Paper 28-25

A Submission Development Environment to Support SAS® Programming and Related Activities During Clinical Data Analysis

Barry R. Cohen, Planning Data Systems, Inc., Ardmore, PA

ABSTRACT

Pharmaceutical companies conduct many activities during the statistical analysis of their clinical data. This is one part in their larger process of collecting, managing, analyzing, and presenting all this clinical data (safety and efficacy). Many companies now provide considerable system support for the early part of this larger process, (i.e., the data collection and management), and for the later part when the New Drug Application (NDA) document is published and presented to a regulatory agency in electronic form. But much less has been done to support the work involved in the statistical analysis of the data using SAS Software, particularly Base SAS and the Macro Language. I discuss automated, integrated support for the statistical data analysis in this paper. I call this support a Submission Development Environment. An automated, integrated environment for this work is surely a challenge, but it offers much potential payoff if it can be achieved.

INTRODUCTION

As pharmaceutical companies conduct clinical trials, they generally follow the top-level activities presented in Table 1 regarding the clinical data (safety and efficacy data). Table 1 indicates that much attention has been given in this industry to system support for clinical data preparation and management, for production of standard safety data reports (count and listing reports), and for publication of the full NDA (New Drug Application) as an electronic document. For example:

- Large-scale Clinical Trials Systems have been and are being developed (both in-house and by vendors) to handle clinical data collection, preparation, and management.
- These Clinical Trials Systems also provide a bevy of standard safety reports.
- Document Management Systems are used to publish New Drug Application (NDA) documents.
- Electronic Submissions, formerly called Computer Assisted New Drug Applications or CANDAs, are used to present the full NDA submission as an electronic document.

But the work involved in the statistical analysis of the data (primarily the efficacy data but not strictly limited to it) has received substantially less system support. I refer to this statistical analysis work as the PAWRS process, where PAWRS is an acronym for **P**rogram, **A**nalyze, **W**rite, **R**eview, **S**ubmit. (The PAWRS process is described further below). So, for example, often:

- SAS programmers are given less (or little) support when developing statistical analysis SAS programs. No environment, beyond SAS Display Manager, supports them as they code, test, debug, store/retrieve, validate, and document the generations of their programs. Nothing like the robust program development environments of other programming languages is available.

Activity	Major System Support
Prepare, manage clinical data (safety and efficacy)	yes
Produce basic safety count and listing reports	yes
Develop SAS programs for statistical analysis (primarily for efficacy data)	no
Run SAS programs, analyze data	no
Write statistical analysis document	no
Review statistical analysis document	no
Publish and present full NDA as electronic document (including the statistical analysis portion)	yes
Present statistical analysis programs and data separately for regulatory review (“Statistical Review Aid”)	no

Table 1: Activities Regarding Clinical Data and its Statistical Analysis

- Statisticians are given less (or little) support as they analyze the data and write the statistical analysis portion of NDA documents. No environment, (again beyond SAS Display Manager), supports them as they retrieve and execute the various SAS programs, as they access the clinical data resident in a database environment outside the SAS environment, as they retrieve and review the program outputs, and as they move the program outputs (tables and

graphs) into the analysis documents they are writing.

- Statisticians and other reviewers are given less (or little) support as they review the analysis document. Limited support is available that facilitates access to the document, or annotation and distribution of review comments in the document. And, in particular, no environment is provided that allows movement from the document text and tables back to the program and data environment to answer questions about the exact programs and/or data involved.
- Statisticians and statistical programmers are given less (or little) support as they gather together the various files that comprise the statistical analysis programs and SAS data sets, and submit these programs and data to the regulatory agency for a review of the efficacy analysis (sometimes called a "statistical review aid").

The limited amount of system support, and especially integrated system support, for the statistical analysis process (i.e., the PAWRS process), historically, is understandable for two reasons: (1) It tends to concern efficacy data which is not standard from drug project to drug project. Safety data, in contrast, is much more standard and thus lends itself more readily to automated system support. (2) Statisticians are usually also SAS programmers and thus have some skills to operate in a native computer environment outside of an automated application. So the need to support them has been less pressing than it has been for the non-programming clinical/medical staff involved in clinical data collection, preparation, management, and review.

But the limited statistical analysis support is still somewhat surprising today because this process is a major part of the larger clinical trials process, and effective system support could have a commensurate major impact. This is the subject of my paper. I will first discuss the PAWRS process as a process, describing its activities and where I feel system support is possible. This is a necessary precursor to building an effective software application to support and integrate the activities of the process. I will then present some thoughts about what a software application to support the process might look like. I call this application a *Submission Development Environment* (SDE).

My ultimate goal is to build an application or environment in which the tasks of the statistical analysis occur, including the SAS programming tasks, and which culminate in the submission of the results to a regulatory agency (i.e., submission of the SAS analysis programs and analysis SAS data sets). I hope to foster more dialog within the SAS community on this subject through this paper, just as I tried to do with my SUGI23 paper (see Cohen, 1998), and my SUGI24 paper (see Cohen, 1999).

PROGRAM-ANALYZE-WRITE-REVIEW-SUBMIT PROCESS

The activities of the PAWRS process are illustrated in Figure 1. The process begins in the lower left of the diagram with the Program Development Environment (PDE), where Base SAS/Macro Language programs are developed and executed. The activities in the PDE include writing, testing, debugging, validating, and documenting code, and eventually executing the code in production to produce the tabular and graphic outputs which express the analysis in raw form and which will become part of the analysis document.

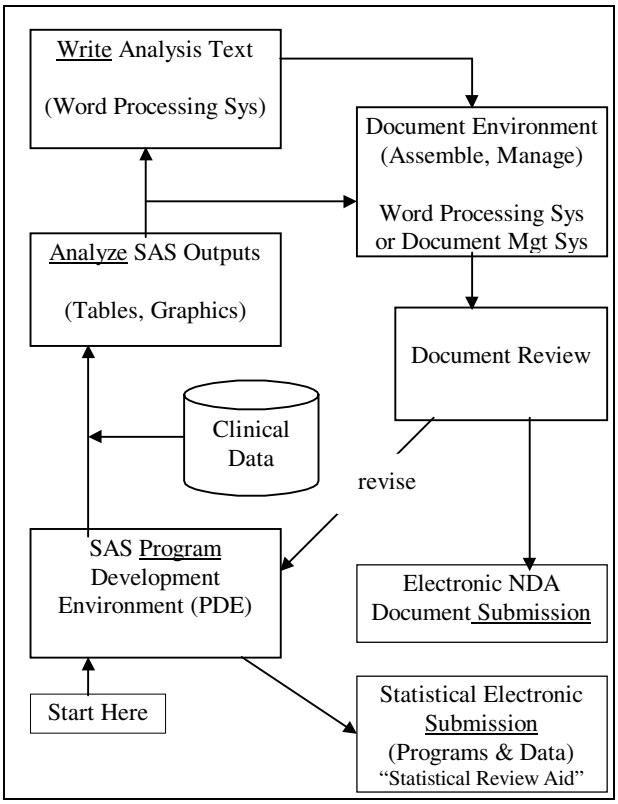

Figure 1: Program-Analyze-Write-Review-Submit Process for Statistical Analysis of Clinical Data

The process continues as the statistician writes the text that discusses the results of the analysis, referring to the SAS tables and graphics. This typically occurs in a word processing environment. The tables and graphics, which are SAS outputs, are incorporated into the word processing environment. Other text is typically also written, generally before the analysis even occurs, (e.g., a Data Analysis Plan), and this text is also incorporated into the analysis document. A file management tool such as MS/Explorer or MS/File Manager is typically used to access and modestly manage the various files involved.

Alternatively, a document management system, (a different genre than word processing software), may be the environment where the text and the SAS outputs are assembled and managed. If so, the SAS outputs may be loaded directly into this software-controlled environment instead of first loaded into the word processing environment.

The next activity of the PAWRS process concerns review of the document. Typically in the pharmaceutical industry many people beyond the statistician-author are involved in the review. This activity includes feedback and questions to the author from the reviewers. Often this requires revising and re-running programs to re-produce SAS outputs and re-write analysis text, which then must be re-loaded to the word processing environment or document management environment. The PAWRS process iterates through these steps as many times as necessary until all involved agree that the analysis is done correctly and the results are expressed in a document correctly.

Finally, once the statistical analysis is complete and the analysis document is written, the analysis document is combined with the remainder of the NDA document material, published, and typically submitted electronically (as well as in hard copy) to the regulatory agency for review. This is sometimes referred to as an Electronic Document Submission. And the statistical analysis (i.e., the SAS analysis programs and SAS analysis data sets) may also be submitted in a Statistical Electronic Submission, sometimes called a "Statistical Review Aid".

PAWRS Activities Typically Not Integrated

I find that across the pharmaceutical industry today, there are varying levels of system support *within* each activity of the PAWRS process. And I see this support as mostly not integrated *across* the activities. Typically, there is no one software application that manages and ties together this PAWRS process. In fact, I am aware of only one software application available today that provides at least some integrated support for the PAWRS process along the lines I suggest herein. That product is SAS/PH-Clinical™. (See, for example, Villiers, 1997).

But more can be done, by the SAS Institute in its SAS/PH-Clinical product, and by others in their own in-house environments. I present some of my ideas in this regard in the remainder of this paper.

SUBMISSION DEVELOPMENT ENVIRONMENT COMPONENTS

As I think about how to automate the phases of the PAWRS process, and integrate these individual applications, I begin to think of an application or environment with a name something like an *"Analysis Development Environment"*, or a "S*ubmission Development Environment*" (SDE). The idea is to create an application or environment that facilitates all aspects of the work done by statisticians and statistical programmers during the PAWRS process. This is an environment that allows these people to spend more time focused on the actual analysis and the actual program development, and less time negotiating the computer environment and integrating the various pieces of work they produce. It is an environment that supports statistical analysis of clinical data, culminating in submission of the analysis to a regulatory agency.

In this section, I briefly discuss several of the various components I envision for this environment. The discussion of each component is brief due to the length limition of this paper and the significant breadth and depth of the subject. Indeed, any one of the components could be the subject of an individual paper. Further, there are other possible components that I will not discuss today, for the same reason. But I have chosen the components that I think a pharmaceutical organization is more likely to implement first. My plan is to discuss additional components at the next SUGI, as my own project work in this area continues to unfold.

In the discussion below, the term "users" should be understood to primarily mean statisticians and statistical programmers.

Standard Directory Structure

The SDE environment will handle many and varied file types, for many projects simultaneously, and accumulate projects over time. Such an environment requires a pre-specified repository for the many files and file types to be generated. The idea is to determine up-front where everything will go. This is important if a full SDE is built and each component application needs defined storage locations for its activities. It is also important even if some of the work is still accomplished without the help of automated applications. In this case, we can at least make all the storage decisions for the users, even if they have to read and write in these areas without further automated assistance.

I call this SDE component the "Standard Directory Structure". It will have the following features:

- Organized – The structure is organized to reflect all the activities of the PAWRS process within a drug project, and the relationship across drug projects. This structure, coupled with the directory naming scheme, makes the location of particular items intuitively obvious to users.

- Centralized, yet Scaleable – The structure allows the work of all the many drug projects to be stored in the same place (i.e., the same hierarchy, the same structure). Given that the number of projects will grow large over time, the structure is located physically where it can scale.
- Consistent – The structure is generic enough to work for all the drug projects, so the work will be done consistently from project to project within the company.
- Controlled – The structure represents a centralized environment and thus a shared environment. It is shared among members of the same drug project, and among the members of the multiple drug projects. Thus, this component will provide tools to allow the individual project teams to restrict access to their teams' files, as needed.
- Structure logic is built around the submission - Larger pharmaceutical companies tend to handle many drug compounds. For them, the primary organization for centralized storage is probably the drug compound. But the active project unit within the structure is probably the submission project, for most companies (large and small). The submission project is comprised of all the files of all the activities that comprise one submission to a regulatory agency. This means that a new set of directories is added to the structure as each new submission project begins, with as many protocol directory subsets as needed. And the submission directory set is located beneath the appropriate drug compound.
- Types of files – The structure will have directories for all the file types involved. For example, SAS programs, SAS data sets and catalogs, SAS outputs (log, listing, reports in external files, analysis documents in word processing form, etc.
- Complexities – The structure will accommodate the many complexities that arise during the course of statistical analysis of drug project clinical data. For example, pre-planned locations will be available in the structure for interim analyses, extension studies, integrated studies, safety update studies, etc.
- Test and Production areas – The SDE environment will use a structure that provides separate areas for development (i.e., testing) and production work. This will increase control over the production work, and thus the integrity of what is submitted to the review agency.
- Utility to create the structure – The directory structure used per submission project will have many individual directories. And this structure will exist many time in the SDE environment because of the many submission projects involved over time. Thus, this component of the SDE environment will include a utility program to create an instance of the directory structure as each new submission project begins. Users will not have to manually create the large set of directories involved for their project.

Database Extraction Facility

Today, a company's clinical data to be analyzed is often resident in a database environment, outside the SAS environment used for the analysis. The users must access the data from the database, yet they tend to know less about this particular aspect of SAS programming than others. So, the SDE application will make it easy for users to access database-resident data and convert it to SAS data sets. This component has the following features:

- Database Information – The component provides full information about what data is available in the database for a given drug project. The user does not need to have database administrator knowledge, or know how to write SQL code to examine the database contents.
- Non-programming environment – Extractions from database tables to SAS data sets are accomplished using menus on the interface. Users do not write SLQ or SAS PROC SQL code to effect the extractions.
- Control over rows and columns – The non-programming requirement also covers the ability to select a subset of rows from a database table, and/or a subset of columns.
- Custom SAS variable names and labels - SAS provides default variable names and labels when SAS/ACCESS® extracts database tables to SAS data sets. These are often uninformative, or ambiguous, or misleading. This component will store and use a set of user-defined custom variable names and labels. This is particularly true for any database tables that are standard across protocols and projects.
- Standard SAS data set structure - If the users have a standard SAS data set structure they use for analysis, and if the corresponding database table is not in this structure, then the component will re-structure the data upon extraction.
- Use of Standard Directory Structure – The component will know where to store the results of the extraction process, within the Standard Directory Structure, based upon the user's identification of a particular drug project. The user will not have to input text strings that identify directory paths.

Program Development Environment (PDE)

Much program development in the software industry today occurs within a PDE. A PDE, simply put, is a software application or environment that facilitates the development of software. PDE's are provided today for many program languages and for many application development tools. PDE's generally provide automated

support for the common programming activities that are done repeatedly, that tend to be tedious, and that can feasibly be supported by a software application.

SAS Display Manager is a PDE for the base SAS Macro Language, albeit it is less robust than the better PDE's seen today. It also does not add much support for the Macro Language beyond what is provided for base SAS. SAS Macro Language programming presents its own unique challenges, and pharmaceutical SAS programmers use the SAS Macro Language often. Hence, a SAS PDE that is part of the SDE application should include a Macro Language focus, too. The SAS PDE will have the following features:

- Custom Program Editor – The program editor will be customized to the Base SAS and Macro Language. It will have a syntax debugger, built-in coloring options for the program statements, and built-in indentation options for the program statements.
- Run-Time Debugger – The PDE will allow the user to watch the code execute, line by line, with trace facilities available for variable values. (This is similar to what is available for the SAS SCL language).
- Symbol Table Generator – The PDE will generate a symbol table which indicates each variable and keyword in a SAS program, and each location (program line number) where it is found.
- Code Documentation Support – The PDE will help the programmer document the code. This will involve a "documentation window" with a template to be completed for each program. The template will have all of the company's required sections for its SAS program documentation.
- Source Code Library Management – Tools will be provided to manage the source code library. This will include:
  - support for storing generations of a program
  - maintenance of a change history and the provision of a "Difference Report"
  - a check-out/check-in facility for code development in a shared, team environment
  - the ability to search the library by "program subject" index or "program purpose" index. The index will be created as the documentation template is completed.
- Macro Parameter Handling Support – A tool will help users handle the macro parameters of each macro program. Specifically, the PDE will provide a "Parameter Properties Sheet" that lets the user input and store a name, use description, default value, acceptable other values, and current run-time value for each macro parameter. This spreadsheet format will be easier to work with than the present format for this information, which is basically the header documentation in a macro program, and in the statement that invokes (calls) the macro. Users will set parameter run-time values in the spreadsheet with more ease and more accuracy that they do now by editing source code.
- Macro Call Generator – The data in the Parameter Property Sheet will be used by another tool to generate syntactically correct macro calls to be inserted into emerging programs.
- Macro Library Search Support – A tool will identify and report to the user all the places in the program library where a given macro is called from.
- Use of Standard Directory Structure – The component will know where to store the source code, or source code documentation, or macro parameter properties sheet, or whatever, based upon the Standard Directory Structure and the user's particular drug project. The user will not have to input text strings that identify directory paths.

Program Execution Facility

The process of executing a statistical analysis program involves handling many files at once. These include, for input: source code files; SAS data sets; SAS catalogs (e.g., formats, macros). For output, these include: logs; listings; graphics catalog entries; tables or graphics as external files; SAS data sets. Further, for a given statistical analysis, any analysis program(s) might be executed several times. For example, the patient group might change and the same analysis table or graph program run again. Or, the efficacy parameter being analyzed might change but the patient group and the analysis output table or graph remain the same.

So there are several files involved with each program execution, and the executions occur many times. Users spend much time managing all the input and output files during development of an NDA. The SDE will have a component to assist with this process. I call this component the Program Execution Facility. Some readers might view this component as just an additional part of the SAS PDE, and this seems equally reasonable. It will have these features:

The Program Execution Facility will function around the concept of a "run-set". The run-set is comprised of all the items involved in one execution of a program, both input and output items. The facility has a user interface that allows the user to provide a run-set identifier (a name) and a run-set description (a text description) and then all the program inputs and outputs are cataloged by this run-set identifier. The identifier includes a time-date stamp, too, appended at the time of actual execution. The idea is that the user can easily retrieve and easily browse all of the inputs and outputs of the run-set together.

Further, the Program Execution Facility will support the process of setting run-time values of macro parameters using the Parameters Property Sheet that was described above as part of the SAS PDE. This sheet will be stored as one of the run-set items for each macro involved in the execution. Then, for example, to determine what values the macro parameters held at run-time, a user will retrieve the Parameter Property Sheet after execution, instead of the present method of searching the program log to see what values were resolved and shown in print.

Finally, the storage location for all items that are handled by the Program Execution Facility will be based upon the pre-defined Standard Directory Structure and the user's drug project. The user will not have to input text strings that identify directory paths.

Statistical Electronic Submission Facility

A Statistical Electronic Submission provides the regulatory agency a set of SAS programs, catalogs, and data sets that comprise the statistical analysis that has been conducted. This is sometimes called a "Statistical Review Aid" or "Statistician's Review Aid". Automated support for this submission can save time and labor, and improve accuracy.

The SDE will have a component to assemble and test the Statistical Electronic Submission. It will have these features:

- It will be built around the Standard Directory Structure which has specific directories that store the files to be submitted separately from other files. Thus, the process of gathering together the files to be submitted will be an on-going process during the NDA development instead of a scramble at the end of the process.
- A tool will allow a user to review a directory listing and mark/unmark individual files for inclusion in the submission. This tool will use any file descriptors that are part of the SAS PDE and SAS Program Execution Facility to assist in the determination. An "inventory of files being submitted" report can be generated from the SDE environment, even before the files are actually moved to the "submission staging area".
- A tool will generate and save a customized PROC Contents summary of each SAS data set being submitted.
- A tool will convert each SAS data set being submitted to SAS Transport File form.
- A tool will perform a test execution of each submitted program, after it is moved to the "submission staging area". This will insure that all required files are present for successful execution, and no errors occurred. The search of logs for errors will be automated, too.
- This component will either keep the files being submitted in the SDE Standard Directory Structure, or convert the storage to an agency-requested structure, as required.

Important SDE Benefit: Program and Process Validation

Regulatory agencies look at the full environment within which the programming and analysis process occurs. They are not concerned only with how individual analysis programs are validated. They want the analysis to be conducted according to a pre-defined, organized, tested, and controlled set of procedures regarding the handling of all the inputs and outputs involved. You can help achieve this goal when you conduct your analysis in an SDE environment. In short, an SDE represents a pre-defined, organized, tested, and controlled set of procedures regarding the handling of all the inputs and outputs involved.

REFERENCES

Cohen, B.R. (1998), "Supporting the Program-Analyze-Write-Review Process with a Development Environment for base SAS and the Macro Language", *Proceedings of the Twenty-Third Annual SAS User Group International Conference*, 23.

Cohen, B.R. (1999), "The Pharmaceutical Program-Analyze-Write-Review Process and a SAS Program Development Environment to Support It", *Proceedings of the Twenty-Fourth Annual SAS User Group International Conference*, 24.

Villiers, P. (1997), "New Architecture for Linkage of SAS/PH-Clinical™ Software with Electronic Document Management Systems", SAS Institute, Cary, NC, *Proceedings of the PharmaSUG '97 Conference*.

CONTACT INFORMATION

Barry R. Cohen,
Planning Data Systems, Inc.
PO Box 666, Ardmore, PA 19003
610-649-8701 cohenbar@bellatlantic.net

Mr. Cohen is an independent information systems consultant, specializing in application development and other support for analytic processing. He has been using SAS software since 1980 in a variety of industries, including a focus on the pharmaceutical industry.