TRANSFERRING A MAINFRAME DATA-CHECKING APPLICATION TO THE SAS* SYSTEM FOR PERSONAL COMPUTERS: THE AGONY AND ECSTASY

Dorothea Collins, West Haven Cooperative Studies Program
Joan Derrico, West Haven Cooperative Studies Program

ABSTRACT

The emergence of the SAS* System for personal computers generated enthusiasm for setting up the data processing for a new Cooperative Studies Program clinical trial in a microcomputer environment. Part of this involved transferring a generalized VS FORTRAN datachecking routine from a large-scale mainframe to a microcomputer using the SAS System for personal computers under DOS. This program is part of a larger data management procedure used for cooperative clinical trials. It was immediately apparent that the mainframe routines would require major redesigning in order to operate under the SAS System for personal computers and that our approach would require different methods than we would have used for the mainframe SAS System. This paper will discuss the redesign that was necessary to handle the table-driven error checking. An evaluation of the microcomputer versus mainframe SAS System for such an application will be presented. Included will be considerations used for arrays, macros, date handling and message generation. Execution times will be presented comparing the PC version with an uploaded mainframe version of the program. This paper should be useful for anyone planning a similar transfer of the SAS System, PL/I or FORTRAN programs, or those generating comprehensive code under the SAS System for personal computers.

INTRODUCTION

The Veterans Administration Cooperative Studies Program Coordinating Center (CSPCC) in West Haven, Connecticut is one of four centers that manages the statistical, data processing, and administrative needs for a wide variety of large-scale multi-center controlled clinical trials. At the present time, the West Haven CSPCC coordinates studies in AIDS, other infectious disease, cardiology, dentistry and gastrointestinal diseases.

The Center utilizes the computing facilities of the Yale University Computing Center for timesharing including mainframe IBM 4341 and 4361 computers. The mainframe SAS System has been available on these machines for several years and is widely used by the staff of CSPCC for a variety of studies. The advent of the SAS System for personal computers prompted CSPCC to consider transferring some of our computing onto the microcomputer to reduce timesharing costs. When a new study involving Percutaneous Transluminal Coronary Angioplasty (PTCA) was assigned to the Center, it was decided to attempt to manage the entire study on the microcomputer. Since all of the staff was familiar with the mainframe SAS System, we thought that most of the routines could be written in the SAS System for personal computers.

BACKGROUND

The CSPCC has in place a data management system that is utilized in all of our clinical trials. The Center receives and processes thousands of computerized forms, either paper or transmitted. Once the data is loaded onto the computer, every variable is checked for outliers and erroneous data. If an error is found, a computer notice is generated to be sent to the participating hospital staff for resolution. At the same time, a computer record is generated to keep track of the type of error, to use as a basis to edit the data file if necessary, and to follow up unresolved data errors if a certain period of time elapses before a resolution.

This data-checking program is written in mainframe VS FORTRAN and utilizes table-driven techniques to process the data file. A set of control records specifying information on every variable in the study is read in by the program and stored. The new incoming data is then checked against this table of information to see if the data falls within the correct value ranges. The program is generalized in that the program is utilized for all our studies.

Once the data problem is resolved and the corrected notice received at CSPCC, an edit program is executed to change the data files for the study. Then the edit record is stored separately to keep track of the types of errors that are occurring in the clinical trial. This also provides a historical record of all data changes for the study.

The data-checking routine was chosen as the vehicle for our first major experimentation with the SAS System for personal computers. This program would have to be a generalized table-driven routine written in the SAS System for personal computers and able to run on the IBM PC-AT and IBM PS/2.
(both with 640K) microcomputers available at CSPCC. Once developed, the program would also be uploaded for use in our mainframe-oriented studies.

**DESIGN CONSIDERATIONS**

Our plan was to develop the PC SAS code along the same design and flow of the FORTRAN routines. That is, to read in the control records into multi-dimensional arrays and make use of a series of loops using imbedded macros to proceed through the data checking on each individual new input data record. Our first attempt to do this resulted in memory requirements too large for our hardware configuration to handle. We referred to a paper presented at last year's SUGI 12 conference by Davis and Sweetland to optimize efficiency and performance of the SAS System for personal computers under PC DOS. This included using a math co-processor and running in the batch instead of interactively. But in spite of all efforts, we could not fit the old design on the microcomputer. We realized that what we needed was a new design.

Having realized that the macros were using a large amount of memory, we eliminated nesting and decided to stay with a top-down design. We also decided to replace the multi-dimensional arrays with single-dimensional arrays and utilize sorting techniques. Input control records would be sorted by type of form and record number. Input data to be checked would also be sorted in this way. This would allow the control records to be read in by sections depending upon form number and the whole control record file would not have to be stored at once.

The program was divided into two parts. The first section performs the actual reading in of the control records, checking for outliers and generating a dataset containing a record for each error that was found. The second section sorts and prints the generated notices. In order for these notices to be generated by hospital number and patient number, a sorting step was added between the two major steps of the program.

In addition, the control record file was required to be sorted by form and card number. In this way, the order of the control record file and the input data file were synchronized so that they both could be read in simultaneously, compared, and matched. This resulted in a marked decrease in the required storage to run the program.

Several features available in the SAS System for personal computers very much eased the designing process. First, the date functions that involved many lines of FORTRAN code in the previous program, were condensed to one line using the MDY function. Also, with the previous program, dates were checked using three separate control cards for month, day, and year. The new program handles the checking using just one control card. The new program checks for an entire valid date whereas the FORTRAN routine checked for valid ranges of month, day, and year. This resulted in a date like '023187' being considered valid erroneously.

The utilization of the ROUND function also enabled the program to read in what type of checks that the data checking program would perform from the control records as one field rather than four separate fields. This reduced the amount of SAS variables necessary and thus further reduced the amount of memory necessary to run the program.

Another feature that was part of the FORTRAN code included saving variables in both number and character forms. This was used to check for valid ranges and also to have an exact specification of what was coded in character format. With the automatic transformation available in the SAS System, this was unnecessary to do in the microcomputer version. This would have doubled the number of variables and memory requirements would have been a problem.

**COMPARISONS BETWEEN MICRO AND MAINFRAME VERSIONS**

The resulting program can be run on either a microcomputer with the SAS for personal computers under DOS or on the mainframe with the SAS System. We have replaced the FORTRAN version of the program. We were interested in running the program with various amounts of input records on both machine types to compare the run times.

Runs were made with 250, 500, and 700 input records on the PC-AT, PS/2 Model 80, and the mainframe. There was a considerable improvement in speed between the PC-AT and PS/2 Model 80 test runs with the Model 80 running approximately three times faster.

**FUTURE ENHANCEMENTS**

The existing program can be used for other studies or for any similar data-checking operation with minimal changes. The specifications for the arrays have to be modified from application to application because they are explicitly subscripted. However, if the SAS Institute implements
the varying array length using _NUMERIC_,
the code could be completely generalized.

In the near future, we plan to have the
complete data management process for this
study on the microcomputer using the SAS
System for personal computers. We expect
this task to go more smoothly than our
first attempt because we have learned the
types of design approaches that are
necessary to develop comprehensive code.
Future programs will include an editing
routine that will process the filled in
edit records once the data discrepancies
have been resolved. Several routines will
also be added that check data errors across
fields and from record to record.

We are also looking forward to a version of
the SAS System for personal computers that
allows for the use of extended memory and
provides a better accounting of memory
utilized.

CONCLUSION

In conclusion, our experience in the
development of generalized code with the
SAS System for personal computers taught us
to rethink our approaches to program
design. We learned to use several steps
instead of one large one to perform the
same functions in the confines of the
existing memory on the PC. We learned to
use sorting techniques to ease the burden
on macro and array processing. The
resulting program was a more
state-of-the-art top-down routine. Future
changes and corrections to code will be
easier to make as a result. In other
words, the SAS System for personal
computers is a different 'animal' than
mainframe SAS, but one that can be managed
with the right approach.

Dorothea Collins
Joan Derrico
West Haven CSPCC 151-A
West Haven VAMC
West Spring Street
West Haven, CT 06516
(203) 932-5711 (X500)

*SAS is a registered trademark of SAS
Institute, Inc., Cary, NC, USA.