under any number of DBMS as long as the appropriate interface modules exist.

Furthermore, the multidivision corporation retains the flexibility of standardizing on a single DBMS at a future date. The negative aspects of this approach include reduced system efficiency and the possibility of ending up with a pseudo-DBMS whose capabilities represent the lowest common denominator of the various DBMS for which interfaces are built or planned. Nevertheless, a number of corporations worldwide have adopted or are adopting this approach.

As a data processing organization goes through the experiential learning necessary to assimilate data base technology, the functions and features of the DBMS package currently installed will tend to be more highly utilized. Users will have positive experiences with the facilities of-

fered by the DBMS and will subsequently place greater burdens on those facilities. Also, the technical capabilities of the DBMS will be increasingly utilized by the data processing staff to meet user requirements.

In short, the tendency to use the full functions of the DBMS over time will place a strain on the capabilities of the DBMS. This is manifested by either decreasing systems processing efficiency or increasing effort necessary to develop systems which meet user needs. These increased costs are recognized by both users and data processing personnel who then initiate a search for increased DBMS capabilities and, thus, begin a data base conversion effort.

This second type of data base conversion can be characterized by either a complete change in DBMS packages or an upgrade in the version of the DBMS currently installed.

As has been discussed, the most prevalent reason to undertake a conversion from one DBMS to another DBMS-1 to DBMS-2 conversion is to install a better DBMS. A better DBMS is usually defined as having improved, more complete functions, better performance, improved query capability, development of richer data structures, more efficient usage of the computer resource through decreased cycles and/or space, improved or added communication functions, availability of transaction processing and distributed processing capability.

Another major reason to undertake a DBMS-1 to DBMS-2 conversion is to standardize DBMS usage within the company. Many large corporations are finding that the DBMS selections made several years ago to meet specific application needs have resulted in the installation of several DBMS packages within the data processing organization. The impact of multi-DBMS usage in a single data processing environment is major.

For example, application programs are constrained to the design and processing characteristics unique to each DBMS. Data files are structured to be accessed by a single DBMS. And design and programming personnel develop the skills necessary to implement systems associated with a single DBMS.

The multi-DBMS environment results in a substantial investment in data processing personnel technical skills and reduces the potential for integrating applications that operate on different DBMS.

For these reasons, many companies are now developing standards for DBMS usage. Those standards are usually application systems to be developed under a single DBMS. Exceptions may exist where the application to be developed is stand-alone in nature with a low potential for integration with other systems.

The last major reason for DBMS-1 to DBMS-2 conversion is that such a conversion is dictated by a hardware change. Many of the commercially available DBMS are offered by large mainframe vendors. As such, a move from one hardware vendor to another will necessitate a change in DBMS usage. This can become quite a complex effort in that the source code and data base storage structures of all programs will require changes. If there is a history of hardware conversions in the company, the wise data processing manager should select a DBMS that is not hardware-dependent.

A key concept that was introduced in the previous section is that the data base conversion effort should be analyzed and managed like any other high-risk systems project. The same concept applies to a DBMS-1 to DBMS-2 conversion effort. As a result, the DBMS-1 to DBMS-2 conversion should be justified on the same basis as any other systems development effort is justified in the company.

An economic justification should be made on the basis of costs and benefits associated with the data base conversion. The economic justification is particularly important if the major reason for the data base conversion is either better DBMS or standardization of DBMS usage. For a hardware change, the cost and benefits associated with the DBMS conversion should be included in the justification for the hardware change.

The economic justification for a DBMS-1 to DBMS-2 conversion should be based on a succinct articulation of the costs and benefits directly associated with the conversion effort. Costs should be identified on an incremental basis and be classified into three categories: one-time conversion costs, incremental costs for each planned application to be converted to the new DBMS and on-going DBMS support costs.

Benefits should likewise be identified on an incremental basis within the same time frames as the associated costs. Benefits are divided into two categories: discernible/definable cost savings in development, maintenance and operations and intangible cost savings.

A more complete description of the types of economic considerations to be addressed is contained in *Data Base Directions: The Next Steps,* National Bureau of Standards Special Publication 451.

Above all, the justification for a data base conversion should be developed and communicated to management in the same manner that any other project is justified.

The impact of a DBMS-1 to DBMS-2 conversion effort can be felt on each of the four growth processes previously discussed. Many of the types of impacts are the same as those previously identified in a conversion to data base technology. Users and data processing management should recognize that many of the same experiential learning processes occur in subsequent data base conversions as they do in the initial effort.

The impact of subsequent data base conversions on application portfolios occurs in application programs,

data bases and cataloged modules.

Because application programs are buffered from actual data storage structures by the DBMS, the unique characteristics of each DBMS will have a major impact on application programs in the following areas: DBMS "call" structures, programs view of data and mappings (model, structure, content), application program logic and data communications.

Physical data storage structures and logical data relationships are implemented via unique DBMS utilities and are patterned after distinct DBMS requirements. As such, data bases developed under one DBMS are not readily accessible by other DBMS packages. Specifically, data bases are impacted by the vagaries of data base management systems in the following ways: data base definitions in both the DBMS and in the data dictionary, data content and storage format, use of data base design and simulation aids and conversion aids.

Although processed just as any other program, cataloged modules differ from application programs in their function and method of development. The specific types of cataloged modules which are impacted by a change in DBMS are cataloged queries, cataloged report definitions and cataloged transaction definitions.

As was previously discussed in the section on conversion to the data base environment, the major impact on the data processing organization structure is the implementation of the data base administration organization. Because this DBA structure has already been integrated into the data processing environment during the initial data base experience, the conversion from one DBMS to another will not have a major impact on it. Only procedural fine-tuning will be required as the functions of the DBMS change. However, it should be recognized that a substantial learning curve will likely exist as the new DBMS technology is assimilated by DBA personnel. Other organizational structure changes in the data processing environment as a result of the DBMS-1 to DBMS-2 conversion will be minimal.

The major organizational impact throughout both data processing and user areas is likely to be in the technical and functional education required before, during and after the conversion effort. Data processing and user personnel in all areas of systems development and operation will have to be trained on the new aspects of the DBMS. Training programs for all people should be identified and initiated in advance of the conversion implementation.

Another major area of impact on the data processing organization from a data base conversion is the modifications in documentation necessary to accommodate the new DBMS environment. Changes in documentation will occur in the following areas:

- DBMS functional and technical support (reference) documentation.
- Functional and technical descriptions of any application systems converted onto the new DBMS.
- Physical and logical descriptions of any data bases converted.
- User-oriented descriptions of application systems processing characteristics.
- System development methodology documentation that references particular aspects of data base or application development.

As previously discussed, data processing management control systems are those sets procedures regularly used to control both systems development and operations functions. The conversion from one DBMS to another is not going to modify the conceptual framework used to control the data processing environment. However, specific changes will affect the mechanics of control.

The chargeback algorithm used to charge users for data processing services is likely to change because of modifications in DBMS overhead (cycles), DBMS space requirements, methods of implementing logical relationships, ownership of data items, differences in efforts required to design and implement application systems, differences in methods used to structure ad hoc queries and periodic reports and methods of charging end-user cost centers for the one-time costs of conversion. The time-frame of allocating these charges can also be important (one lump sum versus periodic payments).

There will be changes in the time frame and types of effort required in systems development. Conceptual approach to developing systems may change due to the total effort or time frame required to generate sample reports on test data bases. Design procedures in the methodology are not likely to change if DBMS facilities are similar; only jargon should change in documentation.

Systems development and operations standards by which data processing personnel are regularly measured should change due to new functions and especially to a new learning curve. Computer operations performance monitors and standards will change due to new processing technology.

Adequate data base backup should be carefully analyzed and managed during the conversion process. Operational restart/recovery procedures will change due to new DBMS functions or utilities. Processing of data exceptions may differ.

Differences in methods of data access security should be recognized. Data manipulation restrictions may vary from DBMS-1 to DBMS-2.

Where appropriate, special care should be taken that all privacy disclosures are logged during a data base conversion per recent government regulations.

A keynote of data base conversions is that the conversion should be as transparent as possible to user areas. This axiom holds that the processing impact on user areas should be held to a minimum and that the necessary technical

capabilities to support the conversion should reside in the data processing area.

The impact of the data base conversion effort should be readily apparent to users regarding:

- Functional improvements such as a new query language.
- Increased data content ("While we are changing, let's add . . .").
- Increase in sharing of data will highlight data inconsistencies, validations and format errors.
- Possible planned disruption of services during conversion period.
- Data ownership changes.
- User mental images or expectations may change.
- Archival data capabilities may change (meeting the needs of IRS, EEO and so on).

The well-managed data processing installation should carefully articulate a data base conversion strategy and plan before initiating any conversion effort. Specifically, the data processing management personnel should determine specific conversion objectives, analyze pros and cons and develop a memo of rationale, develop a conversion strategy and develop a detailed plan for procedures, data and programs.

The following is a list of possible strategies which may be undertaken:

- Unbundle conversion of procedures, programs and data (conversion may affect all).
- Build bridge from old programs to new data structures.
- Convert nonvital program first.
- Migrate data and program conversion as maintenance threshold is approached.
- Consider bringing in outsiders to convert procedures.
- Run parallel for vital systems because frustrated users may be a problem, you will probably be turning up old bugs and parallel may be a security blanket only.
- Avoid parallel running where possible by careful planning and phased cutover or "fast" cutover with fall-back contingency plan, but note the possibility of high-risk exposure.
- Map how all users use the shared resources.
- Split input stream between old and new applications and plan validity checks carefully.
- Use DBMS facilities such as dumping/loading and mapping, if available.
- Convert data all at once, then convert programs as needed.
- Restructure data if necessary before any program conversion.
- Benchmark first.

Once the appropriate conversion strategy is determined, a detailed conversion plan should be developed. The following is a list of considerations to reference when developing a conversion plan:

- Develop a work plan as in any other systems development effort.
- Append original feasibility documents and review in the light of latest detailed specifications for conversion.
- Reevaluate "go/no go" at predefined checkpoints in the conversion process.
- Subject the conversion process to standard project-management control.
- Develop a contingency plan.
- Use planning document as an education tool for project personnel.
- Evaluate and select conversion aids.
- Flesh out impacts (see section on conversion activities and their impact above) into manageable tasks. Schedule these tasks and establish a reasonable work-breakdown structure.
- Go through planning document with an implementation hat on.
- Control project on at least a weekly basis.
- Plan for heavy user involvement in data translation to resolve inconsistencies and consolidate validation checks.
- Involve external and internal audit staffs.

In summary, the panel analyzed the impact of a DBMS conversion in terms of the four growth processes along which the DP function evolves, namely: the portfolio of computer applications, the DP organization and its technical capabilities, the DP planning and management control systems and the user

Two types of DBMS conversions were addressed: the initial conversion to a data base environment and the conversion from one DBMS to a second. Four key concepts summarize the findings of the panel:

- Conversion to a data base environment is primarily a question of how soon the data base environment should begin to be constructed, not whether a data base environment should be implemented
- Because the initial data base application represents an important learning experience for the entire organization, great care must be exercised in selecting the entry-level data base application
- The exposure to risk associated with a DBMS conversion is minimized by managing the conversion as any other large systems project
- An organization will likely be involved in multiple DBMS-related conversions, from the initial conversion of conventional applications to data base technology to the conversion of applications from one DBMS to a second DBMS. Accordingly, the DP organization should plan and structure for future DBMS conversions now to minimize the impact of future conversions

In closing, appreciate the technology involved, but recognize that a data base conversion is a management problem.