The System Planning Grid: A model for building integrated information systems

by B. R. Buckelew

Information systems have evolved as a result of technological advances and the increasing demand for information. Over the past few years, systems that developed separately are being forced to merge. This paper describes a model for building a set of integrated architectural guidelines to ensure that a "system" is being built. The use of the System Planning Grid as a model for setting product standards and organization responsibilities will also be discussed.

Over the past two decades the effective management of information and the technology to produce that information has become increasingly critical to corporate success. The rate of technological change and the growing demand for information technology has put a great strain on the information-handling systems that were conceived in the 1960s and 1970s. These systems, developed separately, are having to either merge or interface. The strain is compounded by three major technological factors. First is the merging of data processing, office automation, and communications. Second is the growth of offerings in minicomputers and microcomputers, each with entrenched advocates. Third is the lack of clear-cut industry standards to tie these systems together.

From work the author has done with several Information Systems (I/S) organizations to develop strategic I/S plans as a consultant and facilitator of structured planning sessions during the past three years, certain issues seem to recur as critical to I/S success. They are

- A strong link between the I/S plan and the corporate strategic plan
- A service-oriented business plan between I/S and its clients
- A professional and technically competent I/S organization
- A blueprint and a process to implement the different technologies into a usable, well-running, cost-effective, integrated system

In this paper we describe a model that has proven useful in building the blueprints. A company should be able to answer the questions "What are we building?" and "What are we going to use to build it?" There is a need to develop guidelines and standards and to provide the level of consistency required to meet the business needs. Mainframe computers, minicomputers, and microcomputers; hardware, software, and communications; office systems, personal computing, and production systems should all be included.

Consider the following example: A business professional acquires a personal computer and does some personal computing/spreadsheet work. Then he adds a word-processing package for correspondence. Afterward he wants to combine the two and produce a memo. Next he wants to send his correspondence around the department electronically. Then he requests access to corporate data bases, so that he will not have to rekey the data. Finally, he wants his secretary to correct his correspondence and send it to another department.

This example started innocently enough. However, each step required the integration of one function with another. The task of integration is difficult, and this difficulty is compounded by components that were never meant to work together. Without a system having a well-designed architecture, a user with a problem to solve and a modern tool available can get very frustrated, although initially the system was easy to use. Integration *within* each subsystem is no longer sufficient. The system should be analyzed as a whole. Conscious decisions should be made on the degree of integration versus local option with both a short- and long-term view.

The System Planning Grid is a model that is useful in building system guidelines.

The System Planning Grid (SPG) is a model that is useful in building system guidelines by visually integrating subsystems into a manageable whole. It was developed to assist IBM customers in defining the structure of their information systems in a way that was easy to understand. It has also proven to be useful in facilitating discussion among technical specialists as they integrate their specialties into a system.

Systems, subsystems, components, integration, and models

Using Greenwood's summary of systems concepts[1] as a basis, the working definitions for the above terms follow. *Systems* have three basic characteristics:

1. A system consists of subsystems or components.
2. These parts interact and are interdependent.
3. Through the interdependence, the otherwise separate parts acquire a degree of wholeness or unity.

The following definition is then possible: "A system is any set of elements which interrelate in some way to form a unified whole."[1]

Other definitions are as follows:

- A *subsystem* is an element of a system and a system in its own right.
- *Components*, in this discussion, are the building blocks of the subsystems and consist of hardware and software products.
- A *model* is a theoretical system that attempts to describe the empirical or "real" system that exists in the world. "The test of the value of any given model is not the degree of approach to the ideal, rather, the test is its effectiveness in serving the purpose for which it is being used."[1]
- *Integration* means to blend, orchestrate, harmonize, and unify. Integration can be achieved in different ways.
- *Integrated systems* are defined as those which were *designed* and *built* to work together. *Nonmodular*, integrated systems require the *entire* system, whereas *modular* systems allow the subsystems to be implemented on a piecemeal basis, when required.
- *Integrating systems* are those which *tie together* separately developed subsystems and make them *appear* as integrated. A dialogue manager is an example.

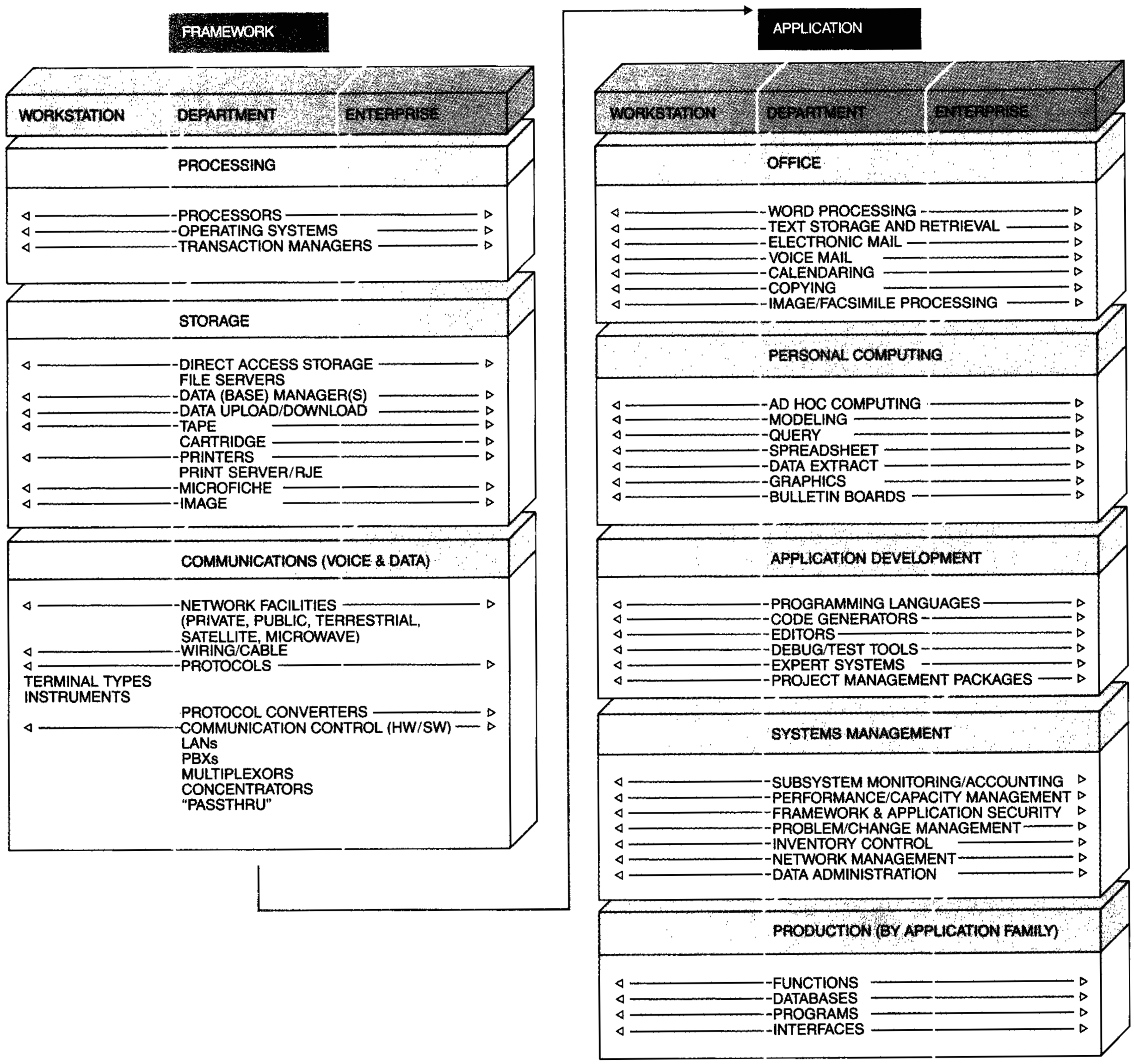
Structure of the model

The SPG model defines the *System* as including all information-handling hardware, software, and telecommunications used by a company. The System will interface with users, operators, other companies' systems, and the physical environment necessary to support it.

The model is divided into two major subsystems. (See Figure 1.) The Framework Subsystem is the "engine." It is responsible for processing, storing, and distributing information. The Application Subsystem is the entire application portfolio. The Application Subsystem rides on the Framework.

The Framework and Application grids are divided into three geographic columns or subsystems. These

Figure 1 The System Planning Grid model



divisions are delineated by the location of the respective subsystem, that is, *where* something is run.

In Figure 1, there are three columns. The *Workstation* column describes the subsystem on the desk (or plant floor workstation, teller window, location in the home, etc.). The *Department* describes one or more levels at the department (or office, plant, branch, distribution center, floor, etc.). The *Enterprise* describes the subsystem that is available to the entire enterprise. It includes corporate data centers.

Framework grid. The Framework is divided into three rows, or Framework subsystems. Each row has three geographic boxes, or components.

Processing is the row containing all of the hardware and systems software involved in processing. Included are processors, operating systems, and transaction managers.

Storage is all of the hardware and system software involved in the storage and retrieval of information.

It includes direct access storage devices (DASD), tape, cartridge, print, microfiche, and image hardware and the system software to support them, including data base software. It can be extended to include any other storage media and supporting software.

Communications is all of the communications hardware and software controlling the distribution of information among locations. It includes voice and data facilities, including private, value-added, public/switched, terrestrial, satellite, and microwave networks; wiring/cable; terminal and line protocols; host communication hardware and software; local area networks (LANs); private branch exchanges (PBXs); controller hardware and software including multiplexors, protocol converters, concentrators, and passthrough software; and communications interfaces to other systems.

The Framework subsystems and components should work together to process, store, and distribute information in an effective and efficient manner. The primary criterion in constructing the Framework, however, is its ability to meet the requirements placed on it by the Application Subsystem it supports.

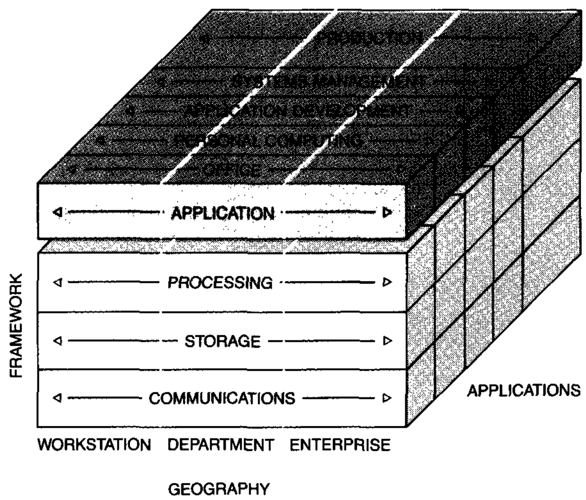
Application grid. The Application Subsystem includes all production applications plus four generic applications that are run by virtually all corporations: office, personal computing, application development, and systems management. Each application consists of the software to perform each application function, and, like the Framework, has three geographic boxes, or components.

The *office* application includes casual and heads-down word processing (memos, letters, and documents), text storage and retrieval, electronic mail, voice mail, calendaring, copying, and image (facsimile).

The *personal computing* (also known as end-user computing or decision support) application includes all *ad hoc* computing, modeling, query, spreadsheet, data extract, graphics, and bulletin boards, whether done on a personal computer or by a host-based Information Center (I/C).

The *application development* application is the subsystem used to *build* systems, including programming languages, code generators, editors, debug/test tools, expert systems, and project management packages.

Figure 2 Three-dimensional perspective of the model



The *systems management* application is the subsystem used to operate or manage the system, including processor, storage, and communications monitoring, security, capacity planning, problem determination, change management, inventory control, system generation, charge back/accounting, problem management, performance management, and data administration.

The *production* application includes all other applications required by a company or industry. They generally include families of applications such as financial, personnel, marketing, manufacturing, process control, order entry, distribution, inventory, engineering, etc. Each family includes functions, data bases, transactions, programs, and interfaces to other applications.

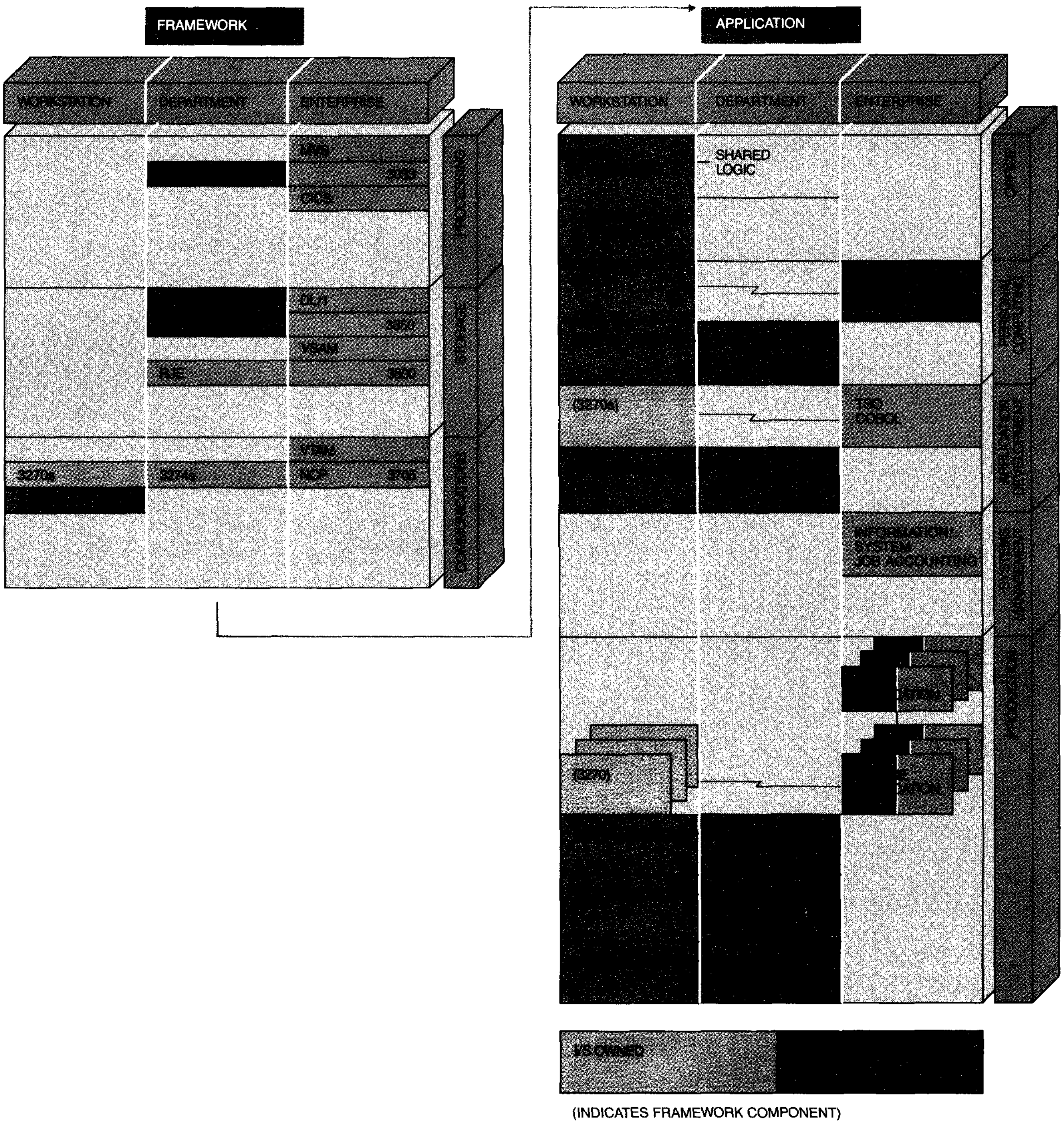
A more complete perspective of the model is shown in three-dimensional form in Figure 2.

Increased complexity

The need to integrate systems has resulted in considerably more complexity today than just a few years ago. Consider the example of the I/S system as viewed by corporate I/S in the early 1980s. It is shown in Figure 3. The same system, a few years later, often has products in every box and integration issues in every box, row, and column.

As shown in our example of the business professional, the functions of the subsystems are merging.

Figure 3 Example of system in early 1980s



Following are a few examples of changes occurring and issues that these changes have raised. These examples are organized using the grid of Figure 1.

The columns. *Where* systems reside has changed as a result of technology, price, and ease of use.

Workstations. Personal computers and terminals are merging to become network-attached workstations. As workstations grow in use they are performing functions in any row and require access to functions and data in the department and enterprise columns of the grid.

Departments. Departmental computers, often considered easy to use, have mostly been superseded by personal computers as the "easy-to-use" computer of choice. As the use of departmental computers has grown, they have taken on more and more characteristics of enterprise mainframe computers. Departmental subsystems provide departmental applications and are being required to act as the integrated interface between workstations and the computer centers of enterprises.

Enterprise. Enterprise, or computer center, growth continues. The growth in end-user (office and personal) computing has often made the virtual machine (VM) systems the fastest growing in the computer room. The effective interfacing of end-user VM systems and high-availability Multiple Virtual Storage (MVS) data base production systems is an increasingly important system requirement.

Overall, there is a growing requirement for software that can operate on the technologies in any of the columns. This is needed to allow functions to move across the columns as technology, organization, performance, or other conditions change.

The rows—Application Subsystems. Significant changes have also occurred in the Application Subsystems in the past few years.

Integrated *office* systems with enterprise, departmental, and workstation components are becoming more common, and the need to integrate the components is generally recognized. Standardization of word processing and integration of secretarial and professional text creation, revision, distribution, and integration with other information types (data, graphics, etc.) are increasing the integration requirements. Document Interchange Architecture and Document Content Architecture[2] provide the structures used as a basis for this subsystem.

The area of *personal computing* has exploded through the use of personal computers, time-sharing minicomputers, and Information Centers (I/CS). Although cooperative processing is emerging, the real work of integrating this subsystem remains to be done. Of particular interest and concern are spreadsheet and data base compatibility and the integration among enterprise, department, and workstation components, including data extraction from production systems. Of growing concern is the use of personal computing to build production applications that are not documented and have no architecture. These applications are often not capable of being maintained, expanded, or integrated with other departmental application subsystems, particularly when the "personal owner" moves on.

Application backlogs, the requirement for development productivity, and the growing need to restructure twenty-year-old production systems has led to a focus on *application development.* The I/C has become a base for end-user-developed applications. Host-based application development centers, at both the enterprise and department levels, are becoming more common.

The *systems management* subsystem has changed from one of applications managing a data center and terminal network to one capable of managing the corporate-wide flow of information, with many more components and interfaces.

The subsystems called *production systems* are under considerable stress. Old subsystems are being questioned on their ability to change as the corporation wants to change. Many of these systems have been equated to our steel mills of the 1950s and 1960s. Large capital investments are required to *rebuild* these systems from their foundations. Demands exist to integrate separately evolved production subsystems and provide data to the growing number of personal computers.

New major subsystems are being questioned on the cost, time, and capability of I/S to implement them. Commercially available application packages are being considered whenever feasible to reduce development and maintenance costs. However, integration is still required.

The rows—Framework Subsystems. The demands of the Application Subsystems and the capabilities provided by new technologies are putting the Framework Subsystems through similar stress.

Processors and their operating systems are cornerstones of the Framework. Being able to run application packages and being able to support the hardware and software for the other Framework Subsystems are primary *processing* considerations. In other words, once the computer is selected, choice of the remaining components is often dictated. The ability to manage and integrate multiple operating system environments (MVS, VM, minicomputers, and microcomputers) is required. Centralized versus decentralized operational control remains as an issue.

In the *storage* area, the problems of effectively distributing data are magnified since disk, tape, and printers are multiplying as rapidly at the workstation and department levels as at the enterprise level. Issues of data integrity, or "Which copy am I using?" are growing. The user demand for personal computer file and print servers at the department level is being integrated with enterprise-driven remote job entry (RJE) data extraction and backup/recovery requirements. Integrating data with text, image, graphics, and voice places further demands on this subsystem.

A decentralized corporation may require more information flow.

In *communications*, systems are growing and merging in many directions. Private and public, Systems Network Architecture (SNA) and American National Standard Code for Information Interchange (ASCII), voice, data, and image networks are converging. The technologies of local area networks (LANS), cabling systems, PBX, control units, and protocol converters play the key role, usually at the department level, in connecting components together. Bandwidth demand is continuing to increase rapidly, with high-performance enterprise systems and increased data downloading in demand.

Summary. We have just described a partial list of issues that affect the building of integrated systems. Books have been and no doubt will continue to be written on the issues in each subsystem and component. The technology abounds with techniques and products to integrate two, three, or four of the "boxes." The subsystems, however, seem to be chained together. Future success apparently requires the integration, loosely or tightly, of every row and column. I/S success requires architectural guidelines that allow integrated systems to be built efficiently, without stifling the ability to provide users with functions that solve their problems.

Building product guidelines

The first determining factor in building guidelines is likely the degree to which corporate processes, personnel policy, use of capital assets, and decision making are centralized or decentralized. The second factor is the degree to which I/S corresponds to the corporate structure. The third factor is the degree to which information flows across the corporation and to corporate headquarters.

A corporation that is currently decentralized may, in the future, require more information flow and hence a more integrated set of subsystems. In a decentralized corporation, a series of individual company or department guidelines, using the SPG, would be useful in understanding what merging or interfacing is possible and where the possibility of merging is desirable. Incompatible product guidelines will generally preclude merging of subsystems.

System requirements: user and I/S view. While it is the purpose of the overall system to respond to changing corporate requirements, *each* subsystem (row, column, and box of the SPG) should meet the following specific requirements:

1. Each subsystem should meet its individual functional requirements.
2. Each subsystem should meet its individual performance requirements.
3. The individual subsystems should work with other subsystems.
4. Department and enterprise subsystems should be accessible through a user's workstation.
5. The appearance of the subsystems to the user should be consistent and easy to use.
6. The system should be configured in an *overall* cost-effective manner.
7. The subsystems should be changeable and expandable without major disruption.
8. The system and subsystems should be manageable.

Users and I/S will generally prioritize these eight system requirements differently. The user has a problem to solve and views technology as a tool to help solve it by *automating* certain functions. I/S sees a system that must be managed over the long term and understands that *integration* is important. The issue is seen from opposite sides of the SPG (as depicted in Figure 4). I/S must ensure, through the architectural guidelines, that today's application solution does not become tomorrow's system problem. More important, the technology must be available to the user and easy enough to use to solve the user's problems. Finally, there needs to be a balance between the pressures from both sides.

Overall recommendation. A rule in establishing architectural guidelines is to *simplify* whenever possible. This should not be taken to the extreme of being so restrictive that the system is functionally weak or difficult to use or to change. But products should be considered for deletion as well as addition. An SPG "cube" with 10 application families and one different product per box has a potential of $14 \times 3 = 42$ application boxes, with a potential of $[n(n-1)]/2 = (42 \times 41)/2 = 861$ *potential* application interfaces.

It can also have $14 \times 3 \times 3 = 126$ Framework boxes with $(126 \times 125)/2 = 7875$ *potential* Framework interfaces. Yet a system with a hypothetical single, standard Framework product per box for all applications would have $3 \times 3 = 9$ Framework boxes with $(9 \times 10)/2 = 45$ potential system interfaces.

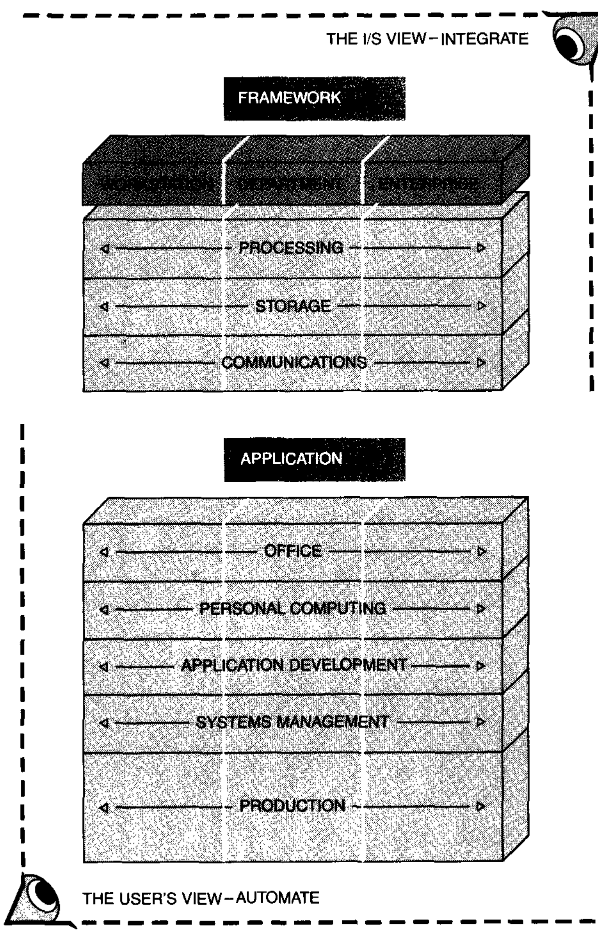
On one hand, although it would be ideal if everyone had a single (universal) workstation which was attached to a single (universal) department system providing all departmental functions, which in turn was attached to a single enterprise-level mainframe computer with *one* set of systems software, it is, of course, not likely. On the other hand, an approach of "pick anything from column A, B, and C and try to hook them together" is by far the most costly and is apt to be doomed to failure.

Methodology for creating guidelines. Using the SPG as a model, the following steps can be taken to create a set of architectural guidelines.

1. Document current inventory. Fill in the rows, columns, and boxes of the Application and Framework grids with *all* currently installed products. Remember, products are placed in columns based on *where* they reside. Component/product *ownership* is also important and should be noted (or color-coded). Of particular interest is where *department-owned* Application Subsystems reside on either the enterprise or workstation subsystems or where *corporate-owned* applications run on department or workstation subsystems. Document current problems with this system. Document foreseeable problems with this system if present trends continue. These problems should be noted overall and for each column, row, and box.

2. Build architectural guidelines. These guidelines include (a) the functional requirements, (b) the architectural strategy to meet these requirements, and (c) any high-level product strategies.

Figure 4 Two views of the System Planning Grid



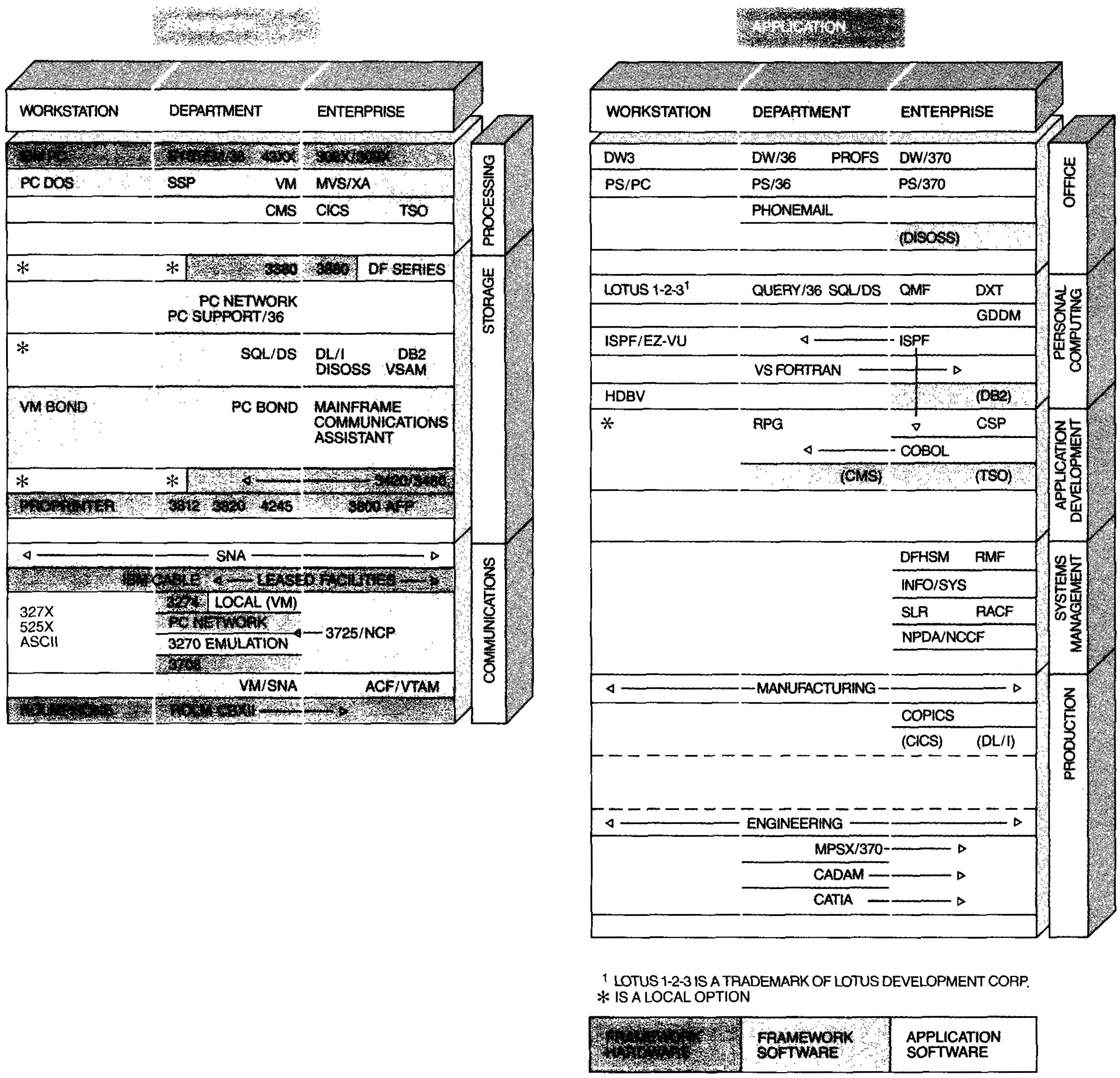
The order in which the guidelines are specified will vary. A recommendation is that the following order be used:

1. Overall (tied to corporate and I/S goals)
2. Columns
3. Application Subsystems by row (or groups of rows)
4. Framework Subsystems (by row)

This step is an iterative process that requires agreement and consistency of rows and columns. The previously documented eight general system requirements and the row and column issues should be useful in this step.

3. Specify product standards. Fill each SPG box with products that meet *both* the column *and* row

Figure 5 Product standards



architectural guidelines. Specify which products are (a) standard and centrally supported, (b) recommended and locally supported, (c) local options and guidelines (if any), and (d) too soon to decide; in question. See Figure 5 for an example.

4. Compare the SPG created in Step 3 with Step 1. Create projects and assign responsibilities to: (a) install any new standard products, (b) remove nonstandard (not recommended) products, and (c) establish research and development projects to determine what should go in the boxes with questions.

5. Establish a process to ensure that new subsystems will integrate with the system. This is particularly important for new applications.

6. Ensure that each row, column, and box has an owner for determining future requirements and

recommendations. See Figure 6 for an example.

7. Establish a process for the *owners* to convene on a regular basis to restate and maintain the guidelines. This step is essential to ensure that boxes, rows, and columns do not become isolated and cease to function as part of a whole.

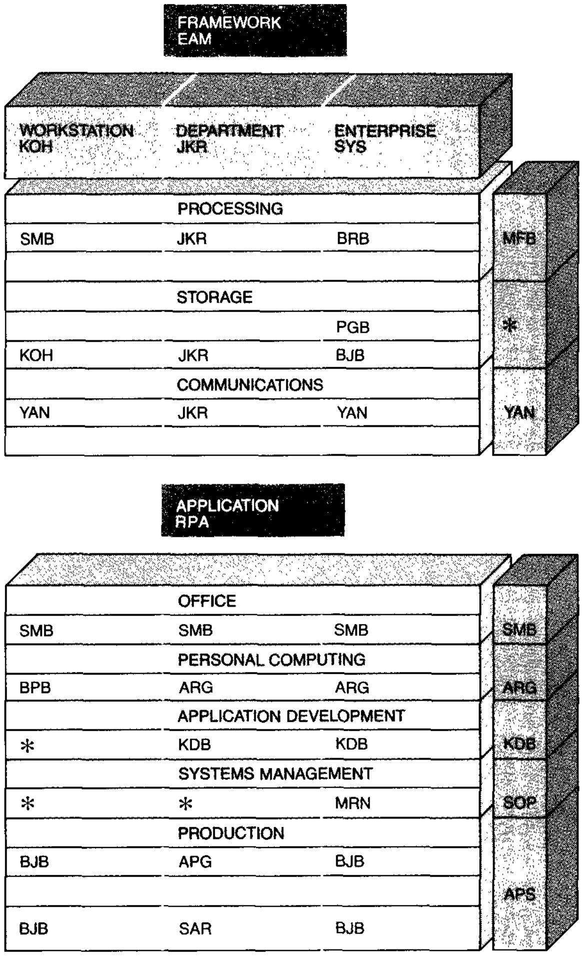
Viewpoints on the *importance* of various rows, columns, and boxes will vary widely and will depend on an individual's frame of reference. For example, someone involved with data base software will consider enterprise storage and its relationship to enterprise production as the driving force. A communications manager might view departmental communications as the most critical. A "user store" manager would consider personal computers for office and spreadsheet work and the ability to download data as most important. In addition to building the guidelines, communication among these varying viewpoints and technologies is an important product of this process. It is also apparent that all eight system requirements cannot always be met. Trade-offs will always be required. Currently installed inventory, current availability of products, and future directions in product development of major hardware and software vendors will be important in making these trade-offs.

After doing Steps 1 through 6 using the two-dimensional model of Figure 1, a more rigorous analysis can be completed using the three-dimensional model of Figure 2 if desired.

For corporations with distinctly different systems technologies and decentralized control, a *separate* SPG can be created for each. The *set* of SPGs would then comprise the corporate architectural guidelines.

Finally, a subsystem or component impact/process analysis can be done, using the grids, to analyze service delivery exposures and processes to deal with them. The following questions can be useful in that analysis. What is the service level requirement in each box? What is the service level across the columns to the user? Who is responsible for delivery of the service? How is it measured? What is the impact when a component or subsystem fails? What are the backup/recovery procedures for short- or long-term (disaster) outages? Who is responsible? What is the performance requirement in each box? Is there a capacity plan to ensure that performance is maintained? Who is responsible for tuning? What are the security requirements? Is security dealt with as a set of Framework issues, or is each Application subsystem responsible for security? How is it monitored? How are problems and changes managed within each box? How are they coordinated across boxes? Again, who is responsible?

Figure 6 **Ownership in SPG. (Initials indicate group or individual responsible for requirements and product specification. Two sets appear in a box when it is "owned" by multiple people. Initials on a row or column indicate row or column ownership.)**



* INDICATES LOCAL OPTION OR NO OWNER

Example

This partial example is for illustrative purposes only and is meant to be neither complete nor a recommendation.

Overall requirements. The following items apply overall and to each subsystem:

1. The subsystems will meet the functional and performance requirements of the intended users.
2. Overall system simplification is a fundamental requirement. Subsystems should work together to form a consistent whole system whenever possible.
3. The system should be manageable and changeable to support new function and growth.
4. The *overall* system should be cost-effective.

Overall architectural strategy

1. Few standard technologies and products will be used throughout the organization and fully supported.
2. Subsystems will be installed that are *capable of integration* with other subsystems. Each row and column will have a documented integration strategy, taking into account currently installed subsystems. Actual integration will occur when functionally required and economically justifiable.
3. Procedures for implementing *exceptions* will be clearly documented.
4. Standards for new technologies will be built on the current standard base through funded research and development projects.

Overall column strategy

1. New *production* applications should be built on enterprise subsystems as part of an overall application plan whenever possible, using standard data base and application generators.
2. Department subsystems should be used when performance (subsecond response) or application packages make this the reasonable choice over enterprise or workstation processors. Department subsystems should also "attach" their workstations through the communication subsystem to the enterprise.
3. A workstation selection process will ensure that a user has the best workstation for his/her primary job and access to department and enterprise subsystems.

Enterprise requirements/strategy

Framework

1. Enterprise subsystems must have virtually unlimited growth capability (50 percent per year).
2. Enterprise capability will be located in two geographic locations.
3. Enterprise processing will run on as few system images as possible.
4. Enterprise processing will be MVS-based.
5. Enterprise storage subsystems will be DL/I- (production), DB2- (personal computing), and DISOSS- (office) based.
6. The communications subsystem will be SNA-based and enterprise-controlled.

Application

1. Enterprise applications will conform to a long-term application plan.
2. Enterprise applications will be built with standard application generator products and use standard data bases.
3. Report writing/query subsystem should use data from number 2 and provide maximum flexibility to departments and users in file downloading and manipulation of data by the enterprise.

Department requirements/strategy

Framework. Department processing subsystems will interface with workstations and enterprise systems to meet *department*

1. Printing requirements (print server and RJE)
2. Direct access storage requirements (file server and spool)
3. Workstation pass-through to enterprise requirements
4. Access security

Application. Department processing systems will be used to meet

1. Specific application requirements where packages exist
2. Performance requirements when the workstation/personal computer is not sufficient
3. Department processing subsystems will be System/36-based or 4300/VM-based.

Workstation requirements/strategy. The same steps are followed in specifying the strategy for the Framework and the Application.

Application Subsystems—office requirements

1. An office subsystem is required that allows effective document interchange throughout the corporation.

2. A standard word-processing application is required throughout the corporation to allow the movement of personnel among departments.
3. Messaging systems, including voice and text, will provide an easy-to-use interface.
4. A document storage and retrieval mechanism will reside at the enterprise level.
5. Personal word-processing and messaging systems outside the above standards will not be attached to the corporate system.

Office architectural/product strategy

1. The corporate-wide system will be built on Document Interchange Architecture (DIA) and Document Content Architecture using the Distributed Office Support System (DISOSS) as the data base manager.
2. DisplayWrite will be the standard word-processing package for all departments. Use of other products which meet the DIA and Document Content Architecture will be considered.
3. PS/370 and PS/36 will be used for enterprise and department text message systems. Rolm Voice Mail will be used for voice messaging.

Application Subsystems—personal computing requirements

The same steps are followed for the remaining application subsystems.

Product standards. Figure 5 offers an example of product standards.

Concluding remarks

There seems to be a need for a model and process to deal with the issues of integrating information systems. This is particularly true in today's environment, where individual products have evolved in different ways and the need to integrate unlike products is great. The I/S department, user departments, and the corporation as a whole are the potential beneficiaries. The System Planning Grid was developed and has been used to assist IBM customers with this process. It has been particularly useful in identifying a specific set of integration issues and providing a model and forum for resolving those issues. It has also been a useful model in identifying overlaps in organization responsibilities and assisting in building plans to resolve the conflict inherent in the overlap. Much work remains in improving the analysis techniques required to build a useful integrated product strategy.

The use of the SPG is complementary with other proven approaches to divide, analyze, and build I/S systems. This is particularly true of the following methodologies.

- Business Systems Planning (BSP), which is effective in translating the requirements of a business into integrated, prioritized, application, and data systems plans.
- The Management of End User Computing, a process for satisfying user needs, is particularly effective in building workstation and personal computing plans including justification and support structures.
- A Management System for the Information Business is useful in planning the activities of the I/S organization as a business.

Acknowledgment

I would like to thank Marilyn Rex, Senior Systems Engineer in the IBM San Francisco Downtown branch office, for her partnership in developing the original model. Also to be thanked are the Cross-Branch Unit, account teams, and customers in the National Accounts Division San Francisco Downtown branch office who have worked with this tool and provided criticism to make the model more useful.

Cited references

1. J. W. Greenwood III and J. W. Greenwood, Jr., *Managing Executive Stress: A Systems Approach*, John Wiley & Sons, Inc., New York (1979), pp. 7–27.
2. T. Schick and R. F. Brockish, "The Document Interchange Architecture: A member of a family of architectures in the SNA environment," *IBM Systems Journal* **21**, No. 2, 220–244 (1982).

General references

W. E. Beregi, "Architecture prototyping in the software engineering environment," *IBM Systems Journal* **23**, No. 1, 4–18 (1984).

B. R. Blackmarr, "5th annual guide to word processing: When WP meets DP," *Management World* **14**, No. 4, 16–17 (April 1985).

Business Systems Planning—Information Systems Planning Guide, GE20-0527, Fourth Edition, IBM Corporation (July 1984); available through IBM branch offices.

D. Didising, "A danger in the stampede to integrate: Data misuse," *Office* **101**, No. 1, 96 (January 1985).

B. C. Goldstein, A. R. Heller, F. H. Moss, and I. Wladawsky-Berger, "Directions in cooperative processing between workstations and hosts," *IBM Systems Journal* **23**, No. 3, 236–244 (1984).

G. E. Greenley, "The integration of product decisions," *Management Decision* (UK) **22**, No. 2, 36–44 (1984).

P. G. W. Keen and L. A. Woodman, "What to do with all those micros," *Harvard Business Review* **62**, No. 5, 142–150 (September/October 1984).

A. MacRae, "PBX technologies," (Office Automation), *Computerworld* **18**, No. 15a, 49–51 (April 11, 1984).

The Management of End User Computing: Executive Direction, G320-0735, First Edition, IBM Corporation (September 1984). Methodology includes sixteen manuals, videodisc, and brochures; available through IBM branch offices.

A Management System for the Information Business, GE20-0662 (six volumes), IBM Corporation (July 1981); available through IBM branch offices.

F. W. McFarlan and J. L. McKenney, *Corporate Information Systems Management: The Issues Facing Senior Executives*, Richard D. Irwin, Inc., Homewood, IL (1983). *The Development Center Project* (MP-1432), The Development Center, Guide International Corporation (February 1, 1985).

S. Pfrenzinger, "Too many tools spoil center," *Computerworld* **19**, No. 17, 39, 44 (April 29, 1985).

B. Popek and H. Jacobsohn, "Managing information as a resource," *Data Management* **21**, No. 9, 32–34 (September 1983).

S. G. Rae, "DP managers reach out to micros," *ICP Data Processing Management* **10**, No. 1, 14–18 (Spring 1985).

C. M. Robins, "Managing the move to distributed processing," Special Report, *Computerworld* **17**, No. 13, 37–38 (March 28, 1983).

W. Roelandts, "Trudging through the interconnect maze: A field guide," *Data Communications* **14**, No. 5, 227–237 (May 1985).

"Technology restructures office planning process," *ComputerData* (Canada) **9**, No. 4, 8 (April 1984).

J. A. Zachman, "Business Systems Planning and Business Information Control Study: A comparison," *IBM Systems Journal* **21**, No. 1, 31–52 (1982).

Bruce R. Buckelew *IBM National Accounts Division, San Francisco Downtown Branch Office, 425 Market Street, San Francisco, California 94105.* Mr. Buckelew is currently a Consulting Systems Engineer and data processing professional specialist in the San Francisco Downtown branch office. He joined IBM as a systems engineer in 1966. He has also served as a regional systems management specialist and systems engineering manager. In his current capacity he has worked as a consultant and planning session leader/facilitator with many San Francisco-based accounts. In the past three years, he has also developed material and run staff and management development programs on the subjects of planning and quality. Mr. Buckelew received a B.S. degree in industrial engineering from the University of California, Berkeley.

Reprint Order No. G321-5255.