New horizons, new standards

Robert M. Poston, Standards Editor

In the fifties there were no national-level computing standards; in the sixties a few were starting to emerge. The seventies brought code and hardware interface standards; the eighties brought standards for the process of creating computing components. Today, professionals are able to work on some computing systems projects without using any standards at all, but tomorrow it will not be feasible to work standard-less.

Papers and discussions from the recent Compstan '86 (Computer Standards Conference) in San Francisco point to that conclusion. At the conference my job was to listen to sessions and discussions, gather reports, eavesdrop on informal conversations, and then put together a summary for the final plenary session. As I prepared my presentation, three perspectives on surviving in the computing industry through increased reliance on standards became evident: the changing marketplace, the shorter time to market, and the change in professional attitudes.

Changing marketplace. Once we could sell a new product because customers were anxious to have the latest innovation. Now customers look for a product that will solve problems, whether or not it embodies the latest technology. What a big change for us—moving from a technology-driven to a user-oriented market. This means that standards now must change to accommodate this new trend.

Two interesting proposals for new standards, both addressing end-user needs, were presented at the conference. Thom Foote-Lennox of the Pelican Group suggested software ergonomics guidelines for user interfaces. His guidelines would provide the end user with a way to evaluate competitive products. Thomas Miller of the University of Idaho and Robert Burton of Brigham Young University proposed a standard for the evaluation of math function libraries. Today, most users just plug into available math function libraries and take it for granted that correct results will be forthcoming. Miller and Burton point out the need for these results to be verified. A math function library standard would enable end users to evaluate the accuracies of function libraries and to choose between competitive products. I believe these recommendations for end-user standards mark the beginning of a new wave of such proposals.

Shorter time to market. There always has been pressure to get the product out the door as quickly as possible. In recent years the development life cycle of a computing product has been getting shorter and shorter, and the pressure to produce has become greater and greater. That shrinking life cycle and continuing pressure has forced many producers (not you, IBM) to concentrate resources on one or two computing components rather than attempting to build all system components. When you're building components, interfaces to other components become very important. If a wrong interface is selected, it can mean death to the product. Interfaces are defined in standards. Shorter time to market suggests the necessity for greater use of standards.

Most standards that we have in place now, especially the IEEE ones, are really interface standards. The interfaces may be between computing experts or computing components. ANSI/IEEE Std 730-1984, a standard for software quality assurance plans, is an interface between computing experts. ANSI/IEEE Std 802.3-1984, local network for computer interconnection logical link control, is an example of a standard for component interfaces. One exception I can think of is P1008, a standard for software unit testing. It is a standard for a process, not for an interface.

All papers at the conference that described standards were actually referring to interface standards. All had an unstated purpose of minimizing time to market.

Changing professional attitudes. In the sixties when professionals delivered "good" software, it meant the software would "work." But, who made that judgment? What did "work" mean? We
had no professional standards for determining quality then.

Herbert Hecht of SoHaR led a lively conference discussion on standards for software reliability. Hecht defined reliability as a way of describing frequency of failures. As the discussion progressed, several useful ideas on how to evaluate the quality of a product were put forward.

Fletcher Buckley of RCA indicated that a reliability measure by itself was not adequate. Availability, a measure of duration of outage after a failure (how fast you can reload), is just as important as reliability. Sometimes it is more important, Buckley believes.

Jim McCall of Science Application International added that either of these measures is acceptable after the product is being used by the customer. Unfortunately, that's too late. McCall stressed the need for a test comprehensiveness measure that could be applied before the product is shipped. Combined measures of reliability, availability, and test comprehensiveness would go a long way toward letting an individual know before the product is released, whether or not he or she has done a good job. Producing a professional quality product in accordance with ANSI, IEEE, X3, EIA, or US government standards will become even more important in the near future. A session titled "Standards and the Law" headed by Mike Carrio of Teledyne/Brown ended in some revealing discussions. Presently, there are several cases in court where computing companies are being sued for delivering products that contained latent defects (bugs). The basic defense being used against these charges is that programmers followed reasonable and prudent professional practices (that is, they followed standards) in developing their products.

As the computer continues to infiltrate new areas of our society, more professional standards will be required. As end users become increasingly sophisticated, demanding easier ways to compare products, we'll need very precise new standards. With mounting legal issues, we'll require a higher degree of professionalism from our standards makers than ever before expected.

Our work is cut out for us.

Acknowledgment
Thanks to everyone who helped organize and pull off Compsan '86, especially the following session reporters, who worked hard to help me cover this event: Rita Alvarodiaz, IBM; Jack Barnard, AT&T Information Systems; Donald Booker, Pace University; Carl Cargill, DEC; Miguel Carrio, Teledyne/Brown; James Clary, RTT; Donald Close, Hughes Aircraft; Stewart Crawford, AT&T Bell Laboratories; Ronald DalPezzo, IBM; H. Dorris, AT&T; William Farr, Naval Surface Weapons Center; Bruce Gundaker, AT&T Bell Laboratories; Charles Hollocker, AT&T; Al Lowenstein, RCA; Phillip Marriott, CTA; Paul Menchini, Calma; Edward Miller, Software Research; Wilma Osborne, NBS; Lee Sprague, IBST; and Gary Whitten, AT&T Information Systems.

Operating system group adopts name, publishes trial-use document

Jim Isaak, IEEE 1003 Chairperson

The IEEE 1003 operating-system standard effort now has a name—Posix—that can be pronounced and used as a quick reference point. The name, suggested by Richard Stallman, combines the initials from portable operating system with the classical "ix" ending of many Unix-compatible operating systems.

The new name will appear on the cover of the IEEE 1003.1 Trial Use Standard, which is now in print and can be used to state conformance to the standard. A system can be a Posix-compatible implementation (or an IEEE 1003.1-compatible implementation), and application code can be a Posix-compatible application.

The challenge facing the working group is to move from the trial-use document to a full-use document, incorporating the comments that come in over the next months, and to complete the tasks outlined in the appendixes.

As a result of the April meeting, held in Florence, Italy, and attended by representatives from 11 countries, we have a much better understanding of international concerns. We also established closer contact with the X/Open group, which is aggressively developing and promoting a standard environment for portable applications.

Recommendations out of the Florence meeting included a draft statement on internationalization and a question about mandatory locking. The group is actively seeking feedback in these and other areas. The primary objective from an international perspective is to ensure the feasibility of Posix-compatible international implementations. A secondary objective is to incorporate changes that encourage international capabilities. In the area of mandatory locking, the group is seeking an application example to help resolve outstanding questions.

We have established a set of priorities for addressing the outstanding issues. Detailed work began at the June meeting in Atlanta and will continue at the September meeting in Palo Alto. Our time frame, as it now stands, is to go back into the balloting phase in the spring of 1987 and to have that completed for review at the September 1987 IEEE Standards Board Meeting. The process of reviewing the Posix standard as a possible FIPS will begin shortly.

The IEEE 1003.1 Trial Use Standard (order number 967) is available for $19.95 US, plus $4 for shipping and handling, from the IEEE Computer Society, PO Box 80452, Worldway Postal Center, Los Angeles, CA 90080; (800) 272-6657 or (714) 821-8380 in California.