SOFTWARE INTERFACE BETWEEN SAS AND THE GDDM INTERACTIVE CHART UTILITY

WOLFGANG SCHREINER
Second Department of Surgery, University of Vienna, Austria
(Received 30 June 1988; in revised form 21 October 1988; received for production 28 November 1988)

Abstract—Data from temporary or permanent SAS-data-sets on disk are transferred to the GDDM Interactive Chart Utility (ICU). Within the ICU these data can be displayed graphically, including all features of modification, supported by the ICU. The transfer is accomplished by a user written SAS procedure, which transforms the data from SAS internal format to the structure required to call the ICU from a PL/1 program. The interface provides interactive menu driven graphics from SAS-data-sets without the need to use SAS control language. The implementation refers to the CMS operating system.

Software interface    Interactive graphics    Statistical analysis system    GDDM

1. INTRODUCTION

SAS (Statistical Analysis System) is known as one of the most powerful integrated packages for data input, data manipulation, statistical analysis and graphical display [1–5]. The respective parts of the SAS package access data stored in SAS-data-sets and can be invoked either interactively or via SAS programs submitted to SAS as complete sequences of statements in SAS control language. Although the SAS control language is a very potent one and provides very efficient ready made procedures, it is still necessary for the user to (a) know editor commands in order to write the control statements onto an SAS program file and to (b) become familiar with SAS syntax, procedures and options.

These requirements are experienced prohibitive, when clinicians think about writing and running their own SAS jobs beside clinical routine work. This is especially true for sophisticated graphics including (a) non-standard labelling of axis, (b) additional remarks and comments within the graphics, (c) special patterns filling the area between curves, (d) individual symbols, interpolation and smoothing for line graphs and (e) bar charts with groups and/or subgroups.

Although SAS provides means for all that (AXIS statement, ANNOTATE data sets, coordinate-wise positioning of text and symbols, the AREAS option, SYMBOL and PATTERN statements), the usage may become increasingly cumbersome if very special attributes are necessary.

On the other hand, the ICU (GDDM Interactive Chart Utility, IBM Corporation [6]) is a very efficient menu driven package for graphical display and renders any kind of programming on the part of the user unnecessary. However, data to be displayed have to be typed in via the keyboard or must be read from formatted output lists generated by some other application program. Moreover, it is impossible to perform any kind of numerical analysis on the data within the ICU.

For the reasons mentioned above it seemed desirable to set up an interface between SAS data sets and the ICU. Data acquired and modified via FSP (Full Screen Product) as well as SAS data sets which are the output of some SAS statistical procedure may then be transferred to the ICU for convenient graphical display.

The present work describes the implementation of such an interface under the operating system CMS on an IBM VM/370 4381 mainframe.
2. THE DESIGN OF THE INTERFACE

The interface, named 'PROC SASICU', is in fact a user-written SAS procedure designed along the lines specified in the SAS Programmer's Guide [7] and may be called within SAS programs in the same way as regular SAS procedures. Like all SAS procedures, PROC SASICU consists of two main parts, (a) grammar module (checking the invocation syntax) and (b) calculation module (performing operations on the data).

For PROC SASICU an individual grammar (defining the invocation syntax) has been specified and implemented in a corresponding module by help of the 'grammar processor' supplied by SAS for the setup of user written procedures [7]. Following mnemonic definition of the grammar according to the specifications of SAS-grammar-files [7], the grammar processor is invoked to produce the source code of a PL/1 procedure (grammar module) which is compiled and finally linked with the calculation module.

The calculation module proceeds as follows to actually transform the data from SAS-data-sets to the ICU.

(i) The invocation clauses for SASICU (see Section 3) are parsed and checked for syntax errors (by calling the grammar module). The informations drawn from parsing (names and types of variables, name of the SAS-data-set to be accessed) are reported in the SASLOG file.

(ii) According to the information retrieved by parsing the invocation clause, storage is dynamically allocated to hold data from SAS-records. The number of observations in the SAS-data-set is determined and PL/1 arrays ("GDMM-arrays", according to the format necessary for GDMM, [8]) are setup for the variables specified in the invocation clause. There are separate arrays for data and labels of variables in GDMM.

(iii) The values of the variables are retrieved from the SAS-data-set (observation by observation) and assigned to the GDMM-data-arrays. The names of the SAS-variables are assigned to the GDMM-label-arrays.

(iv) The procedure "CHART" is called from the GDMM-library with the arrays for data and labels as parameters in order to invoke the ICU with the data and variable names taken from the SAS-data-set.

3. INVOCATION OF SASICU

3.1. Invocation syntax

In order to invoke SASICU the following statements have to be included in an ordinary SAS program, clauses in [square brackets] being optional.

PROC SASICU [DATA=dataset];
[VAR varlist ;]

The clause 'DATA=dataset' specifies a temporary or permanent SAS-data-set on disk. If 'DATA=dataset' is omitted, the data set most recently modified is used, according to SAS conventions. If the data set specified does not exist, the procedure terminates with an appropriate message on the SASLOG.

In the clause 'VAR varlist' one can specify which of the variables in the data set 'dataset' shall be transferred to the ICU. The first variable appearing in the 'varlist' is considered independent (GDMM X-axes), all others are considered dependent (Y1 to Yn). The data are assumed to be tied, i.e. each value of X is common to a set of corresponding values Y1 to Yn. If the 'VAR varlist' clause is omitted, all variables are transferred following the internal order of variables in the SAS-data-set. 'varlist' is a variables' list following SAS conventions.

A 'BY' statement may be specified but will be ignored.

The following example depicts a simple SAS-job which sets up a temporary SAS-data-set which is transferred to the ICU.
Example 1
DATA ONE;
INPUT X Y Z;
CARDS;
  1 4 6
  3 7 9
  5 6 4
;
PROC SASICU DATA=ONE;
  VAR X Y Z;

Since SASICU respects the order of variables within the VAR-clause, Y instead of X may be specified as the independent variable, see example 2.

Example 2
DATA ONE;
INPUT X Y Z;
CARDS;
  1 4 6
  3 7 9
  5 6 4
;
PROC SASICU DATA=ONE;
  VAR Y X Z;

Only some of the variables (e.g. Y, W, R) on the SAS-data-set may be selected for transfer; Y will be plotted as independent variable, see example 3.

Example 3
DATA TWO;
INPUT X Y Z W R;
CARDS;
  1 4 6 8 4
  3 7 9 3 5
  5 6 4 2 8
;
PROC SASICU DATA=TWO;
  VAR Y W R;

3.2. Character variables and missing values

Missing values of numeric variables are set to missing (1E72) values in the call of the ICU.

For dependent variables (Y1 to Yn) a missing value results in a missing marker (e.g. star, bar, line, sector) for that particular variable at the corresponding value of X when the ICU chart is displayed. All other (nonmissing) values for that dependent variable will be displayed, however. If a dependent variable is defined in the SAS-data-set but all its values are missing, the variable name will be the only thing to appear as a label in the ICU graphics.

If a value of the independent variable is missing, all corresponding dependent variables (Y1(Xmiss) to Yn(Xmiss)) will not be displayed due to the tied structure of data. If all values are missing the "empty chart" will be displayed, the values of all dependent variables being passed to the ICU, however. Interactively, values for the independent variable (numeric or character) may be entered via the panel for data modification.

A dependent variable defined as character in the SAS-data-set is transferred as if all its values were missing.

If the variable specified independent (first position in the VAR-clause) is defined as
character in the SAS-data-set the observation number (record number of SAS-data-set) is assigned as a default numerical value. Following invocation of the ICU the user may add labels to each value of \( X \) in the panel for interactive data modification. Appropriate messages are output on the SASLOG.

The following example summarizes the performance of SASICU in the presence of missing values and character variables.

**Example 4**

```sas
DATA THREE.PERM;
INPUT X $ 1 Y Z W R;
CARDS;
A 4 6 8 .
B 7 9 . .
C 6 4 2 . ;
PROC SASICU DATA=THREE.PERM;
VAR X Y W R;
```

Since \( X \) is character in SAS, the observation numbers (1, 2, 3) will appear on the horizontal axes of the ICU-graph. \( Y \) is displayed as \( Y(1)=4, Y(2)=7 \) and \( Y(3)=6 \). For \( W \) only \( W(1)=8 \) and \( W(3)=2 \) will appear, while for \( R \) only the label is displayed.

### 4. CONCLUSION AND APPLICATIONS

Since SAS control language is not simple enough for every day use by clinicians, repeated programming support had to be given for the major part of graphical data processing within numerous research projects at our clinic. Very specific graphical presentation (labelling, text within graphics, etc.) is essential for congress presentations but takes a lot of time when specified e.g. with SAS-annotate. On the other hand, the ICU proved an optimal tool for clinicians setting up their own presentation graphics. With the development of SASICU clinicians are now able to get their data from SAS to the ICU and design their graphics themselves according to personal needs and preference. The SAS support for clinicians has thus become much more efficient and can now be restricted to numerical and statistical analysis, rendering the programming of numerous presentation graphics unnecessary. The usual procedure is as follows.

Data to be displayed can either be raw SAS-data-sets (as created by SAS-FSP) or output files from statistical procedures (e.g. PROC MEANS). In a short SAS-job, similar to the examples above, the ICU is invoked. The only action to be taken in this session is to store the data in GDDVM format via the appropriate interactive panel and to exit. In a later session the data may be reloaded into the ICU, which is then directly invoked from the operating system CMS (no more transfer of data). All possibilities of interactive modification of data and format of display are then open to the user.

### 5. REFERENCE INSTALLATION

The Institute for Medical Informatics, University of Vienna, has developed and implemented a menu-driven user surface for the invocation of some frequently used SAS procedures (means, group comparisons, frequencies, analysis of variance, etc.). About 30 clinics and research institutions within the Vienna General Hospital are currently using SAS via this menu driven surface. Up to now a major drawback of the whole approach was the fact that the menu could support SAS graphics only to a very limited extent, since the numerous graphical options are much more difficult to support than the parameters necessary to invoke numerical SAS procedures.

As opposed to this, PROC SASICU offers a very simple invocation syntax (see above) and could therefore easily be included in the menu driven surface. Since then, PROC SASICU is widely used not only by physicians (on the menu level) but also by software experts (who know how to code SAS graphics) since PROC SASICU, called from any SAS
program, is more convenient than, e.g. the SAS procedures PROC GPLOT, PROC GCHART, etc.

**SUMMARY**

This work describes a software interface which enables users of the statistical package SAS to have interactive graphical display of their data stored in SAS internal format. For a wide range of applications, the use of the SAS graphic procedures GPLOT and GCHART can be replaced by running the interface in a standardized way and then performing all graphical modifications within the IBM GDDM Interactive Chart Utility.

The use of SAS procedures requires programming in SAS control language, and the user has to remember (or to look up) a wide set of options in order to fully exploit the power and versatility of SAS graphics. Moreover, the effect of any option added to the SAS program cannot be visualized "on the spot" but rather needs another run of the program. All in all, it can become very tedious to prepare special graphic effects (such as positioning pointers or special symbols in a chart).

Using the interface only requires a very simple call of PROC SASICU which is similar for any type of chart and does not include the specification of options except the names of variables whose values are to be displayed in the graphics. Once the data have been transferred to the Interactive Chart Utility, all options of graphical display are specified interactively; type of chart, colors, markers, sorting, line types and interpolation, shading patterns, zooming, notes, etc.

In clinical practice, the interface proved very effective, since physicians may prepare their presentation material without the assistance of computer experts. In fact, the superior power and versatility of SAS is retained for advanced statistical evaluations (which at any rate will be conducted by experts in data analysis) as long as data are stored in SAS format. On the other hand, by using the interface, the user benefits from the most convenient and easy-to-handle mode of operation offered by the ICU.

**REFERENCES**


*About the Author—Wolfgang Schreiner* was born in 1954 in Vienna, studied physics and mathematics and received the Ph.D. at Vienna University. He then worked as a research fellow at the University of London and subsequently for 4 years as assistant at the Vienna Institute for Experimental Physics on the computer simulation of thermodynamic model systems. Parallel to this he worked as coauthor on a monography in the field of nuclear medicine (estimation of internal radiation dose). Since 1984 he is running the section of computerized data analysis at the 2nd Department of Surgery, University of Vienna.