Introduction

In every industry there are computer applications that are critical for success. In the pharmaceutical industry, one such “critical success application” is known as a Computer Assisted New Drug Application (CANDA). This is a tool designed to aid federal regulatory agencies in reviewing the effects of a drug before it is released to the open market. As with any key system, performance and ease of use are important design considerations. Merck has created several CANDAs in the past six years, the most recent of which involved migrating our latest CANDA from an Ultrix platform to a PC running Microsoft (MS) Windows.

Our new application is a large system based on MS Word and the SAS 6.08 for Windows (Base, SCL, SAS/AF, SAS/FSP, SAS/GRAPH, and SAS/STAT). The system consists of approximately 400 catalog entries. A major advantage of this migration was the opportunity to use FRAME technology (included in SAS 6.08 for Windows). This product aids in the creation of a more “user friendly” Graphic User Interface (GUI). FRAME is SAS Institute’s implementation of an object-oriented development tool. It allows the creation of screens with built-in mouse driven functionality and native Windows graphical objects such as push buttons, scroll bars, and list boxes.

We learned a great deal about PC configurations, MS Windows, and SAS 6.08 for Windows as we accomplished this migration. Our first purpose in this paper is to review the entire migration project, briefly describing the many steps involved from initial platform evaluation through final application development on the selected platform. Our intent here is that you gain a basic understanding of the interactive system, and that the information we present can be used as a guideline to plan other platform migrations of a large interactive SAS application.

Our second purpose is to present in more detail the platform evaluation and selection process. Performance evaluation is always an important component of a project that migrates an application across platforms, but the effort required on our part to complete it was particularly large because of our attempt to push the PC/Windows platform to its limits. Hence, we hope that by describing our process and findings in more detail, we can provide you with useful data as you consider this new platform for your own SAS applications.

Background - Existing Applications

Drug information is usually evaluated by multiple reviewers within each regulatory agency. These reviewers represent different disciplines such as medicine, statistics, pharmacology, toxicology, and chemistry. At Merck, CANDAs are modified for different types of reviews, but have many common capabilities including a graphical user interface, database querying, graphing, statistical analysis, document review features, file management utilities, and on-line help. In addition to the migration, one of our objectives was to combine our existing CANDA features with those represented in another CANDA system. The best features of both systems remained and were placed on a common computing platform. We joined a statistical based CANDA, developed with SAS 6.07 on a RISC-based Ultrix system, and a medical based PC system that used Microsoft tools for the document handling. Our goal was to create a system with three primary features: a user friendly front-end, a document review facility that depicts a visual duplicate of hard copy documents, and capabilities for querying data, creating graphics, and running statistical analyses in a timely manner.

The ultimate user of CANDA systems in the United States is the Food and Drug Administration (FDA). FDA requirements for CANDA features and response times are of primary concern. Discussions with FDA reviewers about performance yielded the following response time specifications: statistical analyses should take no longer than 30 minutes to complete and simple queries should average no longer than 5 minutes to complete. Further, we determined that this CANDA system would contain large amounts of data and must support some robust statistical analysis programs. This emphasized the critical need for quick system response time.

Initial Configuration Evaluation

An extensive design evaluation was performed to establish the appropriate hardware and software configuration for this new CANDA. The following six networked client/server designs and stand alone models were examined. (See Figure 1 for diagrams of the various configurations.)

- **X-station PC/RISC** - A PC with X-station software LANed to a RISC workstation with the SAS system, data processing, and text processing software.
- **PC Client/RISC Server** - A PC with the SAS system and text display software LANed to a RISC workstation with the SAS system, data processing software and text files.
- **Text PC/Data RISC** - A PC with X-station and text processing software LANed to a RISC workstation with the SAS system and data processing software.
- **Standalone PC** - A standalone PC with the SAS system, data processing, and text processing software.
- **Standalone RISC** - A standalone RISC workstation with the SAS system, data processing, and text processing software.
- **Medical PC/Statistical RISC** - A standalone PC based on the existing medical CANDA and a standalone RISC workstation based on the existing statistical CANDA.

![Figure 1: Diagrams of Possible CANDA Configurations](image)

Drug related data is extracted from the current company database (Inquire), is manipulated using the SAS system, and is stored in SAS datasets. The SAS system is needed to run statistical analysis programs contained in the CANDA. Relational database packages were considered for the CANDA data repository, but there were no packages on RISC or PC platforms that could access SAS datasets. One alternative would be to convert SAS datasets to relational tables. This task, however, would require a major revalidation of output from the statistical analysis programs. This resulted in the decision to leave CANDA data in SAS dataset format.

The various configuration possibilities led us to compare user interfaces presented through X-station software (to a RISC machine), native operating system GUI's, versions of the SAS system (6.07 to 6.08), file storage methods, and file access times between a local and networked system.

The X-station software runs in the Microsoft Windows environment, but provides an X-windows interface to the RISC workstation. Although functionally equivalent to Microsoft Windows, X-windows has a somewhat different "look and feel." In these models the CANDA application resides on a RISC workstation server in an Ultrix environment.

Pathworks was evaluated for the Network Operating System (NOS) in the PC client/RISC server model. The FDA and other international regulatory agencies use Pathworks in their LAN environments. Pathworks offers the capability of creating a remote DOS partition on a server. This partition may be assigned a DOS drive letter in the Pathworks setup allowing DOS software direct access to files stored on the remote partition. Pathworks also handles file buffering; it will download portions of an accessed file as needed. Text files from a DOS partition on the server can be directly accessed by the client through Microsoft Word. Data on the DOS partition can be accessed by the client through the SAS system (using librefs). However, the SAS system under the Ultrix operating system on the server cannot access the DOS partition's file structure; the partition appears as one file to Ultrix.

The SAS system on the server must have access to data for executing queries, analyses, and graph programs. The SAS system on the client must also have access to data for representation of table names, variable names, and unique values for user selection. A solution is to store only a list of table names, column names, and unique values in a DOS partition on the server. Dynamic maintenance of this list, however, would degrade response time. Each time a new table was created, a program must be invoked on the server to recreate the list. The regenerated list must then be downloaded to the client, converted to a DOS format, and then uploaded to the DOS partition on the server. These problems led to a decision against using Pathworks for this CANDA.

A second alternative is to use SAS/CONNECT software. This is a SAS system package that provides access to files, hardware resources, and SAS system software on remote systems. It allows access to SAS datasets across platforms and operating systems. However, SAS/CONNECT does not perform file buffering. It uploads or downloads an entire file when remote access...
is requested. This would seriously degrade response time due to extremely large datasets contained in the CANDA system. Furthermore, data access is not transparent to either the programmer or the user. Requests for remote data must be hard coded as "remote submits" from the client to the server. Also, during download from the server, a data transfer "status window" will appear on the user's display. This feature cannot be turned off in SAS 6.07. Note that future enhancements to the SAS system may alleviate some of these problems. SAS Institute plans to implement a rudimentary data dictionary feature that would automatically track table names, variable names, and variable values.

Further evaluation of these configuration models lead to the development of formal screening criteria. These criteria covered a wide range of issues:

- It was decided that the most user friendly interface would have a Microsoft Windows look and feel.
- Reusability of existing CANDA applications would expedite the development process.
- Data security is more easily maintained when data is not transferred across a network.
- A single central data repository would simplify access and validation procedures. Multiple repositories can require coding of multiple access methods. Divergent results can be obtained from analyses and queries based on differences in data storage methods.
- The hardware/software design should allow easy support, maintenance, and troubleshooting. In client/server environments, troubleshooting is more complex because problems may occur in multiple computer systems or in the network connection.
- If the application remains portable, it could ease future development and expansion. The portability of an application to a more powerful environment becomes critical as the application grows in scope.

The initial evaluation concluded with identification of the standalone PC model as the best candidate for further investigation. The client/server models were considered too network dependent. At federal regulatory agencies, the CANDA must be connected to their Local Area Network (LAN). The client would be located in the reviewer's office with the server connected to the LAN from an outside dedicated line. Since benchmarking is not possible at an agency, there is no practical way to determine accurate, repeatable response times in their environment. It was assumed that this configuration could not guarantee adequate performance.

The standalone RISC workstation model was deemed not as "user friendly". The difference between the X-windows and MS Windows GUI was a factor. Also, because there was no existing Ultrix version of MS Word, we could not reuse the text processing component of the original medical CANDA.

**Benchmarking [Response Time]**

Although the standalone PC model passed the initial evaluation, there were still some reservations concerning system response time. The design team decided that benchmarking a prototype PC was necessary. Response time values from the original statistical CANDA and requirements from the FDA reviewers were used as baseline criteria for acceptable performance levels. Plans were devised to benchmark statistical analysis programs and database queries. The effects of CPU and I/O intensive tasks were closely examined to ensure adequate system response time during high-end processing.

The prototype PC configuration, which was later upgraded, consisted of a DEC 66 MHz PC with an Intel based 486 CPU, 20 MB memory, and two 240 MB SCSI hard drives. It was configured with MS-DOS, MS Windows, and SAS 6.08 for Windows. The same benchmarks were attained from the original RISC workstation (DEStation 5000/200) for comparison purposes. Preliminary analysis programs were loaded onto both the PC and RISC workstation. The statistical CANDA application was then loaded onto the PC. (The SAS 6.07 application was converted to transport files and moved onto the PC. There it was imported into SAS 6.08. Very minor modifications were made to make the application execute on the PC.) Preliminary testing was performed on the PC for performance tuning purposes.

The following parameters were "tweaked" during tuning:

- **Sortsize** - this is a SAS system parameter that lets the user specify the amount of memory that is available for sorting data. Changing this parameter can affect the amount of page swapping (writing information normally stored in memory onto a hard drive when all available memory is used up) a process performs. Using more memory will require less paging; consequently, less disk I/O will occur. The only drawback is that memory allocated for sorting takes away from memory that can be used for other sub processes (i.e. SAS data steps and procedures).
- **Windows mode** - this is the operational mode of Microsoft Windows on the PC. Possible modes are standard (which uses fewer resources) and 386 enhanced (which is more flexible). Changing Windows modes showed no major differences during the tuning process.
- **Loadlist** - this is a SAS system option that displays which Dynamic Link Libraries (DLL) are loaded into memory. Preloading DLLs onto a RAM disk can save access time and improve performance. However, with our testing, preloading DLLs had a negligible effect on processing (smaller applications may yield increased performance).

After tuning was complete, benchmark results were gathered from both the PC and RISC machine. The analysis programs and simple queries ran well within the
This can be attributed to the differences in their bus limits as specified by the FDA reviewers. Timing was comparable on both machines. A minor variance in execution times between these machines was noted. This can be attributed to the differences in their bus architecture (the RISC workstation's bus has faster throughput during short burst I/O).

Complex queries could not be run with the old statistical CANDA application due to its inflexibility. Therefore, the SAS SQL procedure was utilized to benchmark these queries.

SQL query benchmarks ran well within the required limits. Also, the amount of temporary work space used was extremely small. Unlike sorting datasets, SQL does not make a copy of the original dataset before processing. With large datasets, making a copy could take longer than the actual sort itself. With SQL, the amount of workspace used by one of the queries was 1.7 MB as compared to 209.7 MB on the RISC workstation with a PROC SORT. Also, this query executed in 0.69 minutes using SQL on the PC and 32.83 minutes using PROC SORT on the RISC workstation (0.84 minutes using SOL). Due to this major difference, an investigation was done on the processing that takes place within the original statistical CANDA.

The original CANDA creates tables during each processing step. Each time a new data table is selected, a copy of that table is made. When multiple tables are joined, they are physically combined into a new table. All of this processing requires heavy disk I/O. When using SQL code, no tables are created and there is minimal disk writing, thus accounting for the difference in timing. These findings resulted in the utilization of SQL for querying in the new CANDA system.

Additional research indicated that sorting was much slower on the PC with large data sets. Windows and SAS system memory management were examined. Attempts to increase performance from the PC in this area were unsuccessful. Additional memory was obtained (24 MB total) in an effort to reduce the amount of paging/swapping that occurred during sorting. The increased memory resulted a decreased execution time.

Tests for high-end processing led to the expansion of the hard disk and another memory upgrade. The increased hard disk space was needed because during a particular query (old system), the PROC SORT used all remaining space causing the procedure to halt execution. The upgraded configuration provided one hard disk with 440 MB, one with 240 MB, and 64 total MB of memory. Sortsize was then increased to 64MB to use all available memory. Query execution still halted due to a DOS error known as a General Protection Fault (GPF).

General Protection Fault

All applications on a DOS based PC share a common memory area (both DOS and Windows applications). This unprotected memory area allows applications to overwrite each other's allocated memory if they do not "play by the rules." A GPF is generated when memory overwriting occurs causing the application to terminate or lockup the PC. When a GPF occurs, DOS provides the name of the application error module and memory location. The GPF we encountered during benchmarking pointed to a SAS system module. This GPF was consistently repeatable (some GPFs are not). It was discovered that it would always occur when the sortsize parameter was set to 45 MB or more and a dataset greater than or equal to this sortsize was being sorted. At this point, SAS Institute became involved in the problem. They asked for logs generated by MS Windows application called Dr. Watson (comes bundled in with native MS Windows) that gives a memory stack dump when a GPF occurs. The support people at the Institute were unable to recreate the problem. However, it was noted that one of the differences in our configurations was the type of SCSI disk controller card in use. Further research concluded that a memory driver needed for the SCSI disk controller was conflicting with the SAS system memory module causing the GPF.

A new SCSI disk controller card with 4.5 MB hardware cache was obtained (this card did not need any drivers) along with a 3.7 GB hard drive. Test queries finally ran to completion with a further decrease in execution time. This final test established that adequate response time could be achieved during high-end processing. These improved results led to the selection of a standalone PC model as the configuration for the new CANDA.

Software and Database Redesign

A software reevaluation process was initiated once the new platform had been chosen. This included an evaluation of FRAME technology within SAS 6.08 for Windows on the PC. FRAME enables the creation of a user interface that looks much like a native MS Windows package.

The SAS system provided the front-end to the entire CANDA. This approach was chosen due to the SAS system's limited implementation of Dynamic Data Exchange (DDE). The SAS system is a client application and does not have server capabilities. This means that the SAS system can read and send data to another DDE application, but cannot receive commands from it. Specific SAS products cannot be invoked from another Microsoft Windows program. To access SAS products, the entire SAS system must be run. This method involves considerable overhead. As an alternative, a much simpler approach was used. The SAS system is invoked once, but then all other MS Windows applications (i.e. MS Word and MS Excel) are called from within the SAS/AF application.

Since FRAME technology was used, an object-oriented approach was taken in redesigning the user interface. Screen standards were developed and new screens were drawn. Object classes specific to the CANDA application were created. This was accomplished by determining commonly used objects across screens (i.e. OK or Cancel push buttons). Each object was created in a temporary
FRAME entries. Object classes were then defined and saved as CLASS catalog entries. The object classes were then placed in a resource list entry (BUILD.RESOURCE in the current catalog). Finally the resource list was associated with the application catalog using the "resource" command. We kept our object classes and the resource list entry in a catalog separate from our FRAME entries. This was found to be a cleaner way of organizing entries. A drawback of this, however, is that each time the FRAME's catalog is reentered, the resource list must be reassociated. Without this command, no newly created FRAME entries will have access to the application specific object classes.

Another drawback encountered was that FRAME technology does not allow creation of complex object classes. (A complex object consists of multiple objects.) Many objects that were common across our screens were complex (i.e. combo boxes, drop down combo boxes, spin boxes, extended tables, etc.). These objects needed to be copied or recreated as each new screen was developed.

In addition to screen redesign, the CANDA database structure and access methods were completely revised to take advantage of the SAS SQL feature. The old database was composed of a few very large tables with a substantial amount of missing data. The new database, however, is subset into smaller tables with related columns to eliminate these missing data areas.

In the old database, certain columns contained codes representing the type of data stored in other columns. These column pairs were transposed, creating a new column for each data type. The resulting table contained more columns and fewer rows. Due to the storage method the SAS system uses in a DOS environment, the "wider" tables utilized much less disk space than the old "longer" tables. These enhancements to the database structure were instrumental in reducing querying time.

The new access method creates SQL views. These views are used as input to the SAS FSMVIEW procedure. This procedure only runs enough of the query to fill one screen worth of data at a time. Once the new database structure and access methods were completed, tests were run on the first queries used during the benchmarking process. The results were a drastic reduction in execution time. There was more than 50% improvement in the response rate.

Prototype Creation - RAD
The next step of development was the creation of a prototype application based on the new design. The purpose of the prototype was strictly to display system functionality to users. FRAME technology was used for Rapid Application Development (RAD). Sample screens were created with only enough code behind them to allow system navigation.

FRAME entries were built for each screen. Small SCL entries (less than twenty lines) were coded for most screens. In some cases, old PROGRAM entries were converted to FRAME and SCL entries. The entire process was completed in about three person-months. In that time span, approximately thirty FRAME and SCL entries were created (only three were conversions).

During prototype development, another repeatable GPF was encountered. Certain screens were designed with many graphics objects. These screens had multiple list box objects, graphics text objects, and scroll bars. When more than one of these screens were displayed in a series, a GPF was generated. The Microsoft Windows Resource Kit was used to identify the cause of this problem. A program within the kit displays memory usage at system, user, and graphics levels. When run in the background, this program revealed that almost all Graphics Device Interface (GDI) memory was being utilized. MS Windows only allocates 128 KB for graphics memory. This number is hard coded and cannot be altered. Each FRAME object consumes a large amount of graphics memory. Other windows products with similar screen objects utilize much less memory than FRAME objects. SAS Institute has no current solution for this problem. These screens were altered to use fewer graphic objects. Specifically, extended tables were implemented instead of the multiple list boxes. Also, CALL DISPLAYS to these FRAMES were changed to CALL GOTOs. This clears all memory space (including GDI) utilized by previously displayed FRAMES.

Development
The majority of the user interface was completed during the prototype creation. The remaining development added background processing for full system functionality. Also, storage and retrieval algorithms were rewritten to take advantage of the new list structure features in SAS SCL.

SAS 6.08 for Windows provides a data structure called a list. This structure is important for efficient use of FRAME technology. They are similar to arrays, but can change size dynamically. They can be created or deleted as needed. They are loaded into memory for use, making processing quick. Populating list boxes is extremely simple when using SCL lists. List boxes are filled directly by associating a list structure to the object. These lists can be stored as catalog entries allowing them to be used at a later date or for passing information between modules.

Conclusion
Throughout this evolutionary process, several positive and negative points of the SAS System for Windows were discovered. Object oriented design and rapid application development are effective state of the art methods that can easily be implemented with FRAME technology. The enhancements to SAS/AF (list data structures and multiple extended table objects per screen) enable building well-organized screens with responsive supporting code. A mouse driven interface with common
Windows objects allowed for the creation of a user friendly system. However, building a large, complex application always tests the limits of the tools used. General Protection Faults are system halting problems that are very difficult to resolve. There is no overall solution for these DOS errors. Better memory management will be provided in a more robust operating system. Changing operating systems may require another version of SAS. Hopefully later versions will also provide more efficient use of graphics memory. Improvements in FRAME tools allowing the creation of complex objects will also be a welcome enhancement.

Migrating our CANDA from SAS 6.07 on the RISC platform to SAS 6.08 for Windows on the PC/Windows platform has been a significant upgrade. Recent hardware improvements made using the more familiar PC platform possible, but the system enhancements could not have been achieved through the hardware change alone. Redesigning the user interface with FRAME technology, modifying the database structure, and using SQL for data access methods were essential. The combination of these alterations has enabled us to provide regulatory agencies a more responsive and user-friendly system.

Acknowledgments
Special thanks to the following people for their input and support to this CANDA project:

Nancy Bauer  Merck Research Laboratories
Dave Brummit  SAS Institute, Inc.
Jeff Cartier  SAS Institute, Inc.
Mark Cates  SAS Institute, Inc.
Steve Champion  Ziff Technologies, Inc.
Barry Cohen  Planning Data Systems
Andrea Contino  Contino Associates, Inc.
Scott Ferris  Merck Research Laboratories
Bob Krzysik  Systems Unlimited
Gale Lowden  Merck & Co., Inc.
Kim Nosky  Merck Research Laboratories
Scott Seyfarth  Merck Research Laboratories
Loretta Schlatzer  SAS Institute, Inc.
Todd Welch  Merck & Co., Inc.

Trademarks
DEC, Ultrix, Pathworks are a registered trademark of Digital Equipment Corporation, Inc.

Inquire is a registered trademark of Infodata Systems Corporation, Inc.

Microsoft, Windows, Word, Excel, Dr. Watson are a registered trademark of Microsoft Corporation, Inc.

SAS, SAS/AF, SAS/CONNECT, SAS/FSP, SAS/GRAPH, SAS/STAT are registered trademarks of SAS Institute, Inc.