Data base concepts applied to generalized programming packages

by G. CORT STEINHORST and BARRY L. BATEMAN

Texas Tech University
Lubbock, Texas

and

DANIEL L. CURTIS

University of Southwestern Louisiana
Lafayette, Indiana

INTRODUCTION

Software “packages” are becoming more and more prevalent in both use and availability. These packages provide many useful functions, without the costly, time-consuming development and testing normally required to implement the desired operations. A disadvantage of many packages, however, is the difficulty in modifying them in order to receive the desired result, if the original resultant of the package is not totally satisfactory.

TYPICAL TECHNIQUES

Various approaches have been used in package implementation. A common type of package is a series of subroutines (e.g., the Scientific Subroutine Package provided by most major computer manufacturers). This type of package provides great assistance to the experienced programmer but is of little use to the non-computer-oriented researcher. The experienced user can easily modify these routines to meet his particular qualifications. The inexperienced user must normally “make do” with his output, must seek technical assistance, or, in extreme cases, must forgo the use of the computer in his research.

Another type of package consists of a series of separate programs (e.g., The Biomedical Programs). These programs are generally very flexible. A disadvantage is that many times several of these programs must be run in sequence in order to obtain the desired results. Normally, this requires many similar calculations, which are repeated with each execution of the separate programs. There is little possibility of executing the entire set of calculations in one step. The usual procedure is to execute one program, check the results, execute the next program, check the results, and so on until the final answers are obtained. This process requires a relatively large block of time and also requires understanding the function of several programs in order to achieve the final results.

A BETTER APPROACH

For the average researcher, with little computer knowledge, packages can be written which employ techniques developed in data base management systems. The main advantages to this approach are (1) a simple, user-oriented “language” can be developed to direct the processes; (2) all input data and intermediate calculations can be stored in a common, “data base” area; (3) modular programming can be used to allow easy modification of the package without extensive reprogramming.

The user “language” need not be a very sophisticated one. The basic purpose of this language is to provide easy direction of processes by non-computer-oriented researchers. A very likely possibility for the language would be the use of the names of the particular calculations desired, followed by the data, in free form, which is to be used in the calculations. By storing the input data and the intermediate calculations in an area common to all routines, it is possible to perform a complex series of operations upon the data with little redundant calculation. In outline form this type of package consists of (1) an executive, language-interpreter module, (2) a common data base, and (3) a series of operation modules and sub-modules, which operate on the data base.

The executive program determines the sequence of operations and directs the order of calls to each of the operation routines (modules). The data base provides an easy method of determining if an intermediate operation has been performed and also provides a common area for communication among the various modules and sub-modules of the package.

APPLICATION OF DATA BASE CONCEPTS

After considering the problems related to the so-called “packages,” an attempt has been made to design an improved package. An example of this technique is STATPAK (Statistical Package) and was implemented on the RCA Spectra 70/45. STATPAK is designed with the intentions
of: (1) Providing the user a package which does not require the knowledge of programming; (2) Eliminating duplication of input data; and (3) Providing a common area to store the knowledge of the subprograms. The executive program acts as both the communicator between the user and the subprograms and the major statistical subprograms included in the second segment of the STATPAK’s input subprogram, a generalized input subprogram. This subprogram allows free format of data. In other words, the user need not have his input data in a particular form. Another feature of the input subprogram is that there are virtually no limits placed on the amount of data points or the number of variables. Most programs or subprograms have a limit to the number of data points or the number of variables because the data is usually maintained in a rectangular matrix of fixed dimensions within the program or subprogram. In these programs or subprograms, a matrix dimensioned n by m will allow a maximum of n data points of m variables or vice versa. In STATPAK’s input subprogram, a single-dimensioned array of n times m locations is used. Thus, if there are k variables, the number of allowable data points is \([n \times m / k]\) where \([\cdot]\) implies the largest integer such that \(k \times [n \times m / k] \leq n \times m\). For example, if an array contains 1000 locations and the number of variables is 30 then the number of allowable data points is \(1000 / 30\), which equals 33.

**STORAGE OF DATA**

In addition to allowing a varying number of data points or variables within the array, the input subprogram writes the array to disk once the array is filled. The input subprogram then places a “1” in an indicator in the common storage area; thus, each statistical subprogram can determine if the data are in common storage by checking an indicator. If the indicator contains a zero, the statistical subprogram retrieves the information stored on the disk. The technique
of storing the data, allowing the data to be transferred to disk, and free format of data make the input requirements of the STATPAK system very flexible.

EXECUTIVE PROGRAM

To provide the user with a package which does not require a knowledge of programming, an executive program was written which uses a series of control cards and performs the functions indicated by the code on each card. In addition to providing ease of use, the executive program provides the user with other desirable features. The user may reference several statistical subprograms for one set of data; thus, eliminating the need of duplicating common input requirements of each subprogram. Also, the user may execute a specified series of statistical subprograms for any number of data sets.

The executive program reads in the control cards which constitute one job, and performs the job X times using the X sets of data that follow the END JOB card. This process is continued until a STATPAK END card is encountered, which terminates the system. The executive program also has the function of reinitializing the array of indicators of the statistical subprograms for each execution of a job or for the execution of a different job. Since each execution of a job or the execution of a different job requires a new set of data, the results stored in the common storage area are for the previous job iteration or previous execution of a different job; therefore, it is necessary for all calculations to be repeated.

The executive program has a list of the contained statistical subprograms and the corresponding codes for each. As each control card following the JOB card is interrogated, the code is matched to the desired statistical subprogram and control is transferred to that subprogram.

CONCLUSIONS

Generalized software packages can be implemented which will provide valuable assistance to the researcher. By using data base technology, it is possible to create a flexible organization which allows maximum computer usage with a minimum of technical ability. Simple semantic capability along with good error messages augment the generalized approach.

REFERENCES
