Comment

Peter J. Huber

Thisted rightly stresses the importance of *integration* in his definition of "environment," and the primacy of the *user interface*. I particularly like his way of dealing with the latter: "focus ... on the program's interactions with us while we do our work." It is just too easy for the person designing a data analysis system to forget this and to design a system either for a system designer or for a moron, rather than for a working statistician!

However, I am somewhat puzzled by his choice of examples. The ORION-1 project never intended to offer an environment in Thisted's sense, and his principal examples, which use a terminal to access Minitab on a time-shared machine, mainly demonstrate that Minitab lacks essential integration features. Among the working systems mentioned by Thisted, ISP and *S* would seem to be the only ones offering an environment in his sense.

In fact, back in 1979 it was precisely the lack of features such as those mentioned in Section 5.7 that made us reject Minitab when we were looking around for a "data handler" (with Thisted we would now say "environment") to support our incipient project on interactive graphical analysis of high-dimensional data. Since we could not find a ready made system, we then decided to build our own version of an interactive statistical processor (ISP).

In Section 5 Thisted discusses various desirable features. I should comment on some of our experiences with them. Multiple windows (5.2) and the ability to temporarily step out (end of 5.7) are just great and we would never want to miss them again—but unless you are very careful, they can create problems with record keeping. Some crucial interactions of the statistician may be off the record in a very literal sense, and it may become difficult to redo an analysis by running the journal file. By the way, we find (using ISP) that we almost never redo an analysis identically, except for teaching and demonstration purposes. But we continually "modify and redo" by editing and rerunning the journal files (we call them "scripts" for obvious reasons!) and by linking branches. This destroys the tree structure of the analysis (replacing it by a kind of multilayered graph with cycles) and makes it even more important that the journal is complete, including comments (having them on a separate sheet, or in some notebook file, simply won't do). Partial redos are easy on an Apollo because of the scroll, cut, and paste facilities of its display manager—but they are dangerous (some named items may no longer be what you think). I would vote against Thisted's proposal in Section 5.4. The line numbers are bound to create problems with editable and executable journal files, even if the system also provides a RENUM command ... and this at a time when True BASIC at long last is trying to do away with the line numbers!

Thisted barely touches the hardware issues. But it is an implicit consequence of his discussion that the proper hardware to host a system as envisaged by him is a single user workstation with a *very fast* processor (otherwise the continuous monitoring proposed in Section 5.8 is not feasible—for example, I found that an Apollo DN600 with hardware floating point was too slow—and with a time-shared machine the response time would be too erratic), virtual memory (if only to allow nontrivial multitasking), and high resolution color graphics (high resolution for multiple windows and color to facilitate comparison).

In this connection, I should like to add a few remarks on hardware and software life cycles. While the hardware substrate may be given and beyond the control of most of us, it certainly does not look unchangeable to me (Section 3). I have now been using computers in a serious way since 1956, and hardware and operating systems seem to change quite regularly every 3 years, with the programs somehow surviving several such changes. Thus, some large programs I wrote in 1972–1974 for a CDC-6500 now have migrated via DEC-10, VAX, and Apollo to a Compaq. It may be tempting to design a specific piece of hardware for a specific purpose, but in the long run I guess you will get further ahead by adapting and riding the crest of the general purpose hardware wave. The point is that a software package as large as an interactive data analysis system is unlikely to reach maturity within a single hardware cycle, and if you design a special purpose device, you will be overrolled by the next general purpose wave. Also, you never can accurately predict the direction the next cycle is going to take. For example, I would never have thought in 1981–1982 that by 1985 one could squeeze a more powerful and faster commercial version of ISP, with a larger workspace to boot, into a personal computer (an IBM PC compatible with an 8087 and 640k memory) than what we had then on a VAX.... I believe that the writing is on the wall, and that in interactive data analysis the days of the time-shared computer are counted.

Peter J. Huber is Professor of Statistics, Department of Statistics, Harvard University, One Oxford Street, Cambridge, Massachusetts 02138.