3. COMPUTERS AND THE STATISTICAL REVOLUTION

Chaired by Gordon Bear, *Ramapo College*

Extensible programming: Beyond reusable objects

DAVID E. ANDERSON

Allegheny College, Meadville, Pennsylvania

This paper describes the rapid development of a set of 30 custom-designed statistical routines that make use of the special features of the NEXTSTEP operating and development environment. The speed and flexibility of this effort was made possible because of NEXTSTEP's object-oriented approach to programming, including the ability to dynamically load new objects as a program is running.

I recently showed a faculty member a new drawing program that I had purchased. She made an often heard comment, "It would be really great if only it had _____" (fill in the blank with your feature of choice). One of the problems with commercial software is that users, especially faculty, often want a program to "do it my way." The solution, although not practical in many cases, is to design and write custom software.

However, as anyone who has done much custom programming knows, you often have the feeling that you are reinventing the wheel—that is, you find yourself writing code for menus, input/output, and so on, over and over again. The use of reusable objects is one step down a path to reduce the need to rewrite code. Extensible programming, especially when it is paired with the ability to load new objects while an application is running, further reduces the need to rewrite commonly used routines.

This paper describes the rapid development of a set of more than 30 custom designed statistical routines (StatTools) that make use of the extensible programming capability of the NEXTSTEP¹ development environment. StatTools uses a locally written software framework called Workbench. Workbench not only provides for the dynamic loading of "tools" during program execution, but it also provides the necessary input and output capabilities that the tools need.

Extensible Programming

One way to avoid having to provide for every possible option in a programming project is to make the program extensible—that is, write the program so that it is possi-

ble to add features without rewriting the source code for the main module. Think of a CD player as an extensible environment. You don't buy a stereo with 10 albums built in; you buy a CD player and some CDs to go with it. When a new CD is released, you don't buy a new CD player to listen to it. One type of CD player suffices whether you listen to classical, country, or rock 'n' roll. It is extensible!

Reusing the products of the software development process as a way to reduce software costs and to make programmers and designers more efficient is not new. However, the NEXTSTEP object-oriented development environment makes the creation and implementation of such programs quite straightforward. Object-oriented programming permits the reuse of designs as well as programs. As Johnson and Russo (1991) noted, there are at least two techniques for reusing program components: abstract classes that reuse components and frameworks that reuse designs for entire applications.

An abstract class is a design for a single object, whereas a framework is the design for a set of objects that collaborate to carry out a set of tasks. Thus, frameworks are larger scale designs than are abstract classes. Frameworks usually provide the design of only part of a program, such as its user interface, though application-specific frameworks sometimes describe an entire program.

Johnson and Russo (1991) use the term *ensemble* to refer to an instantiation of a framework (i.e., a set of objects working together to carry out a set of tasks). They note that the first widely used framework was the Smalltalk-80 user interface framework model/view/controller (MVC) that appeared in 1980 (Goldberg, 1984). MVC showed that object-oriented programming was well suited for implementing graphical user interfaces.

MVC has spawned many other user interface frameworks. MacApp is a popular commercial user interface

This work was supported by National Science Foundation Grant USE-9250625. Correspondence should be addressed to D. E. Anderson, Department of Psychology, Allegheny College, AC Box 39, Meadville, PA 16335 (e-mail: danders@alleg.edu).

framework designed specifically for implementing Macintosh applications (Schmucker, 1986). Recently, there have been several user interface frameworks from universities, such as the Andrew Toolkit from Carnegie Mellon University (Palay et al., 1988), InterViews from Stanford (Linton, Vlissides, & Calder, 1989) and ET++ from the University of Zurich (Weinand, Gamma, & Matry, 1989).

The NEXTSTEP development environment not only provides all the advantages of object-oriented design but adds the ability to develop generalized framework applications that can then be easily customized for specific uses. The NeXT Application Kit and the associated Objective C run-time environment provide all the tools that are needed to create extensible applications. Workbench, and its associated StatTools, is an example of such a "framework application" that can then be customized by writing specialized "tools" as needed.

Workbench

Workbench actually consists of three components: the Workbench application itself, a ToolBuilder application, and a set of objects to facilitate tool development. The application provides the interface for the user, who then loads and uses the tools that he or she needs. The ToolBuilder application generates a framework (including the necessary C code and a blank inspector window), thus relieving the developer of the need to redesign the connections with each new tool. Finally, the Workbench environment includes a set of objects that can be used by the developer to simplify writing new tools. Two of the most powerful objects are Mathematica Connector and Improv Connector (Improv is a spreadsheet). Both provide simple, one-line command access to very useful commercial software. For example, in the tools described here, the Improv spreadsheet was used for data input and, in several cases (e.g., transformation of data), for

Figure 1. Workbench application with the Bar Chart Tool active.

data output. Specialized objects used in the project include DataCounter, Histogram, and RandomData.

StatTools

StatTools is a set of procedures for common statistical methods that may be used for the analysis of data, as well as for generating random data from specified populations for teaching purposes. StatTools is not a "statistical package" in the usual sense of that term. Rather, the tools provide a way to summarize and analyze data that have been entered and stored using a spreadsheet.

The advantage of this approach is that the user need not learn any new techniques for entering, reviewing, storing, and retrieving data beyond those learned to use the spreadsheet. In addition, of course, the user has available all of the features of the spreadsheet, including some powerful graphic capabilities.

The more than 30 tools currently available cover most of the common statistical procedures (i.e., descriptive statistics, correlation and regression, Student's *t*, and analysis of variance), data manipulation and transformation, nonparametric tests, as well as tools for doing some simple graphs (e.g., bar charts, histograms, box-and-whisker plots, etc.). A DataGeneration tool provides for eight design types ranging from one group to multivariate experiments.

An example of both Workbench and StatTools can be seen in Figure 1. The top of Workbench includes a "Tool Shelf" where icons representing specific tools can be stored. The user simply "drags and drops" as many tools as might be useful for a particular session. In this case, the tools represent those that might be used in some initial data exploration, including Bar Chart, Histogram, Descriptive Statistics, and Scatter Plot tools (from the left of the ToolShelf). The Bar Chart tool has been selected, and the inspector for that tool can be seen on the right. The inspector allows the user to select any necessary options or provide additional information that might be needed to complete the analysis. A single column of data has been copied from the spreadsheet to the Input window since that is what is needed by the Bar Chart tool. (It is not really necessary to enter the data directly into Workbench since the tool can just as easily read data directly from the spreadsheet.) After the appropriate options have been selected in the inspector and the axis labels entered, the user clicks "Apply Tool" to initiate the process. In this case, numerical output is sent to the "Output" window and a separate window is created with the

graph. All the output, including the graph, can be printed or moved via "copy/paste" to another document.

Advantages of Extensible Programming

The primary advantages of this approach to programming flow from its flexibility, for both the developer and the user. Once the framework for the projects has been developed (in this case, Workbench), writing associated modules becomes quite simple and straightforward. Because so much of the code is hidden (the developer writing a tool never sees the Workbench code), updates, changes, and maintenance of the software are greatly simplified.

The user is presented with a flexible interface that not only builds on prior knowledge (the spreadsheet) but also provides an environment that encourages thoughtful problem solving. To try something new, all the user need do is drag and drop a new tool to Workbench and apply it to the data. And finally, where appropriate, the output of one analysis can easily be used as the input for another.

REFERENCES

- GOLDBERG, A. (1984). *Smalltalk-80: The interactive programming environment*. Reading, MA: Addison-Wesley.
- JOHNSON, R. E., & RUSSO, V. F. (1991). *Reusing object-oriented designs* (Tech. Rep. No. UIUCDS91-1696). West Lafayette, IN: Purdue University, Department of Computer Science.
- LINTON, M. A., VLISSIDES, J. M., & CALDER, P. R. (1989). Composing user interfaces with InterViews. *Computer*, 22, 8-22.
- PALAY, A. J., HANSEN, W. J., KAZAR, M. L., SHERMAN, M., WADLOW, M. G., NEUENDORFFER, T. P., STERN, Z., BADER, M., & PETRE, T. (1988, January). *The Andrew Toolkit: An overview*. Paper presented at the USENIX Association Winter Conference, Dallas, TX.
- SCHMUCKER, K. J. (1986). *Object-oriented programming for the Macintosh*. New York: Hayden.
- WEINAND, A., GAMMA, E., & MATRY, R. (1989). Design and implementation of ET++, a seamless object-oriented application framework. *Structured Programming*, 10, 63-87.

NOTE

1. The NEXTSTEP operating system and associated development tools are available for a variety of hardware platforms, including those using Intel chips (486 and Pentium), as well as workstations from Hewlett Packard (PA-RISC workstations, Model 712/60 and above) and Sun Microsystems (SPARC Station 5 and above). Information about the software can be obtained from NeXT Computer, Inc., 900 Chesapeake Drive, Redwood City, CA 94063.

(Manuscript received November 1, 1994;
revision accepted for publication January 19, 1995.)