A FLEXIBLE USER INTERFACE DEVELOPED USING SAS/AF® AND SAS/GRAPH®

Jane P. Loftis, Oak Ridge National Laboratory
Teresa L. James, The University of Tennessee

ABSTRACT

Oak Ridge National Laboratory (ORNL) was asked by the Military Traffic Management Command's Personal Property Directorate (MTPP) to design and prototype a decision support system to quickly and easily locate and display data to assist them in their task of evaluating and monitoring the Personal Property Program. This paper describes the user interface developed in SAS/AF® and SAS/GRAPH® that supports the system and explains how the SAS® user interface works in conjunction with an ORACLE® interface. It also discusses the goals of the SAS user interface, describes the screens and pop-up windows developed for the interface, and explains how these screens achieve the system goals.

BACKGROUND

The Worldwide Household Goods Information System for Transportation Modernization (WHIST-MOD) being designed and prototyped by Oak Ridge National Laboratory (ORNL) is a decision support system for the various organizations of the Military Traffic Management Command's Personal Property Directorate (MTPP) that establish and implement the Personal Property Movement and Storage Program. ORNL was not tasked with implementing the entire design, but rather was asked to prototype significant pieces that will serve as the basis from which MTPP staff will implement and expand the system. The decision support system will benefit the staff of the Personal Property Program in their tasks of program evaluation and policy setting. WHIST-MOD will be a dynamic system that evolves to meet the changing needs of the MTPP staff.

This system is designed to access a centralized database through a powerful set of information management tools. The prototype system offers users, even those with minimal computer experience, easy access to a large selection of data elements and the ability to perform complex queries of the database.

User requirements for the WHIST-MOD system were identified during the analysis phase of the project. During this phase, ORNL identified three modules that needed to be prototyped to aid decision support activities at MTPP. These modules include applications to provide a description of the database (system dictionary applications), applications to manage data (data acquisition and administration applications), and applications to retrieve and display data from the database (user applications). This paper describes the user applications module, which was prototyped in two design and development phases.

THE APPLICATIONS MODULE

Round 1, the front-end interface, was prototyped using a relational database management system, ORACLE®, and its associated toolset. The front-end interface includes screens that allow users to choose, retrieve, and store a subset of the data contained in the database. These data are then passed to the back-end interface.

Round 2, the back-end interface, was prototyped using SAS software (i.e., base SAS, SAS/AF, and SAS/GRAPH). The back-end interface allows the user to

- specify report types and formats
- produce output based on the dataset passed to the back-end interface from the front-end interface.

GOALS FOR THE INTERFACE

We identified the goals for the interface during the requirements analysis phase of the WHIST-MOD project. The goals were based on our analysis of current user needs and our understanding of the current and future needs of the users and the system itself. The following is a list of goals for the interface:

- The interface must support modifications and expansion.
- The interface must be generic enough to be used to produce a variety of reports.
- The interface must be easy to use.

Modular Design

Because of the evolving nature of the personal property business, the interface must easily incorporate modifications and expansions. The need for modifications and expansions may come from a variety of sources. As MTPP staff monitor the Personal Property Program, they will identify the need for changes and additions to the program. For example, MTPP is currently in the process of consolidating separate quality assurance (QA) programs, International QA and domestic QA, into one standardized QA program. Once this new QA program is implemented, the WHIST-MOD system will need to support this new program. Also, external sources, such as Congress, the carrier industry, and service members, may create the need for changes in the Personal Property Program. For example, because of the changing needs of service members, MTPP recently expanded the definition of personal property to include privately owned pleasure boats. This change added a new category of personal property that must be supported by the system. To remain useful, the system must easily accommodate these frequent and often unpredictable types of changes.

Because of the need to support changes to the system, the design of the back-end interface is modular. Currently there are four program sets:
• a program set that produces tabular reports,
• a program set that produces bar charts,
• a program set that produces line charts, and
• a program set that produces pie charts.

Each function within a program set is a separate entity. For example, for each program set there is one program to support output type selection, and there is a separate program to support the selection of data for a report. Because these programs are designed as separate entities, it is easy to add programs to support additional functions. For example, MTPP could add a program that provides regression options without significantly affecting existing functions.

Generic Design

Another goal for the back-end interface was that it be generic enough to be used to produce a variety of reports. The interface must support

• the production of 36 report types,
• the production of 4 output types for each report type, and
• the selection/manipulation of data for each output type.

Variety of Report Types

Staff at MTPP currently produce different reports based on their work division. There are three divisions of work at MTPP: QA and Operations, Rates, and Management Support. Some staff assigned to work in the QA and Operations Division monitor carrier performance. They often produce reports that identify carriers who miss delivery dates, carriers who have been disqualified from the Personal Property Program, or carriers who damage shipments in transit. Staff who work in the Rates Division monitor the changes in carrier rates that were bid for moving and storing personal property. The staff who work in the Management Support Division monitor the overall program and produce reports for external agencies such as Congress and the Armed Services.

During the requirements analysis phase of WHIST-MOD, 36 report types that encompass the needs of these three work divisions were identified. For example, there is a Missed Required Delivery Date Report that is frequently produced by the QA and Operations staff. Rates staff need to produce a Change in Rate Levels Report. All three divisions produce a Number of Shipments Report and a Net Weight Shipped Report.

The prototype front-end Interface completed by ORNL in March 1990 retrieves data for each of these 36 report types. The back-end Interface is generic enough to accept the data retrieved by the front-end and produce output for each of the 36 report types.

A query in the ORACLE front-end selects data from the appropriate database table based on the parameters the user chooses. Although some of the data selected and the specific database tables used for the selection of data are dependent on the report type the user chooses, these data are stored in a single dataset file. Because the Interface is generic, this single dataset file can be used by the back-end Interface to produce any of these 36 report types.

Variety of Output Types

For each of these 36 report types, the staff at MTPP may need, depending on the purpose, to produce different output types. They need bar charts, line charts, and pie charts to represent trends in the Personal Property Program and tabular output to report detailed information. These output types may be used internally as an analysis tool or externally to provide information for other agencies. For example, if MTPP staff are analyzing trends in shipment weights for specific carriers, MTPP staff need the ability to quickly produce tabular or graphical reports.

Selection/Manipulation of Data

For each report type, staff at MTPP need the ability to produce reports that display different data. For example, users need to produce a Net Weight Shipped Report that shows the weight shipped for specific carriers for a particular year. They also need to produce the same report type that shows the weight shipped by particular codes of service for the same year.

Therefore, another goal of the design of the back-end Interface was to provide the ability for users to select different data columns from the dataset file created in the front-end Interface. The dataset file prototyped in the front-end Interface is created from a query of the database and contains a prescribed set of data columns, depending on the type of report the user has chosen to produce. From this set of data columns, the Interface provides the capability for the user to choose specific columns for a report type.

Not only do the users need to select data columns from the dataset file, but they also need the ability to manipulate these data. For a tabular report the users need to choose data columns for sorting the output. For example, they may need the tabular output sorted by codes of service, or by carriers, or by one of the other data columns. They also need to perform different calculations on these data. They need to select the particular data column to use for the calculation, and they need to be able to specify the calculation type. For example, the user may want to calculate an average on the weight column for a Net Weight Shipped Report. The back-end Interface is generic enough to support interactive sorting and the interactive selection of calculation variables. It also supports the following calculation types: average, percent, and total.

Ease of Use

MTPP staff also need an Interface that is easy to use because there is a high rate of staff turnover and they have disparate computer skill levels. And although some MTPP staff currently are experienced computer users, others have minimal computer experience. All MTPP staff must produce reports.

Another goal for the back-end Interface was to make it easy to use so that even inexperienced computer users can quickly and easily produce a variety of reports. The Interface was designed to protect the user from the sophistication and complexity of the application development software. Each screen allows users to select and manipulate data by simply pressing a key, and extensive interactive help messages are incorporated into the Interface.
THE SAS INTERFACE

The SAS/AF programs written for the WHIST-MOD prototype produce screens that allow the user to manipulate the data selected from the ORACLE front-end. The user may manipulate these data for reporting purposes by choosing a variety of options such as the following:

- the type of report or chart to be produced,
- the data columns to be included on a report/chart,
- the data column on which to perform arithmetic operations,
- the type of arithmetic operation to be performed, and
- the destination for the output.

Other options, depending on the type of report or chart the user has chosen to produce, may also be available. Currently there are four program sets: programs that produce tabular reports, programs that produce bar charts, programs that produce line charts, and programs that produce pie charts. Additional program sets that produce other charts could easily be added to the prototype.

Each of these program sets consists of two basic types of screens that the user will see: SAS/AF program screens and SAS function screens. The program screens may call SAS function screens. There are four types of SAS and SAS/AF display screens that have been used in the back-end prototype: the SAS TITLE window, the SAS LEGEND window, the SAS HELP window, and the SAS VARLIST window.

The Menu Program

There are an initial MENU.PROGRAM screen and four program sets in the WHIST-MOD back-end interface: the tabular report program set, the bar chart program set, the line chart program set, and the pie chart program set.

The MENU.PROGRAM screen allows the user to choose one of the four program sets (see Fig. 1). The tabular, bar, line, and pie program sets have, for the most part, unique SAS/AF programs. The horizontal and the vertical bars use the same program set. However, each program set provides similar functions (i.e., the ability to choose variables, choose a calculation option and a calculation variable, specify titles, and direct the output). The bar program set is used in this paper to illustrate the functionality that is common among the program sets.

The Bar Program Set

The bar program sets are called when the user specifies either horizontal or vertical bar on the MENU.PROGRAM screen. Figure 2 is a hierarchy chart of the programs that constitute the bar program set. The first screen the user will see is BAR.PROGRAM (see Fig. 3). This screen allows the user to call one of the following screens: BARDATA.PROGRAM, BOPTION.PROGRAM, the SAS TITLE window, or BOUTOPT.PROGRAM.

The BARDATA.PROGRAM screen (see Fig. 4) allows the user to choose column names from the SAS dataset file that contains data selected from the ORACLE front-end. The user may choose any column from the dataset file in a VARLIST window for the X axis variable and any numeric column from the dataset file in a VARLIST window for the calculation variable. Figure 5 shows the BARDATA.PROGRAM screen with a VARLIST window. These column names chosen by the user are stored in macrovariables and are used by subsequent programs. Figure 6 shows all the macrovariables that originate and are used in the bar program set. The BARDATA.PROGRAM will call TIME.PROGRAM if the user chooses a date column from the VARLIST window. It will also call the BRANGE.PROGRAM when the user chooses a numeric column from the VARLIST window. When the user exits this screen, he/she is returned to the BLOGAROUND.PROGRAM screen. The process of using SAS/AF VARLIST windows to choose column names, storing these choices in macrovariables, and calling subsequent programs depending on these choices is common to all the program sets.

The TIME.PROGRAM screen (see Fig. 7) allows the user to choose a date format for the bar chart. The user may choose to display dates on the bar chart by month, by year, by month and day, and by day, month, and year. The user's choice is stored in a macrovariable that is used by BARDATA.PROGRAM. The process of selecting date formats for reporting purposes is common to all the program sets.

The BRANGE.PROGRAM screen (see Fig. 8) allows the user to specify up to three minimum and maximum boundaries for data values for X-axis variables for the bar chart. For example, from this screen the user may choose to display three bars: one with shipment weights between 1,000 and 5,000 lbo.; one with shipment weights between 5,001 and 10,000 lbo.; and one with shipment weights between 10,001 and 100,000 lbo. The maximum and minimum boundary values are stored in macrovariables that are used by BARDATA.PROGRAM. The process of specifying boundaries for numeric values is common to all the program sets.

The BOPTION.PROGRAM screen (see Fig. 9) allows the user to select a calculation type (i.e., total, average, or percent). The user's choice is stored in a macrovariable that is used in BREADY.PROGRAM. The process of selecting a calculation type is common to all program sets.

The SAS TITLE window allows the user to specify titles for bar charts. This option is available in all the program sets.

The BOUTOPT.PROGRAM screen (see Fig. 10) allows the user to choose the destination for the output. The user may choose to send the bar chart to a printer, to an operating system file, or to the screen. The user's choice is stored in a macrovariable that is used in BREADY.PROGRAM. The process of selecting an output option is common to all the program sets.

The BREADY.PROGRAM generates a bar chart using the values in the macrovariables set by the user in the other bar programs. It sends the bar chart to the destination specified in BOUTOPT.PROGRAM. The user sees a screen displaying a message: "Processing ... one moment." Figure 11 shows an example of a bar chart generated using the SAS interface. The process of generating and directing the output is common to all the program sets.
CONCLUSION

The programs that constitute the bar program set and the other program sets can be used by MTPP staff to make choices that will produce an almost infinite variety of reports. We designed the prototype interface to meet the essential needs of our sponsors. Because the interface is modular, enhanced functionality can be provided easily by the addition of programs or program sets. Because the SAS/AF programs can be used with any data contained in the dataset file produced from the ORACLE front-end interface, it was not necessary to write separate programs for each of the 36 report types our sponsors needed. Instead, we wrote separate programs only for the different output options provided by the interface. Furthermore, this single interface can be used by users who know nothing about SAS programming commands and language to produce the output they need to perform their jobs.

ACKNOWLEDGEMENTS

The authors would like to thank Tai-Lun Chiang for all her work on the coding of significant pieces of the SAS interface.

CONTACT INFORMATION

Jane P. Loftis
Teresa L. James
Oak Ridge National Laboratory
P. O. Box 2008, 4500N, MS-6207
Oak Ridge, TN 37831-6207

SAS, SAS/AF, AND SAS/GRAPH are registered trademarks of SAS Institute, Inc., Cary, NC, USA. ORACLE is a registered trademark of Oracle Corporation.
Fig. 11. An example of a bar chart.