A Computer Assisted New Drug Application (CANDA) Using SAS/AF Frame and SAS/PH-Clinical

Ai-Yu Wu and Doug Brown, Genentech, Inc., South San Francisco, CA

ABSTRACT

This presentation describes some of the approaches we used in developing a Computer Assisted New Drug Application (CANDA) for use by a Food and Drug Administration medical reviewer. We first describe some strategies for loading clinical data into SAS/PH-Clinical® Release 1.0. Second, we describe various SAS/AF® Frame entries developed to supplement SAS/PH-Clinical.

INTRODUCTION

We assume the reader has been introduced to SAS/PH-Clinical, but may not have used it. We assume no familiarity with SAS/AF Frame and Screen Control Language (SCL).

We will discuss some of the issues involved in implementing a study in SAS/PH-Clinical and will show how we resolved them for our CANDA.

SAS/PH-Clinical provides one menu icon which can be customized to be an entry point to a SAS/AF application. We used this as the entry to the Genentech Clinical System which consists of several SAS/AF Frame menus with extensive use of SCL. We will describe the features of these Frame menus without discussing the underlying SCL code.

CANDA Development Strategy

We have found SAS/PH-Clinical to be a powerful and efficient tool that enables the CANDA developer to create a clinical review system rather quickly. It contains various reporting and analysis modules that become functional once the clinical study data is defined to PH-Clinical.

This ease of implementation has its costs. For instance, PH-Clinical creates outputs in a standard layout that can be modified, but not totally designed to meet the needs for a specific protocol. Also, the PH-Clinical menus were developed for the use of statisticians and other SAS users, and are difficult for other clinical researchers to understand.

We decided to use PH-Clinical as the core of our CANDA, and to supplement it with AF Frame entries to make it more useful for medical reviewers. The Frame entries access the same data as PH-Clinical by reading the PH-Clinical created dataset "work.phdata".

One advantage of AF Frame entries is that they can be designed for our specific type of study (multi-visit study) and our medical indication. Another advantage is the convenient user interface of Frame entries using a Graphical User Interface (GUI) that is readily understood by medical reviewers. These interfaces can be created rather quickly.

The real power of such entries comes from SCL code that may be developed for each object on the screen, as well as for the entire screen. This SCL code is considerably more difficult to develop than the GUI interface, but the payoff is enormous in the kind of interaction with the user obtained.

In addition to SCL code, standard SAS code may be run from AF Frame screens in order to create outputs that are tailored to your study and your users.

We chose a PC platform because our ultimate users were reviewers at the U.S. Food and Drug Administration who were already familiar with IBM® PCs running Windows. We chose an OS/2® system for this application, because published reports indicated SAS software runs faster under OS/2 than under Windows.

PH-Clinical Implementation

Data Preparation

Our main analysis dataset on the mainframe consists of one record per patient. This one record has all efficacy and lab data for a given patient over a series of visits. However, the record becomes quite long, and typically has over 3,000 variables.

This dataset needed to be transposed for two reasons. First, the number of variables per observation has a strong effect on performance on the PC. That is, the length of the program data vector (PDV) should be kept as short as possible. Second, PH-Clinical is designed to accept repeating data observations on a 'by visit' basis, so we needed to create such observations before loading into PH-Clinical.

Also, because PH-Clinical creates one large dataset (work.phdata) to operate on, it is important to understand how PH-Clinical merges the datasets loaded into it. The number of observations can increase dramatically if certain types of datasets are merged.

For instance, the "PE" (one observation per patient event) dataset, such as concomitant medications, is a type that can have multiple observations per patient, but these are not tied to any particular visit. If we were to merge two such datasets, then each observation in the first dataset must be merged with each observation in the second dataset for the same patient. For instance, if patient 1 has 5 observations in dataset A and 6 observations in dataset B, then the merged dataset will have 30 observations for patient 1. This new dataset is not only rather large, but it also contains duplicate values on separate records. Such a dataset could slow down processing and possibly lead to errors because of the duplicate values.

In order to minimize the size of the working dataset and to minimize the duplicate data, we performed our own merge in a data step prior to loading the data into PH-Clinical. So in the example above with 5 observations in dataset 1 and 6 in dataset 2, the resulting dataset now had 6 observations for patient 1, and no duplicate data. We loaded the resulting dataset as a "VE" dataset (one observation per visit event).

Study Definition

The PH-Clinical user guides and references are very good at describing proper study definition. We will not describe that process here. We followed their advice in creating the various levels of datasets - P, VE, etc.

Initially we kept demographic variables only in our main analysis dataset which was loaded as a "VE" type dataset. Since PH-Clinical merges all defined datasets into one working dataset (work.phdata) we assumed that demographic data would always be available for analyses. Unfortunately, if we were analyzing data originally from other datasets (such as Adverse Events), we frequently had observations with missing demographic data which would give erroneous statistical or listing results. We discovered that this was due to data collected at a given visit, for which there was no data in our main dataset.

In order to correct this problem we created a separate demographic dataset with a level of "P" which is used by PH-Clinical to merge demographic data onto all observations in the working dataset.

AF Frame

We developed the AF Frame entries with two main purposes in mind: 1. to make the user interface as user-friendly and fast as possible, and 2. to tailor the screens to fit our specific NDA (New Drug Application). We will explain our approach for each of these purposes and discuss example Frame entry screens.

1. User Interface

a. Selecting Variables

PH-Clinical relies on variable names and variable labels when allowing the user to select data for analysis. This can be cumbersome for a variety of reasons: 1. variable names are limited to 8 characters, so are frequently meaningless. 2. Labels are somewhat useful, but are only displayed to the right of the variable names , 3. with a large number of variables, for example 275, the selection list is quite long, and 4. the selection list is not easily sorted or grouped in a useful way.

Our solution to this was two-fold. First, we used Frame "icon" objects to create rectangular buttons on the screen that users can click on. Typically, these represented certain groups of variables, and would lead to a secondary menu with icons for each variable in that group. We labeled all icons with clinically meaningful terminology. Our Laboratory Listings menus are good examples of this approach.

Our second approach was to use Frame "list box" objects on the second menu to enable the user to scroll through a small list of variables or values. Our scatterplot menu is a good example of this.

b. Patient Subsetting

PH-Clinical provides for creation of patient groups that are then used in all analyses throughout the PH-Clinical environment. We wanted a similar functionality in our AF Frame menus in that any patient subset created would carry through to any other menu during that session.

We decided to implement our own patient subsetting menus, as we considered the PH-Clinical approach cumbersome for medical reviewers for three reasons. First, the variable selection was awkward, as discussed above. Second, the subsetting criteria was expressed in SAS code. Third, the criteria used for subsetting was not readily available at all times.

Since we wanted Frame created patient subsets to be available to both our Frame entries and to PH-Clinical it was important to maintain the structure of the work dataset. This meant that any subset had to contain all the variables in the input dataset. In addition, for those patients not excluded, it was important to keep all observations. Thus, our subsetting was strictly 'patient' subsetting, and not some subset of variables.

Our approach was to use icons and list boxes to make it easy for the user to get to the variable of interest. Our main (parent) subsetting menu has icons that represent groupings of clinical data. By clicking on the appropriate group icon the user selects the subsetting menu desired.

The subsetting menus (child menus) use icons and list boxes to guide the user in variable selection. Once a variable is selected, then additional objects appear to guide the user in specifying criteria.

All menus in the subsetting group share four additional functions: 1. Number of patients in the currently active dataset is displayed, 2. a push button to cancel all subsetting, 3. a push button to cancel the last subsetting. and 4. access to a window displaying all subsetting criteria applied in the current session.

c. Text Search

The PH-Clinical OUTPUT Window does not provide for searching for specific text in any report generated. Since many reports in our CANDA are quite lengthy, such search capability is important to users.

In our Frame entries, when reports are generated, the SCL code automatically opens the standard SAS OUTPUT window which does have text search capability.

2. NDA specific enhancements

a. Structure like paper NDA

Since medical reviewers will be referring to the paper copy of the NDA, as well as the CANDA, it is important to be able to go between the two easily. To facilitate this, we structured our AF Frame menu system to reflect the table of contents in the paper NDA. For instance, our Main menu shows icons for Tables, Listings and Adverse Event reports which parallels our paper NDA.

b. Identical terminology to paper NDA

The use of graphic text objects, icon objects and list boxes enabled us to use the same terminology as in the paper NDA.

c. Output reports identical to paper NDA

Our AF Frame menus generate a wide variety of reports using the same SAS code used to generate the paper NDA. The resulting Output has the same titles, footnotes, labels and layout as the reports in the NDA. The only difference is if the user has created a subset, then only those patients are included.

While PH-Clinical does have ways to run our SAS code, it is not easy for the medical reviewer to understand. In addition, the PH-Clinical OUTPUT window is not as wide as the standard SAS OUTPUT window, and it does not have the text search capability of the standard SAS OUTPUT window. By making our standard OUTPUT window as wide as the display screen we were able to display most reports just as they appear in the paper NDA.

d. Customized Patient Drill-down

While PH-Clinical does provide drill-down to individual patient data from a larger report, these standard drill-down reports are not modifiable to any great extent.

Using AF Frame entries we created our own NDA specific patient drill down which runs SAS code (data _null_) to generate a patient report similar to that in the paper NDA. Our crossplot display is a good example of this. The user clicks on a data point on the graph displayed, and this runs a detailed report for that patient.

CONCLUSION

SAS/AF Frame is a powerful tool to develop a CANDA to suit particular needs. While development time is substantially longer than that required for implementation in SAS/PH-Clinical the advantages of a user friendly interface and NDA specific screens and analysis make the extra time worthwhile.

ACKNOWLEDGMENTS

We are grateful to Guy Pawson of Genentech, Inc. for his advice and assistance in implementing SAS and SAS/PH-Clinical in an OS/2 environment.

We wish to thank Greg Smith of SAS Consulting, Rockville, MD. for his valuable advice on developing AF screens and for his SCL coding of the more advanced Frame entries. We also wish to thank Sandra Schlotzhauer of SAS Inst., PH-Clinical development manager, for her advice on interfacing our Frame entries with PH-Clinical.

SAS, SAS/PH-Clinical, and SAS/AF are registered trademarks or trademarks of SAS Institute, Inc. in the U.S.A. and other countries.

IBM and OS/2 are registered trademarks or trademarks of International Business Machines Corporation.

® indicates U.S.A. registration.

Other brands and product names are registered trademarks or trademarks of their respective companies.