

Briefing

Information technology: ISO/IEC JTC 1 Subcommittee 22, Programming Languages, their Environments, and System Software Interfaces

By John L. Hill, Chairman, ISO/IEC JTC 1 SC 22, Programming languages, their environments and system software interfaces, Sun Microsystems

This article provides an insight into the scope, work programme and strategic direction of a standards committee that 'silently' and 'invisibly' serves the needs of IT programmers and compiler developers.

The relative cost and value of software and hardware has shifted dramatically since the first commercial general-purpose programmable computer UNIVAC, which became available 50 years ago. Software development has become a major capital item and a vital part of the information infrastructure. Software supports practically all dimensions of social, educational and industrial life e.g. automated teller machines, airline reservation systems, avionics control, payroll systems. These services typically rely extensively on computers that run programs written in the languages standardized by ISO/IEC JTC 1/SC 22.

SC 22 scope covers a limited set of popular programming languages, some specific formal specification languages, POSIX and various language bindings. Its standards primarily focus on high level programming environments to improve portability of applications, productivity and mobility of programmers as well as compatibility of applications over time. SC 22's standards are produced and maintained in response to extensive and diverse market requirements.

At a high level of abstraction the work program of SC22 has three main goals

- support the current, massive, global investment in software applications by maintenance and improvement of standardized programming languages
- continuously improve programming environment standardization through the documentation and application of forty years of lessons learned in the specification of these standards.
- respond to emerging technological opportunities such as network applications and global development practices

Programming Languages

Although press attention is primarily devoted to newer language and interface areas, SC22 focuses on both the important mature and the emerging language and interface technologies. This is due to the inventory of installed software (based upon the more mature languages) and the revolutionary developments taking place in the newer languages. Legacy software packages often span 25 or more years of application

development and comprise an inventory of hundreds of millions of lines of currently operational code. Written in programming languages such as COBOL, Fortran, Basic, APL, and PL/I these software applications often support a huge installed base of programs that are fundamental to enterprises all over the world. The SC22 work-program in these areas focuses on enhancements to improve compatibility for interchange of data, and possibly processes, with new hardware and operating systems. Adding current technological capabilities to older programming languages in this way extends the life of applications and provides for continued return on investment to current users. Additions to existing standards for the more mature language standards typically implement sufficient features of the newer languages to maintain a high level of compatibility and migration of the installed software base in a deliberate, controlled manner.

The 'newer' programming languages standards (e.g. C, C++) typically offer a richer set of features and functions. These languages tend to focus on delivering the semantics and syntax to implement new functionality while also interoperating with the more mature languages.

SC 22 also standardizes bindings between programming languages and system services. Conformance to the SC22 POSIX standards is required by a wide variety of private and public sector users for "UNIX" systems procurement. Co-operation between IEEE, The Open Group and SC22/WG15 has resulted in a set of vendor adopted stable base standards which ensure mature products meeting their customers' requirements. These standards also facilitate the conformance testing suites and activities that provide the user community with a wide choice of well tested products.

Programming Environment

SC 22 has worked diligently to improve its standards by years of close work with user groups. Two examples, at a high level of abstraction, are productivity of programmers and internationalisation. In terms of programmer productivity, SC 22's standards have enabled significant reductions in the effort required to write meaningful computer programs. Higher-level languages have contributed greatly. The emergence of the world wide web has given rise to the need for programs that are easily modified for working around the world. Internationalization, enabling simplified ways to write programs that can be localized to suit most any culture, is the means for accomplishing this. SC 22 is working to implement the Universal Character Set contained in ISO/IEC 10646.

Technological opportunities

Current work in progress and future strategic directions include development of a viable application environment for network-based computing, standardization requirements for safety-critical software, real-time embedded system software, high performance computing enhancements and component-based software capabilities.

SC22's work on standardization of formal description languages provides the means for expressing truly complicated processes such that one can model those processes in a mathematically precise manner. This helps with the development of highly reliable applications of a mission critical nature.

Methodologies

An important aspect of SC22 work is that it uses a variety of development methods to make the most efficient use of standards development resources. These include:

- Development within traditional international working groups
- Assignment of development responsibility to a National Body, with a requirement for appropriate synchronization of the international and national comment processes
- Joint or co-located meetings of SC 22 WGs with National Body or liaison organization development committees working in the same area, to produce standards with identical text
- Work with NB or liaison organizations to fast-track documents in the area of SC 22 responsibility e.g FORTH, ECMAScript, CHILL
- *Permit* National Body Observer members allowed to participate actively in the Technical Work

Summary

For over three decades ISO/IEC JTC 1 SC 22 programming language standards have contributed to enabling users to run applications across vendor and system architectural boundaries. The availability of its standards enables application programs to be transported from one hardware and operating software platform to another with a minimum of expense and enables users of computing to more easily grow their computing resources according to their operational needs. SC 22 includes many of the programming language experts from all over the world. Their work, coupled with that of the liaison organizations, provides the essential high-quality, stable, interoperable standards to enable different cultures throughout the world reap advantage from information technology.