TAMING THE UNIX END-USER INTERFACE

Stephen M. Libster,* Internal Revenue Service

KEY WORDS: UNIX, Interfaces, Statistics, SPSS

OVERVIEW

The purpose of this paper is threefold. First, the author introduces UNIX[1]/ and surveys the availability of statistical packages under this operating system. Second, the range of UNIX end-user interfaces encompassing command lines, graphics windows and text-based instructions is surveyed. Third, the paper summarizes the author's own implementation of a UNIX text-based interface within the Statistics of Income Division (SOI) of the Internal Revenue Service. This interface has been in operation since January 1989 and has a user base of 25 SOI employees. These employees are mostly economists, systems analysts and statisticians. The design of the SOI interface should be of particular interest to computer professionals who must assist nonprogrammers with the migration of applications to a UNIX environment. The paper also outlines some built-in utilities for users of the SPSSX statistical package.

UNIX AND STATISTICAL USERS

UNIX is an operating system which is widely available on computers ranging in size from personal computers to mainframes. Much of its recent popularity is attributable to the relative ease with which applications can be ported to different UNIX-based computers. UNIX permits multiple users to access the same computer simultaneously. Moreover, it enables any single user to run several jobs at the same time, a capability known as "multi-tasking."

Because UNIX was originally a product of academic and research environments, the availability of commercial software has been limited. [2]/ The author recently conducted a brief phone survey of statistical packages to determine their availability under UNIX. Figure 1, below, summarizes these findings.

GOALS OF THE END-USER INTERFACE

The end-user interface governs the way in which nonprogrammers interact with the computer. The quality of the interface is especially critical at the introduction of a new computer system to get applications up and running quickly. However, even experienced users can often benefit from a well-designed interface. Figure 2 describes some goals which an end-user interface should strive to achieve.

Among these goals, the first three warrant some additional comments. The first goal advocates the trapping of potential errors before they reach the operating system and the safeguarding of files. Many operating systems, UNIX included, do little to prevent an inexperienced user from making disastrous errors. A classic case in point is the "format" command in PCDOS. This command

Figure 1. Author's Informal Survey of Statistical Software Vendors for UNIX Computers (Conducted April, 1989)

Vendor	UNIX Versions Offered
BMDP	Callan, Harris*, DEC VAX, MASSCOMP, Pyramid, Sun.
S-PLUS	AT&T 3B2, Apollo, HP9000, IBM RT, Apple Mac II, Sun3, Sun4, Sun 386i, DEC VAXStation.
SAS	HP9000, Sun3, Sun4, Sun 386i*, Apollo*.
SPSS	Arix, AT&T 3B2/2B15, DEC VAX, HP9000, NCR Tower, Harris, IBM RTPC, Pyramid, Sequent Symmetry, Sun3, Sun4, Sun 386i, UNISYS 5000/6000/7000, Xenix.
SYSTAT	NCR Tower, Sun3*, Sun4*.

Notes: Not all versions of the makes and models may be supported. Listings denoted with an asterisk (*) indicate a near-future release date.

Figure 2. Goals for an End-User Interface

1. Trap potential errors before they are processed by the operating system and safeguard files from inadvertent destruction.

2. Create an intuitive and consistent interface between the user and the computer.

3. Enable users to customize certain aspects of their interface as appropriate to satisfy their needs.

4. Provide context-sensitive help instructions to explain features or choices.

5. Allow for easy access to the most frequently used application programs.

6. Minimize the response time between the user's action and the production of results.

7. Accommodate common file management functions, such as copying, moving, deleting and printing.

8. Standardize directory structures and naming conventions within each user's account, to ease the burdens of instruction and maintenance.

has often resulted in the inadvertent reformatting of the PC's hard disk. Moreover, system error messages can be incomprehensible and even disruptive to the operator.

The second goal from Figure 2 advocates the creation of an intuitive interface between the user and the hardware. It is difficult to pin down a precise definition of an "intuitive" interface. Clearly, this concept is closely related to ease of learning and usage. A leading expert on the human factors of computing once cited the Pacman arcade game as being an outstanding example of an intuitive end-user interface. Pacman requires only a minimum amount of operator instruction to play. The Pacman character's motions accurately correspond to the movements of the operator's joystick. Similarly, the ideal interface for UNIX would require a consistent set of instructions which would be quick to grasp and, perhaps, mnemonic in nature. In the next section, this paper will explore how some UNIX interfaces relate to this goal.

The third goal from Figure 2 expresses the need to incorporate into the interface features which customize the interface to meet certain user requirements or preferences. These features are advantageous, not only because they can save the user some programming or set-up time, but also because they increase the degree of participation that the user experiences.

DESCRIPTION OF UNIX END-USER INTERFACES

The next focus is on how the three types of UNIX interfaces relate to the goals enumerated in Figure 2. The three types are Command Line, Graphics User Interface and Text-Based Instructions such as menus.

Command-Line Interfaces

All UNIX systems come with a standard command-line interface. This interface is similar to that of most personal computers running PCDOS. For example,

```
"cat pf85t1.sps pf85t2.sps"
```

is a valid UNIX command for displaying the contents of two existing files on the terminal screen.

As the above example illustrates, the commands tend to be both ambiguous and difficult to memorize. Several other common UNIX commands pose a similar problem. The "mv" command either moves a file from one directory to another or renames the file in the same directory. The action of the "mv" command depends on whether or not the second argument following "mv" is a pre-existing directory name. A similar problem occurs with the UNIX "cp" command, which performs file copying. With either command, the action can result in a new file which overwrites the contents of a pre-existing file of the same name.

On-line help facilities are limited to the "man" command, which accesses an entry from the UNIX reference manual. This facility is largely geared towards programmers. All UNIX directory and file facilities are readily available through the command-line interface.

Graphics User Interface

While the Command-Line Interface is more useful for experienced programmers, the Graphics User Interface (GUI) is specifically oriented towards new and occasional users. The GUI can display both textual and graphical windows representing concurrent sessions running on different UNIX hosts. A "window" is merely a rectangular region of the terminal screen. It can be resized, scrolled or stacked without affecting the contents of any other displayed window. Because a GUI environment can display the output of various applications in its windows, it is ideally suited to the multi-tasking capabilities of UNIX.

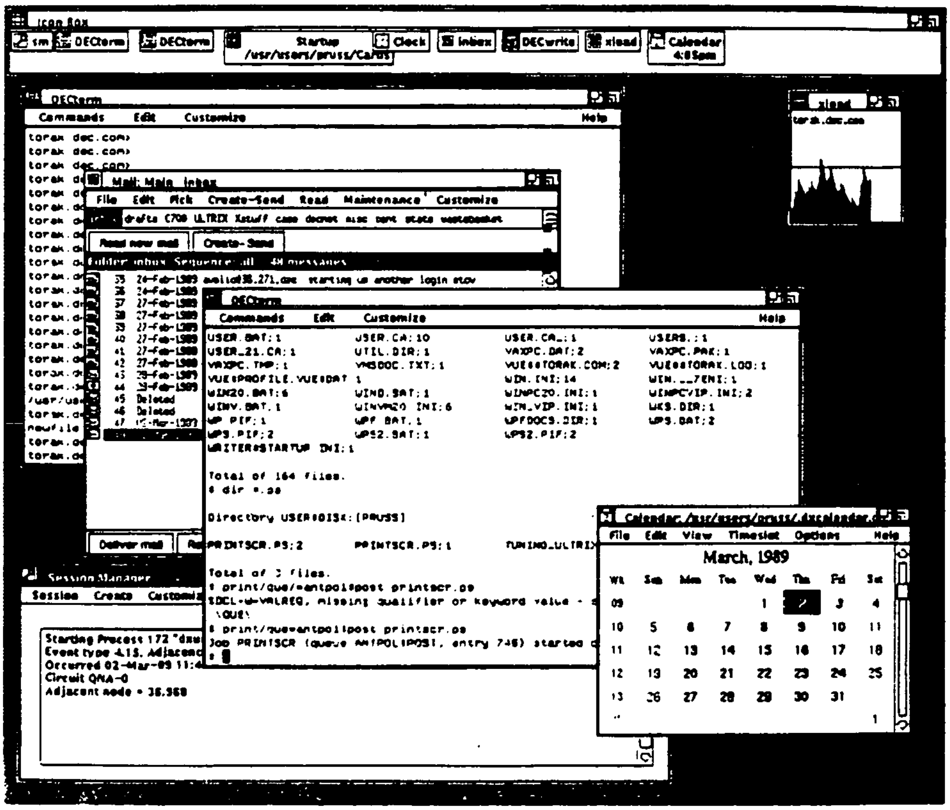
A typical GUI provides different mechanisms to improve the intuitive qualities of the user environment. Instead of a command line, the user manipulates a mouse device to position an arrow onto a "widget," an object on the graphics window. Icons are widgets that symbolize the functions that they perform. They are typically placed along the perimeter of windows. For example, a calendar icon can be activated to place a window containing a calendar image on the terminal screen. Scroll bars enable the user to access different regions of the window. Help windows provide context-sensitive instructions to assist the user through the steps of complicated procedures. Figure 3 is an example of DECWINDOWS, Digital Equipment Corporation's GUI for UNIX.

Application developers employ "toolkits," which are libraries of precompiled routines to adapt software to run in a GUI environment. For instance, AT&T's Open Look toolkit has a procedure to display a "notice" widget when the user is about to overwrite a file.

The development of GUIs for UNIX is in its infancy. At this writing, two incompatible standards are emerging, Open Look and the Motif standard, developed by the Open Software Foundation. Each standard is based on XWINDOWS, a set of windowing protocols developed by MIT. Within a given standard, disparate application programs can be controlled using a consistent set of actions. However, the lack of consensus concerning a single UNIX standard may serve to forestall the conversion of many applications from a text-based environment. 3/

Another barrier to the widespread adoption of these interfaces has been the cost of workstations to display high-resolution graphics. Workstations from Sun, Hewlett Packard, Apollo and DEC often cost $10,000 or more. Recently, however, several peripheral manufacturers (Network Computer Devices, Visual Technology, HDS, Tektronix, et al.) have developed dedicated XWINDOW terminals for as little

Figure 3. DECWINDOWS: A Unix-Based GUI


Icon Box
sm
DECterm
DECterm
Startup /usr/users/pruss/Carus
Clock
inbox
DECwrite
xload
Calendar 4:05pm
DECterm
Commands Edit Customize Help
xload
torak.dec.com
Mail: Main inbox
File Edit Pick Create-Send Read Maintenance Customize
drafts C700 ULTRIX Xstuff cams decnet misc sent stats wastebasket
Read new mail
Create-Send
Folder inbox Sequence all 48 messages
DECterm
Commands Edit Customize Help
Deliver mail
Session Manager
Session Create Customize
Calendar /usr/users/pruss/.dxcalendar
File Edit View Timeslot Options Help

```
torak dec.com>
torak dec.com>
torak dec.com>
```

```
35 26-Feb-1989 awalto@38.271.dec starting up another login stov
36 26-Feb-1989
37 27-Feb-1989
38 27-Feb-1989
39 27-Feb-1989
40 27-Feb-1989
41 27-Feb-1989
42 27-Feb-1989
43 28-Feb-1989
44 28-Feb-1989
45 Deleted
46 Deleted
47 12-Mar-1989
```

```
USER.BAT;1            USER.CA;10            USER.CA_;1            USERS.;1
USER_21.CA;1          UTIL.DIR;1            VAXPC.DAT;2           VAXPC.PAK;1
VAXPC.TMP;1           VMSDOC.TXT;1          VUE$$TORAK.COM;2      VUE$$TORAK.LOG;1
VUE$PROFILE.VUE$DAT 1                       WIN.INI;14            WIN.__7ENI;1
WIN20.BAT;6           WIND.BAT;1            WINPC20.INI;1         WINPCVIP.INI;2
WINV.BAT.1            WINVM20 INI;6         WIN_VIP.INI;1         WKS.DIR;1
WP PIF;1              WPF BAT.1             WPFDOCS.DIR;1         WPS.DAT;2
WPS.PIF;2             WPS2.BAT;1            WPS2.PIF;2
WRITER$STARTUP INI;1

Total of 164 files.
$ dir *.ps

Directory USER$DISK:[PRUSS]

PRINTSCR.PS;2         PRINTSCR.PS;1         TUNING_ULTRIX

Total of 3 files.
$ print/que/=antpoli$post printscr.ps
%DCL-W-VALREQ, missing qualifier or keyword value - s
 \QUE\
$ print/que=antpoli$post printscr.ps
Job PRINTSCR (queue ANTPOLI$POST, entry 746) started o
$ 
```

```
Starting Process 172 "dns
Event type 4.15, Adjacenc
Occurred 02-Mar-89 11:4
Circuit QNA-0
Adjacent node = 35.368
```

March, 1989

Wk	Sun	Mon	Tue	Wed	Thu	Fri	Sat
09				1	2	3	4
10	5	6	7	8	9	10	11
11	12	13	14	15	16	17	18
12	19	20	21	22	23	24	25
13	26	27	28	29	30	31	
14							1

as $1,500 per unit. These devices typically download the XWINDOW software from a host UNIX computer. A few vendors are also offering software to access the GUI from personal computers and local-area networks. 4/

Text-Based Interface

A third class of UNIX interfaces consists of text-based instructions to the user. Most text-based interfaces employ menus which provide users with options. Compared to GUI, the text-based interface offers lower hardware costs, as little as $200 per terminal. The trade-off is that text-based terminals can only display character sets and not graphics images.

AT&T has developed several facilities to assist the application's developer with the creation of text-based interfaces. The simplest facility is FACE, an acronym for "Framed Access Command Environment." FACE provides a series of predefined menus to assist the user with the management of his or her workspace. To afford greater customization, AT&T offers FMLI (Forms and Language Menu Interface), a shell-like language which can quickly generate simple menu-based systems.

Beyond these offerings, there are several facilities for "C" language programmers. ETI (Extended Terminal Interface) includes menu and window facilities, as well as specific routines to accommodate data-entry forms. These routines include form layout, data validation and field justification. "Curses" is a second toolkit of precompiled routines which provides the greatest flexibility and efficiency of window management. Most UNIX implementations include "Curses" without additional cost, whereas the other facilities are not as widely available or must be purchased separately.

The SOI Shell: A Text-Based Interface

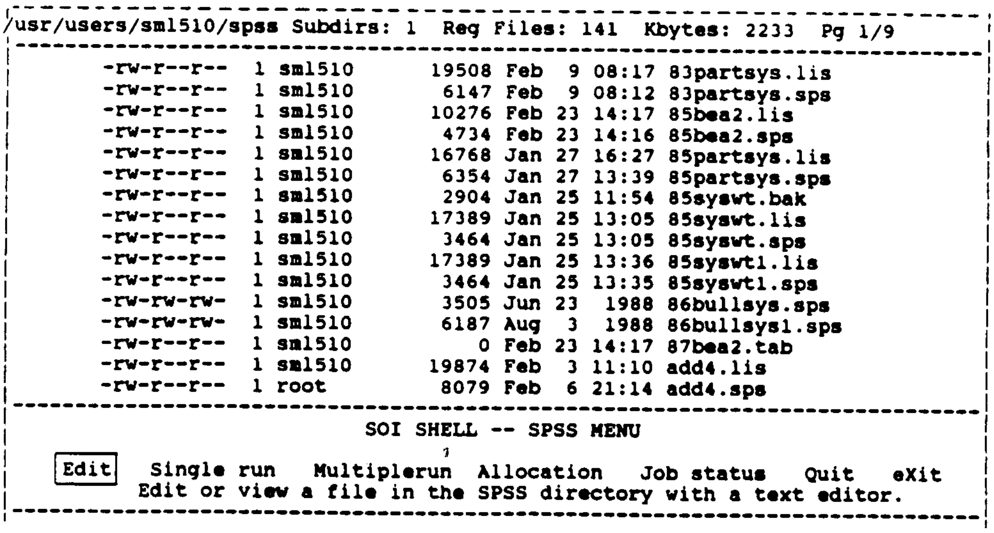
The remainder of this paper focuses on the text-based interface the author has implemented at the IRS. This interface, known as the "SOI Shell," attempts to address both the end-user goals of Figure 2 and the specific needs of the SPSS user community within IRS. The SOI Shell is coded using the "C" programming language and the "Curses" text-window management library. Although the author believes strongly in the merits of GUI, the choice of a text-based interface was dictated by the existing hardware and software in his organization.

One of the SOI Shell's design features is to provide each user with an identically structured and self-contained workspace. Each user has a "home" directory and several standard subdirectories for mail, SPSS statistical files, database files, file transfers and the user's own utilities. Except for the transfer subdirectory and certain public directories, one user cannot access files owned by other users or by the system. This standardization promotes system security and eases the administrative burden.

The SOI Shell screen has three zones, each of which is a separate window. The top zone provides information about the current directory, including its name and the number of files it contains. Below it is an alphabetically-ordered listing of files within the current directory. The listing is arranged in pages of 15 files. At the upper-right corner of the screen appear the current and total page numbers. In the bottom zone are the menu name and options.

The SOI Shell is controlled by a series of menus. These menus employ a design similar to Lotus 1-2-3 for the selection of options. 5/ The space bar highlights different menu options which are explained on the bottom line. Figure 4, below,

Figure 4. The SOI Shell's SPSS Menu



```
/usr/users/sml510/spss Subdirs: 1  Reg Files: 141  Kbytes: 2233  Pg 1/9
---------------------------------------------------------------------------
        -rw-r--r--  1 sml510      19508 Feb  9 08:17 83partsys.lis
        -rw-r--r--  1 sml510       6147 Feb  9 08:12 83partsys.sps
        -rw-r--r--  1 sml510      10276 Feb 23 14:17 85bea2.lis
        -rw-r--r--  1 sml510       4734 Feb 23 14:16 85bea2.sps
        -rw-r--r--  1 sml510      16768 Jan 27 16:27 85partsys.lis
        -rw-r--r--  1 sml510       6354 Jan 27 13:39 85partsys.sps
        -rw-r--r--  1 sml510       2904 Jan 25 11:54 85syswt.bak
        -rw-r--r--  1 sml510      17389 Jan 25 13:05 85syswt.lis
        -rw-r--r--  1 sml510       3464 Jan 25 13:05 85syswt.sps
        -rw-r--r--  1 sml510      17389 Jan 25 13:36 85syswtl.lis
        -rw-r--r--  1 sml510       3464 Jan 25 13:35 85syswtl.sps
        -rw-rw-rw-  1 sml510       3505 Jun 23  1988 86bullsys.sps
        -rw-rw-rw-  1 sml510       6187 Aug  3  1988 86bullsys1.sps
        -rw-r--r--  1 sml510          0 Feb 23 14:17 87bea2.tab
        -rw-r--r--  1 sml510      19874 Feb  3 11:10 add4.lis
        -rw-r--r--  1 root         8079 Feb  6 21:14 add4.sps
---------------------------------------------------------------------------
                         SOI SHELL -- SPSS MENU

    Edit   Single run  Multiplerun  Allocation  Job status  Quit  eXit
          Edit or view a file in the SPSS directory with a text editor.
---------------------------------------------------------------------------
```

displays the SPSS Menu as well as a page from the author's own SPSS file directory.

Most SOI Shell options permit a user to operate on files using either of two methods. In the "single" method, the user types the name of the desired file in a pop-up dialogue box. The "multiple" method enables a user to operate on multiple files residing in the current directory by highlighting each desired file. Using this method, the user can move a group of files to another of his or her subdirectories. Error conditions, such as the attempted movement of a directory file, are flagged in a pop-up dialogue box.

Due to space limitations, the full details of SOI Shell features cannot be described in this paper. However, the two following lists, Figures 5 and 6, summarize the most important features relating to SPSS and UNIX, respectively. It is clear from the points shown there that the SOI Shell provides an improved interface for end-users. As such, it substantially satisfies the goals set orth in the beginning of this paper. Those desiring further information are invited to contact the author at the address shown above.

Figure 5. SPECIAL SPSS FEATURES OF THE SOI SHELL

1. SPSS jobs execute using a script file, which designates the locations for program listings and error messages. Error messages will not overwrite the current contents of the screen.

2. The user can opt for either immediate or deferred job execution. Deferred jobs can be scheduled 24 hours into the future.

3. A job status option enables users to track up to nine SPSS jobs. The status report window lists job number, program name, time of execution and elapsed time. Jobs in progress or scheduled can be removed from the job queue.

4. When an SPSS job is finished, the SOI Shell displays a message in the dialogue box, identifying the program name and its job number.

5. The user can interactively increase or decrease the default memory allocation for SPSS jobs in one megabyte increments, ranging from one to four megabytes.

6. Automatic conversion to 132-column mode on VT100-mode-capable terminals to display the contents of special output files such as reports and cross-tabulations (tables procedure).

Figure 6. SPECIAL UNIX FEATURES OF THE SOI SHELL

1. Incorporation of standardized subdirectories in users' "home" directories, including bin (utility programs), spss (SPSSX statistical files) and mbox (electronic mail)--users are denied access to most areas outside of their home directories.

2. Development of an intuitive interface--the system features text windows for session status and directory listings; uses Lotus-style menus to select options; and employs cursor keys to select files and options.

3. Inclusion of extensive error-trapping routines to intercept problems before they are acted upon by UNIX--errors are reported to the operator in plain English and captured to an error log.

4. Availability of multi-file options for copying, moving, deleting and printing files--users can select all directory files or employ pattern-matching criteria.

5. Notification if destination files already exist for copying and moving--the SOI Shell will prompt for overwrite instruction, if desired.

Figure 6. SPECIAL UNIX FEATURES OF THE SOI SHELL -- continued

6. Provision of an enhanced file-dump utility to examine file contents, including control characters--output from the file dump can be routed to the screen or printer.

7. Employment of environment variables to control various session parameters--included are location of SPSSX files, default SPSSX memory allocation, overwrite permission preference and default text editor.

8. Application of safeguards--various tests prohibit the printing of files exceeding a specific length or exceeding a specific percentage of unprintable (e.g., binary) characters.

9. Provision of a utility to examine and change the permissions of files owned by the user--user, group and other permissions are displayed in plain English.

CONCLUSIONS

UNIX is clearly a very useful and versatile system for many computer applications, including statistical analysis. However, before investing in a UNIX system, the following points should be considered:

* UNIX is a multi-user and multi-tasking operating system which is widely available for both large and small computers. Its portability promotes the migration of source code from one machine to another. However, the availability of some statistical software is limited.

* The standard command-line interface of UNIX can pose serious problems to those unacquainted with this operating system.

* A Graphics User Interface (GUI) is highly desirable for end-users. However, the ease of use must be weighed against the relatively-high cost of graphics workstations and ancillary software. Moreover, a single, standard interface for graphics has yet to emerge for UNIX.

* Organizations can create their own custom interfaces to application software, such as statistical packages, using either GUI toolkits or text-based languages. Although such interfaces are expensive to program, the IRS experience has been that they do provide greater control over end-user actions than that provided by UNIX. The author's SOI Shell interface has also resulted in the savings of thousands of dollars in training costs for its target population of economists, statisticians and systems analysts.

FOOTNOTES

1/ UNIX, Open Look, FACE, FMLI and ETI are trademarks of AT&T. Motif is a trademark of the Open Software Foundation. SPSSX and SPSS are trademarks of SPSS, Inc. Lotus 1-2-3 is a trademark of the Lotus Development Corp.

2/ According to an informal survey conducted by the Seybold Office Computing Group, the lack of software was cited as the greatest impediment for UNIX in a commercial environment (UNIX Today!, May 15, 1989, p. 48).

3/ The survey results of software vendors' preferences between Open Look and Motif can be found in Unix Today!, July 10, 1989, p. 1.

4/ For a discussion of XWindow terminal vendors, see Digital Review, January 23, 1989, p. 4. A good overview of the overall XWindow market can be found in Digital Review, December 5, 1988, p. 77.

5/ The author selected the "Lotus" menu approach for several reasons. First, the interface is widely known among users of personal computers. Second, Lotus menus occupy only two or three lines of screens, whereas other approaches, such as "pull-down" menus, occupy larger amounts of territory. Since all of the IRS Statistics of Income Division's terminal links are asynchronous, the amount of time required for screen updating is minimized using the Lotus approach.

*This paper was originally presented at the '89 Interface Conference in Orlando, Florida, April 1989.