MOVING A SAS/AF APPLICATIONS SYSTEM FROM THE MAINFRAME TO THE PC

W. D. Hutchison, Portland Water Bureau
Geoffrey Wheatley, Intel Corporation

Abstract
This paper discusses integration of PCs and workstations into a SAS mainframe based computing environment. The characteristics, components, and users of a comprehensive SAS-based computing environment are described. The appropriate roles for multiuser and single user computers are explored, along with issues of portability, conversions and information systems management.

Introduction
The recent availability of SAS for PCs and workstations linked to the corporate mainframe has brought new flexibility and increased complexity to the overall implementation of the SAS environment. It has become necessary to match applications to the suitable environment in order to optimize performance and control costs. In order to understand the division of the computing load between multi and single user systems, we examine the components of a comprehensive SAS-based computing environment along with the classes of users requiring these components. The analysis defines an integrated approach that can be used as a guide in the planning process.

The SAS Computing Environment
With SAS/AF, SAS/FSF, and the macro language an environment can be built to support a broad class of computing functions and users. As the environment becomes more comprehensive, gains in productivity are realized by both the users and the programmers. Users interact with a standard screen interface and can apply skills learned in one application to a broad collection of enterprise-wide data and analysis tools. Programmers and advanced users can take advantage of a growing collection of tools and building blocks as well as the ability to plug their applications into an existing robust environment. The computing environment described is based on experience in industry, but remains an ideal. Because SAS/AF is the primary user interface, SAS is invoked immediately at logon without exposing the user to unfamiliar and unnecessary operating system shells such as ISPF. SAS’s autoexec feature is used to place the user directly into the SAS Primary Menu. During logon a standard collection of tools and the user’s personal files are made available. File allocations and other requirements for additional applications are set up when selections are made from the central menus. Each user has an individual menu system which can be modified to include direct access to often used applications as well as personal data and programs. If needed, many well-behaved non-SAS facilities (including ISPF, TSO, or REXX commands) can be selected from a SAS/AF menu, using the X statement. Upon exiting the called facility, the user is returned to SAS with macro variables and work files intact. Standards are maintained throughout the environment regarding such items as key definitions and screen layout. It is assumed that the user has no prior knowledge except familiarity with the interface. The on-line help documents not only how to use the applications, but the meaning of data elements, policies and procedures controlling business practices connected with the application, and the interpretation of reports and analyses. Users are encouraged to spend time exploring available data and tools. CBT courses on using the software are purchased from the SAS Institute and made available from the primary menu. The Institute’s policy of barring access to the source code has made this technique increasingly difficult to implement as it is assumed the courses will be run as independent SAS jobs. The inability to customize the training to local techniques and conventions also limits this otherwise excellent product.

Rather than systems requirements definitions, or even prototyping, the environment is based on a model of continuous improvement. Small enhancements are constantly made to any system as needed. In the spirit of the SAS Ballot, a suggestion box is included on every system menu to solicit feedback and suggestions. A menu selection for Proc Build is included to allow viewing or modification of the underlying code. Because the shared environment is accessed read-only and must be available during working hours, a mirror image test system is necessary. Modifications and additions are made to the test system which can be transparently allocated to end-users for review. A utility is provided to move approved code to the production system on nightly batch runs. In addition, tools and systems developed in an end-user’s local work area are often moved to the central library after review.
System Components

GENERIC TOOLS. Using SAS/AF, a shell can be written around SAS procedures for ad hoc reporting, data manipulation, and code generation providing a "generic" tool. The user is not required to write any SAS code, but can still become familiar with the functions of various procedures and options. SAS/ASSIST is an excellent illustration of this approach and includes the ability to save the resulting SAS program. Unfortunately, the number of options and statements supported for most procedures currently limits this product to being essentially a marketing gimmick. The SAS Report Writer prototype also illustrates this technique albeit with surprisingly inefficient coding.

SYSTEM EXTENSIONS. In order to provide portability, enhanced functionality, and to maintain the standard interface, system extensions written in SAS form a shell around most common operating system calls. Changes to peripheral hardware or new operating system features can be accommodated through changes to a small collection of shared programs. An example of a system extension is Menuprt, an in-house autocall macro, which allows optional viewing or editing of output prior to routing. A menu allows destination selection (printers, file, PC download, etc.) along with printing parameters (number of copies, orientation, forms, etc.). While implementation may vary between operating systems, the macro calls, user interface, and essential functionality should remain the same. Other examples include %Plot, similar to %Print but for graphics and based on Proc Greplay. To manage a user's programs which do not reside in a SAS/AF catalog and are normally submitted with Display Manager, there is a utility resembling a Proc Build directory. Activities such as sending messages, and scheduling and submitting batch jobs also benefit from being written in SAS.

PROGRAMMING TOOLS. In this category are utilities that improve productivity, reduce memorization requirements, and prevent mishaps by automating common programming tasks. An example within the SAS/FOBASE product would be interactive Proc Datasets. Using SAS/AF and SAS/TSP an enhanced interactive dataset restructuring facility was written to allow modification of type, length, and position of variables. Other facilities for programmers are a utility for the editing and management of user written formats, the SAS Usage Notes, Fsedit screens to collect SAS Ballot suggestions (if you wait a year you'll forget), and a directory of users. The user directory also forms the basis of a security system which does not require passwords.

Users and their tools

END USERS. Many users in management, in the field, or on the production floor have no desire to learn programming and their interactions with the computer can be limited to making menu selections and filling in selection fields or data entry screens. They rely on applications programmers and super users to provide menu-driven programs.

DATA ENTRY. Beginning data entry specialists need only know keyboard skills and Proc Fsedit user commands. However, with additional training they can improve their productivity and deal with a wider range of issues by creating and modifying Fsedit screens and using the programming tools relating to dataset structures and user-defined formats. This reduces the impact on other programming staff and allows creativity and control.

SUPER USERS. Most organizations contain a large contingent of administrative clerks, engineering technicians, planners, schedulers, and secretaries who provide the 'human glue' which binds the various operations and departments into an information network. A large percentage of their computer applications can be handled with simple Fsedit screens and Proc Print. These data collection, tracking, and reporting systems can be set up and maintained by users themselves once they have gained some basic programming skills. Training of these users begins with the use of generic tools for ad hoc reports and can be extended through CBT and classes.

DATA ANALYSTS. Some applications require extensive processing and a thorough understanding of the content of the data. Specialties in statistics, economics, logistics etc., usually having only limited training in data processing, make use of many advanced mathematical and graphical procedures which tend to be submitted repeatedly, normally from Display Manager, in developing their analyses.

APPLICATIONS PROGRAMMERS. Skilled in data processing, this group provides turnkey systems to end users by working with the entire range of available tools. The user is a partner in creating the application and shares responsibility for success. Super users often participate in portions of the coding. Assigning programmers to functional areas increases their familiarity with end applications and reduces conflicts over priorities and resource scheduling.
SAS SYSTEM PROGRAMMERS. Part of the programming staff must be dedicated to maintaining and enhancing the SAS environment. Along with creating utilities and interfacing hardware, this group uses SAS for computer performance evaluation and storage maintenance.

Multiuser vs Single User Environment

No doubt there are many approaches to fitting an application to the appropriate environment and hardware. One factor which could be examined is the concept of local vs shared data. Based on our experience in a variety of enterprises, data which is never needed outside its group of origin is extremely rare. As the SAS environment becomes more integrated, this fact becomes increasingly evident, and therefore a discussion of local data is outside the scope of this paper.

It is possible to posit an intermediate category between that of multi or single user based on distributed topologies using mini computers or a micro-based file server. However, ideally these topologies should present a single system image such that they appear indistinguishable from the traditional multiuser environment from the standpoint of the user/programmer.

Our approach will be to review the breakdown of users and system components developed above with an assessment of the appropriate platform for each.

Users

END USERS. This group uses the computer primarily to post or extract information on the status of operations and processes. The data is of interest to a number of people, and the program being run was written by someone else. Thus this group is almost exclusively in the multiuser environment.

DATA ENTRY. Often data entry personnel are involved in the conversion of paper files collected in the past to new computerised systems. As there is little need for the data to be available globally until data entry is complete, the PC becomes a viable option. For these and many other large volume applications, the normally faster and more predictable console response aids productivity. Optional input devices, such as voice recognition are being developed. For collection of real-time data via a machine interface, a dedicated processor is often a requirement. However, using existing file transfer techniques the savings in mainframe resources by off-loading data entry can be disappointing.

SUPER USERS. This group is the backbone of the corporate communications network. Starting with small systems for their local group, they soon find others in the organization who need or can supply parts of the data. Many such systems eventually wind up as major, centrally supported applications. Inappropriate use of PC's by this group can seriously inhibit productivity and should be discouraged.

DATA ANALYSTS. This group is often perceived as 'resource hogs' by the mainframe community, usually because of the memory and CPU requirements of their analyses. Because the data tends to be fairly static historical or test files, shifting to a PC or workstation does not incur a high cost in communications. Powerful processors and thrilling graphic displays are becoming increasingly affordable. Because of the unique nature of most analyses, sharing of tools and central programming resources is not necessary. Jobs exceeding the capacity of the workstation may need to be submitted to the mainframe in batch.

APPLICATIONS PROGRAMMERS. In a shop based on the continuous improvement model, the bulk of the applications programmers time is spent adding enhancements (and fixes) to existing applications. Because the programs are highly interactive, programming problems often revolve around the existence of files at a given time, shared access to files, unpredictable user selections, etc. This limits the opportunities to develop "off-line" on a PC. However, some well-defined but difficult stretches of code can benefit from being downloaded to a more responsive machine.

SAS SYSTEMS PROGRAMMERS. Utilities have a better chance than applications of being isolated, but often cannot be downloaded as they relate directly to hardware or the operating system.

System Components

GENERIC TOOLS. These are typically used for exploratory analysis on a central database or by super users generating ad hoc reports from their small tracking systems. Thus these tools are most useful in the multiuser environment. As the ease and efficiency of file transfer increases, and the analysis provided by generic tools becomes more complex and processor intensive through the addition of artificial intelligence, the use of generic tools in the single user environment may eventually become more common.
SYSTEM EXTENSIONS. The majority of the system extensions should appear identically in both environments although in the single user environment one normally need not select among output devices.

PROGRAMMING TOOLS. These tools are most often used by applications programmers and super users on the mainframe, but can also benefit data entry or even data analysts in their single user environments.

Converting to Version 6

In a traditional shop imbued with the COBOL programmer's desire to freeze code for eternity, conversions are considered a costly necessity to be avoided as long as possible, and then minimized. In the dynamic environment described above, they can be viewed as an excuse to halt the constant clamor for new applications, and instill some order into the existing code. A considerable period of time should be reserved to review all applications to make use of the latest utilities, conventions, and new software features.

Because version 6 of the SAS System is already available on PC's and many Unix workstations, a start can be made in learning the new features and beginning conversion now. Programming tools and system extensions which assist data entry and data analysts should be ported or created anew. Improvements to data entry can also be made using the new features of SAS/FSP.

Once the new version is available on the mainframe, the SAS system programmers need to begin work on the system extensions, programming tools, and generic tools. Because SAS/SHARE will not be available in the initial release, conversion of applications may need to be performed selectively, with both versions in use for a considerable time. When the applications programmers have gained experience converting the core programs, time can be scheduled to work with each super user. By combining consultation and assistance on conversion with applications review and programming training, you and your users can head into the 90's on a solid foundation.

Conclusion

The mainframe's reputation for unfriendliness is undeserved given the availability of revolutionary 4GL software such as SAS/AF, while current trends towards LANs and 'groupware' essentially continue the 'islands of automation' problem. By examining how computing tools are used the best use can be made of the new microprocessor technologies.

SAS, SAS/AF, SAS/FSP, and SAS/SHARE are registered trademarks of SAS Institute Inc., Cary, NC, USA.