SOFTWARE STANDARDS

Software standards are distinguished from software engineering standards in that they define characteristics of a software product, while the latter specify the process to be used in developing a software product. A software standard is an interface specification—a specification that describes services provided by a software product—to which any vendor can build products. There are two important points. First, the specification is available to any vendor and evolves through a consensus process that is open to the entire industry. Second, the specification defines only an interface or protocol, so different vendors can provide the standard functionality on their proprietary systems even though they may implement it with different methods.

Software standards are developed with a specific market or industrial application in mind. They are usually produced by professional associations such as the Institute of Electrical and Electronics Engineers (IEEE) Computer Society or by industry consortia such as the Open Group. They may be offered for formal standardization if they appear to be of general interest to the computer industry as a whole. IEEE 802.3 (Ethernet) was an electronic industry communications standard prior to adoption as a formal standard. Formal standards are sometimes referred to as de jure standards, since they evolve through a formal process that is open to all, consensus based, and results in public specifications.

Formal standardization is conducted by (a) the American National Standards Institute (ANSI) in the United States and (b) the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) in the world community. These organizations are voluntary activities that support the formal development and use of standards by industry and government. They have no specific authority to enforce the use of these standards other than the commitment by industry and government entities to require their use. However, many government contracts require compliance with formal national and international standards, and large user organizations require many of their suppliers to also meet these same standards. Because of the complexity of software products and the tremendous growth of international trade, the use of national and international standards is at an all-time high.

Popular software products are often referred to as de facto standards. Postscript by Adobe Systems and Microsoft Windows are two widely known examples. De facto standards have been with us since there was more than one way to do something. Typically, de facto refers to a method or product that has become so predominant within a given arena that it is accepted as the "standard" way something is done or produced. Usually, de facto standards are proprietary technology (i.e., developed and owned by a specific company) that is copied or imitated by others, most probably through a license agreement for a fee—that is, a specific company owns the intellectual property rights to the software. Intellectual property rights are a critical factor in the adoption of formal standards. Because of their proprietary nature, de facto "standards" are not true standards and are not considered further in this article, although much of the discussion applies to them as well.

INTEROPERABILITY AND PORTABILITY

Standard components for software have been advocated for at least two decades, but until recently only limited versions of such components, such as mathematical subroutine libraries, have been available. Barriers to more complex components have been both technical and economic.

Modern programming languages such as C++, Java, and Ada are helping to solve technical problems associated with component development, but a software component created today in a language such as C++ must still use system services (e.g., timers, database access, communications) that vary depending on the operating system, database, data communications, and other vendor specific functions. If a component makes extensive use of Digital's VMS operating system services, for example, then a different version of the component must be created for UNIX. Software standards change this situation, making it economically practical to develop standard components.

Software standards make it possible to develop standard software components that can be implemented on a wide variety of hardware, making a software components industry economically practical. But standards do not solve all problems associated with building interchangeable software components. Software designers need to understand the capabilities and limitations of software standards, as well as how to deal with these limitations. This article describes important software standards and explains how they can be used to build portable, interoperable application software components.

There are two important aspects to software standards: interoperability and portability. Interoperability refers to the capability of applications running on different computers to exchange information and operate cooperatively using this information. Portability refers to the capability of software to run on different types of hardware. Portability can further be broken down into binary portability and source code portability. Binary portability makes it possible to move an executable copy of a program from one machine to another. Source code portability requires a program to be recompiled when moving from one machine to another. The development of portable application software components depends on portability standards. Software systems that are built on standards for portability and interoperability are called *open systems.* Interoperability standards are necessary but not sufficient for a complete open systems environment.

A good example of interoperability is provided by the X Window System protocol, which specifies how graphics primitives can be communicated between an application program and graphics software running on a workstation. An X Window application running on an IBM workstation can interact, for instance, with a user sitting at a Sun workstation. The ISO Open Systems Interconnection (OSI) standards also promote interoperability. The OSI reference model defines a structure or reference model for data communication standards. The reference model defines seven layers of communications system components, from hardware at the bottom to application software at the top. The model describes how components communicate with each other; that is, it is a model for interoperability. Open system standards for application portability are complementary to data communication standards. Communication standards define communication services, but open system applications require a standard way to use those services.

Binary portability specifications are designed to provide software portability at the object code level. For example, the IBM PC hardware interface can be regarded as a de facto standard for binary portability. Executable copies of software can run on PC clones from many different manufacturers. Another example is the Application Binary Interface for systems based on the Sun SPARC processor. This specification for workstations makes it possible to move executable programs between different makes of workstation as easily as programs can be moved between different IBM PC clones.

Sun's Java bytecode format provides portablity for software distributed across the Internet. Java bytecodes are downloaded from a server and executed on a user's machine using a Java interpreter. As such, the Java bytecode format can be considered a form of binary portability. Binary portability is more difficult to achieve than source code portability, because it places constraints on hardware features or instruction sets, or because it requires special interpreters. To date, standards efforts have concentrated primarily on developing interfaces for source code.

Open system standards for source code portability define interfaces available to application programs for services such as timing functions, security features, database access and many other essential functions. Standards could be defined by cooperatively developed source code or "reference implementation," but within most standards organizations the preferred approach has been to specify interfaces and let vendors develop competing implementations. Thus, two different operating systems may provide the same services, but one may have better performance or fault tolerance characteristics than the other. The application program interface (API) is normally specified in terms of a set of procedure calls for a particular programming language.

The best-known standards for operating system services are the POSIX standards being defined by the IEEE Portable Application Standards Committee (PASC), formerly known as the Technical Committee on Operating Systems (TCOS). (The acronym POSIX is derived from Portable Operating System Interface, with an "X" to denote its UNIX origin.) Beginning with the POSIX System Application Program Interface standard (IEEE 1003.1-1988), the IEEE has been developing a comprehensive set of standards for application portability. In

1997, the POSIX effort comprised more than 20 working groups.

The POSIX efforts (1003.x) have been supplemented with projects to develop standards for application interfaces to services such as windows and X.400 message handling, which are useful on non-POSIX systems. Other open system standards have been developed through ANSI, ISO, and other organizations. Many of these other specifications have been combined with the developing IEEE standards to define an open systems environment using the POSIX interface standards as the basis.

STANDARDS DEVELOPMENT ORGANIZATIONS

Software standards are developed by a variety of organizations. There are two major categories of standards organizations, both of which use a consensus-driven process. One consists of formally recognized standards bodies, responsible for definition and dissemination of public standards. Their specifications are known as formal or accredited standards. National and international standards groups, many professional and technical organizations, and certain trade associations are examples of formal standards bodies.

The other standards organization category consists of informal bodies. Informal standards bodies are typically created by suppliers or users of information technology to enable the implementation of standards. They produce specifications known as industry standards. Certain trade associations, industry groups, vendor consortia, and user groups are examples of informal standards bodies. Informal standards groups often submit their specifications to formal standards organizations for approval as a recognized accredited standard.

Formal Standards Bodies

The most significant and influential standards are those with internationally agreed status. In information technology, these are produced and published primarily by the International Standardization Organization (ISO) and International Electrotechnical Commission (IEC) in Geneva, Switzerland, and the International Telecommunications Union (ITU), an agency of the United Nations. National standards bodies are organized under ISO. (The acronym "ISO" is said to derive from the Greek prefix "iso," meaning "same," and is not an acronym of the organization's name in any language.) Founded in 1947, the ISO includes national standards bodies from 90 nations, one from each nation. The ISO cooperates with other international standards organizations, including the IEC and ITU.

The principal international standards body for information technology is ISO/IEC Joint Technical Committee One (JTC1), which is operated jointly by the ISO and the IEC. ISO/IEC JTC1 is composed of a set of subcommittees that develop standards for computing and communication systems, including a great number of software standards.

Industrial Bodies

In addition to formal standards bodies that are associated with government organizations, some industry consortia produce public specifications that serve as industry standards. The largest such consortium is the Open Group, with nearly 800 member organizations. Other important consortia include the World Wide Web Consortium, which supervises common specifications for Web software and protocols, and the Internet Society, which oversees evolution of the Internet in cooperation with the Internet Engineering Task Force (IETF), which develops protocol standards.

Formed in 1996 by a consolidation of the Open Software Foundation (OSF) and X/Open, the Open Group develops specifications for open systems products. Member companies produce products that meet these specifications, and organizations that purchase software may cite these specifications in product requisitions. The Open Group also provides testing and branding services for its specifications.

THE STANDARDS DEVELOPMENT PROCESS

The development of software standards typically requires several years of effort by large working groups. Each standards organization has its own process, but most national standards bodies follow a process similar to that of the ISO. Because industrial standards bodies work with proprietary specifications, their development process may be shorter. The development process used by ISO JTC1 is explained in this section.

ISO Procedures

The rapid evolution of software technology poses a significant challenge to developing formal standards. In response, the ISO/IEC JTC1 Directives provide three different procedures for developing international standards:

1. A five-stage process which has been the traditional approach used in cases where development of the standard is expected to require an extended time period.
2. A fast-track process where economic forces or other considerations call for rapid adoption of an existing standard that has already been approved by a recognized standards development organization.
3. Finally, in cases where it is advantageous to convert an existing de facto industrial standard into a formal international standard, JTC1 now provides an experimental procedure for transportation of publicly available specifications.

ISO/IEC JTC1 Five-Stage Process

Under the traditional approach, the development of a standard progresses through five stages, with a special Preliminary stage being optional:

0. *Preliminary.* An initial study period may be conducted.
1. *Proposal.* Voting members ballot on the creation of a new standards project.
2. *Preparatory.* A Project Leader supervises development of a Working Draft.
3. *Committee.* A committee develops consensus on a Committee Draft.
4. *Approval.* National bodies vote on a proposed Final Draft International Standard.
5. *Publication.* An International Standard is published within the ISO and IEC.

Preliminary Stage. This stage is optional. During the preliminary stage, a study is conducted to determine if there is a need to submit a proposal for a new standard. A study period may be required when it is too early to formulate a precise proposal but when agreement exists that the subject area is likely to need future standardization.

Proposal Stage. The Proposal stage begins with the submission of a written proposal for the creation of a new standard. The proposal may be submitted by a JTC1 National body, by a JTC1 Subcommittee, by an ISO or IEC Technical Committee or Subcommittee, or by a JTC1 liaison organization. The proposal contains a description of the scope of the new work item, its purpose, a program of work (including a timetable), the identification of relevant documents that could serve as a basis for the new standard, and a list of organizations or bodies with which cooperation will be needed to develop the standard. The proposal is circulated to all national bodies that are members of JTC1, to JTC1 subcommittees, and to liaison organizations for consideration. In order for approval as a new work item, the proposal must be supported by a majority of national bodies that are members of JTC1. In addition, at least five national bodies of the subcommittee to which the project will be assigned must provide technical experts who will participate in the project. The proposal stage ends when the proposal is approved and included within the list of projects in the program of work of a JTC1 subcommittee.

Preparatory Stage. The Preparatory stage covers the creation of a working draft of the standard. Within the subcommittee, the work is performed by experts from national bodies who may be organized into working groups under the guidance of a Convener. Working groups may be further subdivided into project areas, each under the direction of a Project Leader. Project Leaders report to the Convener of their parent Working Group (WG). The Working Group Convener may serve as a Project Leader but is additionally responsible for coordinating any other Project Leaders in the WG as well.

As technical work is completed by a Working Group, it is documented in a working draft of the standard. Each working draft has a Project Editor responsible for the electronic form of the document, but the Project Leader has the overall responsibility for coordinating the efforts necessary to gain approval of the draft as an international standard. Each JTC1 subcommittee may establish its own process for developing a working draft. During the Preparatory stage, several successive working drafts of a standard may be created and reviewed by the Working Group. The Preparatory stage ends when a working draft of a standard has been approved for elevation to the Committee stage by the JTC1 subcommittee that the Working Group is part of. It is then registered as a Committee Draft.

Committee Stage. The Committee stage begins with the circulation of the document in the form of a Committee Draft (CD) among the national bodies of the subcommittee. The CD may be distributed for discussion at an upcoming subcommittee meeting, for comment by correspondence, or for formal balloting.

The CD is sent out by the subcommittee Secretariat to all its national bodies that are members of JTC1 and JTC1 liaison organizations. All recipients are asked for comments on the CD. If a ballot is being conducted, voting members are asked to vote on the acceptance of text of the CD. The period for review of a CD for either commenting or voting is normally three months, but it may be up to six months. The results are collected and summarized by the subcommittee Secretariat. A CD may be designated by the Project Leader, in consultation with the subcommittee Secretariat, as a Final Committee Draft or FCD. When this is done, the document is circulated to all JTC1 national bodies, to all ISO national bodies, and to all members of IEC for a voting period that normally lasts four months but may be up to six months. Voting members are asked to vote on the acceptance of the text of the FCD as a Final Draft International Standard and of the advancement of the project to the Approval stage.

The subcommittee Secretariat reviews the ballot and the comments to determine the degree of consensus obtained. The subcommittee Secretariat decides whether to discuss the CD at the next subcommittee meeting or to ask that a revised CD be prepared for circulation to the national bodies. This decision can be made by the Secretariat in conjunction with the Project Editor. Decision-making responsibility may also be delegated to a Working Group or to a team comprised of affected Working Group Conveners, Project Leaders, and Project Editor. If at least three national bodies disagree with the decision of the subcommittee Secretariat to call for a ballot or for comment by correspondence, the CD is discussed at a meeting. In the case of an FCD, a decision may also be made to advance to the Approval Stage and register the CD as a Final Draft International Standard.

Consideration of successive drafts continues until consensus is reached on an FCD and the substantial support of the national bodies is obtained to either elevate the project to the Approval stage or to abandon the project. An FCD is advanced to the approval stage only after the text is stabilized. The determination of the extent of support for a decision is the responsibility of the subcommittee Secretariat which, if necessary, may consult with the administrative body that both oversees JTC1 and coordinates its actions with ISO and IEC.

If the subcommittee decides to proceed and the FCD moves to the Approval stage, the document is registered as a Final Draft International Standard.

Approval Stage. The Approval stage begins with circulation of the Final Draft International Standard (FDIS) for a ballot to all national bodies that are members of JTC1, to all members of ISO, and to all members of IEC for a two-month voting period. Recipients are asked to vote on the approval of the FDIS for publication as an International Standard. Recipients may vote only yes or no. Further technical or editorial changes are not allowed during this stage.

An FDIS is approved if a two-thirds majority of votes cast by voting members of the committee are in favor and if not more than 25% of the total number of votes cast are negative. Abstentions are excluded when counting votes. If the FDIS is not approved, the project is reverted to the Preparatory stage, and the document becomes a working draft again.

The Approval stage ends when the decision is reached either to elevate the FDIS to the Publication stage or to revert to the Preparatory stage. If the decision is to advance to the Publication stage, but before submission to ISO for publication, the Project Editor is asked to make any minor corrections to the manuscript that may have been discovered during the ballot period.

Publication Stage. The subcommittee Secretariat submits the final manuscript in electronic form for publication by ISO. Further editorial or technical amendments are unacceptable at this stage. The Publication stage ends with the release of the document as an International Standard.

Maintenance of Published Standards. Like any software system or technical product, a developed international software standard must be maintained so that it is kept up-to-date. Because of the rapid advances that are continually occurring in the field of software technology, maintenance of software standards is a particularly critical activity. Thus, in addition to procedures for developing standards, JTC1 also prescribes procedures for maintaining existing international standards after they are finalized.

Typically, the JTC1 subcommittee that develops an international standard is responsible for maintaining it. Existing standards are reviewed within the subcommittee on a periodic basis, although not more than five years may elapse from the date of publication of the latest edition of a standard before such a review takes place. As a result of such a review, a standard may be confirmed if it is found to be up-to-date, it may be revised in order to ensure it is consistent with recent technical developments, or it may be withdrawn if it is judged to be obsolete.

Maintenance of international standards also involves identification and correction of recognized defects in the standard. JTC1 has developed a defect correction procedure for this purpose. If a subcommittee determines that it is necessary to apply these procedures to a standard being maintained, an editing group is created for this purpose and a Project Editor is appointed. The editing group receives defect reports submitted by JTC1 national bodies, by a JTC1 subcommittee, or by a liaison organization. The Project Editor develops a consensus response of the editing group to each report. The response indicates whether no change is required, an editorial correction is required, further consideration is needed, or a technical correction is required.

In the case where an editorial change is needed, a revision to the text of the standard may be proposed. In cases where further discussion is needed to formulate a response, the defect report may be discussed at a future meeting within the responsible subcommittee. In cases where it is decided to respond to the defect report by making a technical correction, a proposal for making the correction first is circulated to national bodies that are members of the subcommittee for a three-month ballot period. If the proposal is approved or if a proposal modified on the basis of comments made by national bodies is approved, a technical corrigendum is issued which corrects the standard. Published technical corrigenda normally incorporate a substantial number of technical and editorial changes that are processed by an editing group over time.

ISO/IEC JTC1 Fast-Track Process

It is also possible to propose an existing standard for fast-track processing, a procedure that greatly shortens the time needed to develop an international standard. An existing standard from any source may be proposed by a national body that is a member of either JTC1 or a JTC1 liaison organization. A project proposal may be suggested, or a working draft that is being processed under the JTC1 five-stage procedure also may be proposed. The proposer must provide a Project Editor and may recommend a JTC1 subcommittee to which the standard should be assigned.

After setting any copyright or trademark issues that would prevent unrestricted circulation, the proposed standard is registered for fast-track processing by the administrative body that oversees JTC1. The standard is circulated to all JTC1, ISO, and IEC Member National Bodies as a Draft International Standard (DIS) for a six-month ballot period. The JTC1 Secretariat determines which of its subcommittees should process the DIS and ballot results. The subcommittee may form a ballot resolution group and appoint a Convener. The group is comprised of representatives of Member National Bodies and the Project Editor.

The results of the ballot and comments by Member National Bodies are provided to the subcommittee and discussed at a ballot resolution group meeting. At this meeting, the representatives attempt to reach consensus on the final content of the DIS. If necessary, voting takes place with votes being cast by individual Member National Bodies. A determination that there is a sufficient number of votes for approval is made by using the voting criteria described in the Approval stage of the JTC1 five-stage process.

If agreement is reached on the final version of the text of the DIS, the document is amended by the Project Editor and is then returned to the subcommittee Secretariat. It is then forwarded for publication as an International Standard. If agreement cannot be reached, the original proposal for fast-track processing is deemed to have failed, and the procedure is terminated.

Transposition of Publicly Available Specifications

The procedure for transposition of publicly available specifications (PAS) into international standards was established for trial use by JTC1 in 1995. The procedure is similar to the fast-track process, but it applies to a broader class of technical specifications originating from a wide range of organizations that are external to ISO or IEC. To be considered a PAS, a specification must meet a set of criteria that were established to ensure a high level of quality and proper treatment of matters concerning intellectual property rights.

To submit a PAS into the transposition process, an organization that has developed the specification must first be recognized by JTC1 as an eligible PAS submitter. To obtain this recognition requires submission of a detailed application showing that the developing organization meets criteria that include the demonstration of a cooperative attitude in developing consensus agreements, a desire to pursue standardization within JTC1, and a willingness to comply with ISO/IEC policies on intellectual property rights. The submitter must also identify in the application the PAS that will be submitted. The application is circulated for a three-month vote to all national bodies that are members of JTC1. Approval gives the submitter the right to submit specifications to the transposition process that are within the scope of the approved application for two years.

Once a PAS is submitted, the specification is processed using the fast-track procedure. Prior to submission, the PAS submitter may recommend which JTC1 subcommittee should process the PAS. The submitter may also provide a representative to serve as a Project Editor or may request that the specification is to remain unchanged throughout the process. After submission, a PAS submitter may choose to withdraw the specification from the transposition process at any time prior to publication.

After conclusion of the six-month combined vote by JTC1, ISO, and IEC national bodies, a ballot resolution group may reach a decision to accept the PAS, to modify the PAS on the basis of the vote and accompanying comments, or to reject the PAS and terminate the proceedings. If the decision reached is unacceptable to the submitter, the document may be withdrawn and the procedure terminated. If agreement is reached on the final version of the text of the PAS, the document is forwarded for publication as an International Standard. Maintenance of an International Standard transposed from a PAS is handled in accordance with agreements made between JTC1 and the submitter on the basis of information provided at the time of submission of the PAS.

USING SOFTWARE STANDARDS

Developing software products using software standards can be substantially different from development on proprietary platforms. This section explains the characteristics of standards-based development and describes methods of building software using open systems standards.

Characteristics of Standards

They Represent a Consensus of Industry. The Institute of Electrical and Electronics Engineers (IEEE) requires a 75% return rate and 75% approval from the ballots returned for a standard to be adopted. Other standards bodies follow similar rules. The development of a standard depends on the contribution of expensive personnel time and resources by industry. There is tremendous pressure to develop consensus, to ensure that the standard is approved, and, above all, to ensure that it is used as a basis for products by most of the industry. A standard that is ignored by the vendors who build systems is of no value to users. The standard that results from the consensus building process often contains features that represent the "lowest common denominator" among systems provided by vendors. Other features are likely to be entirely new, the result of compromises worked out where there was no common service definition.

They Change Periodically. Recognizing that standards must stay reasonably current if they are to remain useful, standards bodies revise their work periodically, typically with a revision cycle (to revise or reaffirm the standard) of about five years. The revision period is a tradeoff between lagging the industry and changing so fast that the standard has so many versions that it is not a standard at all.

They Often (But Don't Always) Lag the State of the Art. Most of the work of national and international standards bodies is involved with standardizing things that are already existing practice. The development of standards requires support from industry, which is usually not achievable unless there is a consensus already. Some recent efforts have been directed toward developing standards in relatively new areas of technology, but these efforts may not succeed if there is no common existing practice.

Since strong support from industry is required to approve a standard, usually only functions that are provided by most vendors are standardized. Novel features provided by one or two vendors, no matter how technically interesting, are not acceptable by the majority unless they are easy to implement. There are some exceptions, such as the OSI Class 4 Transport protocol and the POSIX Real Time extensions (1003.4). In these cases, standards have been developed where there is no universal existing practice, but many standards will continue to lag developments in industry.

They Are Typically More Precise Than Most Software Specifications. Standards are developed by large groups working over several years. The focus of the effort is on making the definitions as precise as possible. However, except for data communication protocols, standards documents generally describe system functions in natural language, as do most software specifications. Despite the inherent problems of natural language specifications, standards documents tend to be more precise and complete than typical software specifications, because they are given careful review by more people than most software specifications. A standards balloting group may have more than 200 people, whose incentive for careful review is to protect the interests of their organizations.

Potential Pitfalls of Software Standards

Standards reflect industry consensus at a given point in time. As innovations are made and spread among vendors, standards must change to reflect what has become new common practice. To accommodate this process, the IEEE and other standards bodies schedule standards for periodic revision. A typical revision period is five years. If an application depends on the use of k standard components, there is a potential for changing the application to meet changes in the standards an average of $k/5$ times a year. A large application might use 10 or more standards. Since most large applications take several years to complete, developers can expect some of the standards they are using to be revised during the development period.

Standards revisions are also a factor in maintenance. For an organization with n applications, $kn/5$ change efforts (or more if the standards revision cycle is less than five years) may be needed just to keep up with evolving standards. While the changes needed may be small, as most changes to standards are to provide new functions rather than modify existing ones, an effort will be required to review applications to determine whether changes are needed or not.

The varying stages of completion or revision of standards can add to integration difficulties during development. A system may need services that are not available from any existing operating system, either because a standard for the service interface is not completed or because the existing standard is about to be revised and changes are expected in revision. Since standards usually lag the state of the art, some applications will need services that cannot be provided by any standard interfaces. Different systems may have different ways of providing a service, requiring developers to evaluate and decide which implementation to adopt.

Design for Standard Software Components

Given some of the shortcomings of open system standards, what is the most effective way of using them to achieve appli-

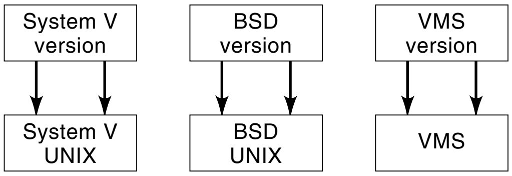

Figure 1. Application development without software standards.

cations portability or interoperability? One approach is to build components that provide services specific to an industry or an individual organization, resulting in a hierarchy of services: generic system services provided by standards such as the POSIX kernel, industry-specific services provided by components built on the system services, and organization-specific services provided by components built on the industry-specific services and the generic system services. Application programs for end-users can be built to use the application program interfaces (API) provided by the hierarchy of components.

A simple example of an organization specific interface might be a specification for a function that displays a company logo, department name, and time of day on a graphics terminal. A software component to provide the specified service uses operating system functions to obtain the time of day, bitmap for the logo, and department information from a database. Many of the organization's application programs use the same service, and the application programs may run on many different brands of computer. The application programs call these APIs, rather than calling operating system services directly. This approach is sometimes used today to deal with system dependencies, and it will still be necessary when open system standards are used.

It is also possible to specify an API for a particular industry. For example, ISDN service standards have been defined for the telephone industry, banking, retailing, and other industries. Components providing the services specified in the API can be built on standard interfaces. Because they are built using open system standards, the components can be ported to a wide variety of hardware at low cost. Without open system standards, multiple versions might need to be developed to accommodate differences in file system structure, network interfaces, and other system services. Component development becomes more attractive for software developers because it is not necessary to produce a separate version for different operating systems.

Application architecture is usually as pictured in Fig. 1. With open system standards, a model such as that shown in Fig. 2 is more useful. Arrows represent use of a component's

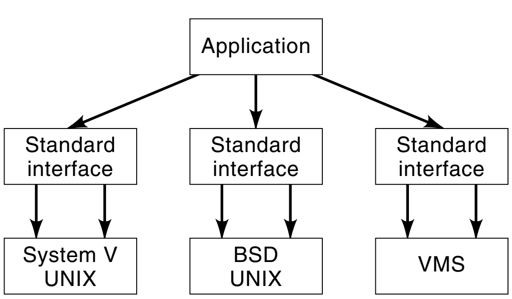

Figure 2. Application development with software standards.

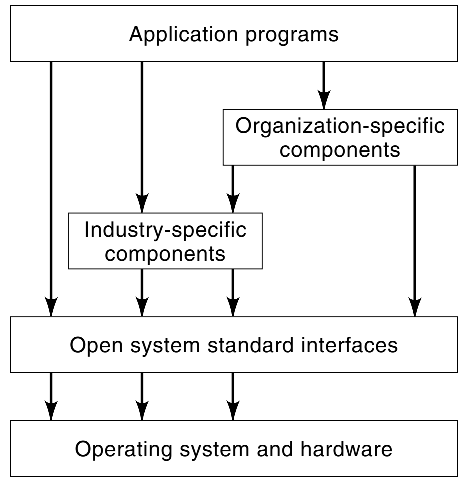

Figure 3. Component-based software architecture.

services by the component at the tail of the arrow. Application programs use functions provided by an organization-specific API or by an industry standard API, or they may use standard functions directly. An organization-specific API is built using services provided by industry standard APIs or by standard open system services such as POSIX. Components supplying industry-specific services can be built using the standard services and can be marketed for use on many different hardware platforms (Fig. 3).

Organization standard services isolate the application from the interface with industry standard system services. Modifications to accommodate changes in organization requirements are restricted to the organization standard service modules. The organization standard services can be compared with objects in the object-oriented programming paradigm. The difference is that an organization standard service is more general than a typical "object." Consider a video game example: An object might be a ship that is moved by the user, the organization standard services could be functions that calculate speed and heading from coordinates, while the standard functions used are provided by the compiler's math library. In this case the organization standard services are equivalent to a very general class of object that moves in a two-dimensional plane.

Industry standard system services thus provide independence from hardware and operating systems, while the organization standard services provide independence from system services. This helps insulate the application from proprietary operating system services as well as from changes in standards.

SOME IMPORTANT SOFTWARE STANDARDS

POSIX Open Systems Environment

No single standard provides all the functionality needed by a modern computing environment. Portability and interoperability require a comprehensive set of standards. The POSIX Open Systems Environment (OSE), being put together by IEEE PASC working group P1003.0, provides a standard set of interfaces to information system building blocks, covering both portability and interoperability standards. Not all of the specifications in the POSIX OSE are IEEE POSIX (1003.x) standards. POSIX functions serve as a basis, supplemented by other applicable open systems standards.

Two types of standard interfaces are specified in the POSIX OSE: the application program interface (API) and external environment interface (EEI). The POSIX OSE Reference Model, shown in Fig. 4, shows the relationship of these interfaces to the other parts of the computing environment.

The External Environment refers to external entities with which the application platform exchanges information, including both the human end user, hardcopy documents, and physical devices such as video displays, disks, printers, and networks. External environment interfaces provide for external communications and interoperability. EEI standards take the form of communication protocols, record and document formats, and display formats. The application program interfaces in the POSIX OSE are source code interfaces, generally in the form of programming language procedure calls, to the application platform, which is the operating system and hardware. By specifying a standard set of procedure calls, an API provides source code portability. To explore how standards can be used in constructing open software it is useful to examine the POSIX OSE application program interfaces in some detail.

The POSIX OSE API contains four categories of interfaces: System Services, Communications Services, Information Services, and Human Computer Interaction Services. A typical computing environment will require some, but not all, of the standards contained in each of these four categories. A fifth category, Domain Services, is provided for special-purpose environments, such as transaction processing. Each of the four main categories includes a collection of applicable standards, although some of the standards provide different services than others within the same category. The POSIX OSE guide also includes specifications such as UNIX System V Interface Definition and OSF/1. These are not formal standards, but proprietary specifications that include standard functionality as a subset.

Fundamental System Services. This category includes both language services and operating system services. Language services are the functions typically provided by programming languages such as C, Fortran, Pascal, and others. Operating system services are the services normally associated with an operating system or executive such as UNIX or VMS.

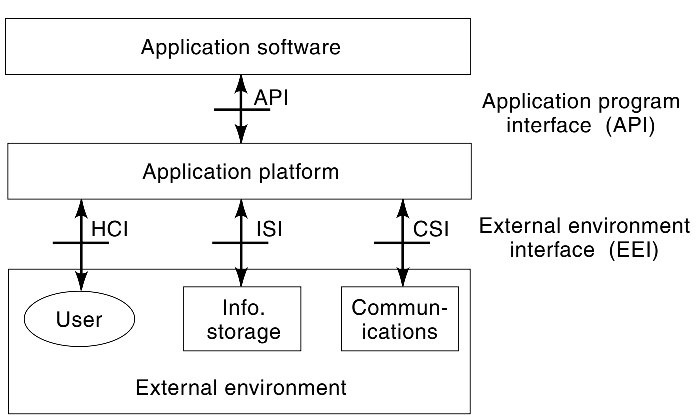

Figure 4. POSIX Open system environment reference model.

The language service area gives standard specifications for such programming languages as Ada, BASIC, C, C++, and Pascal. To make other services in the OSE accessible from applications programs, language bindings are needed for one or more of these languages. The POSIX kernel standard (1003.1), originally defined using C, has Fortran (1003.9) and Ada (1003.5) language bindings also. Standard interfaces to all of the other services have not necessarily been defined for all of the languages in the POSIX OSE. The most common language for POSIX interfaces is C. In addition to providing a way to call system services, the languages themselves provide functions for programs, including mathematical functions, data definition and representation, error handling, I/O operations, and program control logic.

Operating system services are used to control the resources of the computer system. They are the functions provided by an operating system or executive program, such as UNIX, VMS, Multics, and many others. The major categories of operating system services in the POSIX OSE are process management; task management; environment services; process communication and synchronization; input/output; file management; event, error, and exception management; time services; memory management; logical naming; system initialization, reinitialization, and shutdown. Standards in the OSE providing these functions include POSIX Shell and Utilities (1003.2), Realtime (1003.4), and the Microprocessor Operating System Interface (IEEE 855).

Communications Services. Communication services make communication possible for application programs running on machines connected via a network. They include services for file transfer, namespace and directory services, network file access, remote procedure call, protocol-independent network access, and data representation. Both Application Program Interface and External Environment Interface functions are included in this area.

The EEI functions provide interoperability and include standards for physical connections, network protocols and formats, and distributed system services. The TCP/IP protocols are the most commonly used standard for packet switched digital communications.

The interface to the interoperability functions is through the standard APIs, such as the Protocol Independent Interface (1003.12) and the remote procedure call interface being developed by ANSI X3T5.5. The communications services component of the POSIX OSE is still evolving; other standard services may be added later.

Information Services. Information services include Database Services, which provide the capability to store and retrieve data from long-term storage, and Data Interchange Services to exchange data between systems.

Data Interchange Services include data description protocols, character sets, and data format protocols. Data description protocols provide a standard means of associating a name with individual data elements. Data format protocols add attributes that describe the physical characteristics of the data. Among the Data Interchange Services are the Standard Generalized Markup Language (ISO 8879, 1986) that can be used to define the layout and structure of a document, the Electronic Data Interchange standards being developed in ANSI X12 for Data Format Protocols, the Computer Graphics Metafile (ANSI X3.122, 1986) which provides a standard means for storage and exchange of computer graphics, and the X.400 Message Handling System (ISO/CCITT X.400-1984, 1988) which defines message protocols for electronic mail.

Database Services are the functions associated with database management systems. These include: data definition and manipulation—the ability to create, update, delete records, fields, or tables; data access—the ability to retrieve data based on complex search conditions; and data integrity—locking of data items, transaction control, and synchronous writes. Application programs use database services extensively, and the APIs in the Information Services area include the Structured Query Language (ISO 9075, 1982) and Network Data Language (ISO 8907, 1987).

Human–Computer Interaction Services. Human–computer interaction services provide functions for communication between user and computer, using the window and mouse style of interaction popularized by the Apple Macintosh.

Applicable external environment interface standards include (a) the X Window protocol, which specifies the format and meaning of messages between application program and display device, (b) human factors standards such as the User–System Interfaces and Symbols standard being developed within ISO-IEC/JTC1/SC18/WG19, and (c) the ISO 9241 standards for Ergonomics of Visual Display Terminals.

INDUSTRIAL STANDARDS DEVELOPMENT ORGANIZATIONS

The Object Management Group (OMG) is an industry consortium of over 700 software vendors and end users. The goal of the OMG is the creation of a set of industry specifications that will establish a common framework for development and management of application systems in distributed, heterogeneous computing environments. These specifications define standard interfaces for software components that are intended to enable interoperability across a wide variety of hardware platforms and operating systems. The OMG has been active since 1989 and has produced a significant body of specifications that are impacting commercial software development. Other industrial standards organizations follow similar procedures.

OMG employs an object-oriented programming approach. In this view, software components are "objects" that store data and perform functions. Each "object" software component is an independent entity that communicates by sending messages and exchanging data with other software components. In the OMG framework, software objects perform defined computing services. Application systems invoke or directly incorporate "object" software components through interfaces standardized by OMG (see Fig. 5).

The OMG framework describes broad classes of services that may be performed by software components that employ standardized interfaces. The OMG also defines a Common Object Request Broker (CORBA) to manage the invocation of these components.

Common Facilities provide services that are of general use to application developers. There are two categories of Common Facilities: Horizontal and Vertical. Horizontal Facilities include Desktop Management, Information Management, System Management, and Task Management Facilities.

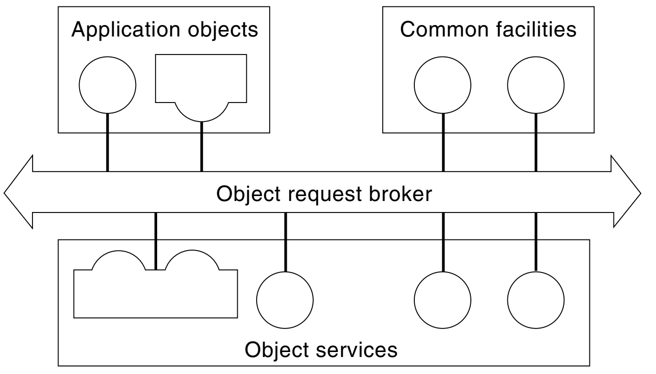

Figure 5. Reference model: context of common facilities.

These facilities are intended to be useful in all application domains and are more user-oriented. Vertical Facilities, on the other hand, are targeted at specific application domains—for example, healthcare, computer integrated manufacturing, accounting, geographic information systems, telecommunications, and others.

Object Services are lower-level and more widely applicable than Common Facilities. They are intended to standardize the life-cycle management and maintenance of objects. Examples include an Object Naming Service, a Persistent Object Service (for creation of objects that are stored permanently), a Transaction Service, and an Event Notification Service.

The OMG allows for definition of Domain Interfaces to software components that provide very specific services of direct interest to end-users in particular application domains. Domain Interfaces may utilize Common Facilities and Object Services, but are intended to perform particular tasks for users within a certain vertical market, industry, or other defined area of activity.

Similar to Domain Interfaces, the OMG also provides for the creation of interfaces to Application Objects. These interfaces are not standardized by OMG but are instead defined by users for software components that perform specific tasks as part of application systems.

In the OMG view, an application is typically built from a large number of different objects, including Application Objects intended to perform functions for a particular application, Objects that perform functions within the domain the application belongs to, Common Services, and basic Object Services.

The CORBA provides the communication infrastructure that allows the application system to communicate with the software components that provide the different services described above. The CORBA allows software components developed using OMG-defined interfaces to send and receive requests and responses to each other. Communication and exchange of data may take place without regard for the programming languages in which the software components are implemented, the type of platform on which they execute, the specific host on which they execute, or the networking protocols used to communicate between hosts.

At the present time, the OMG specifications are beginning to come into widespread use. Successful OMG implementations have been reported by software vendors, universities, and developers of application systems. As the need for distributed computing continues to grow and interest in object-oriented computing increases, it can be expected that the OMG will remain a driving force in software standardization.

APPENDIX 1. COMMON STANDARDS ACRONYMS

AAP: Association of American Publishers
ACSE: Association Control Service Element
AFNOR: Association Francaise de Normalisation
AJPO: Ada Joint Progam Office
ANS: American National Standard
ANSI: American National Standards Institute
AOW: Asiatic Oceania Workshop
API: Application Program Interface
APP: Application Portability Profile
APTL: Accredited POSIX Testing Laboratory
ASI: Application Software Interface
ASME: American Society of Mechanical Engineers
ASN.1: Abstract Syntax Notation One

BRI: Basic Rate Interface
BSD: Berkeley Systems Development
BSI: British Standards Institution

CAD/CAM: Computer-Aided Design and Manufacturing
CADETC: CAD/CAM Data Exchange Technical Centre
CAE: Computer Application Environment
CASE: Computer-Aided Software Engineering (see ISEE)
CCITT: International Telegraph and Telephone Consultative Committee (renamed International Telecommunication Union Telecommunications Standardization Sector [ITU-T])
CGM: Computer Graphics Metafile
CNIDR: Clearinghouse for Networked Information Discovery and Retrieval
COBOL: Common Business Oriented Language
COE: Common Operating Environment
CORBA: Common Object Request Broker Architecture
COS: Corporation for Open Systems
COSMIC: Computer Software Management and Information Center

DAC: Discretionary Access Control
DBMS: Database Management System
DCE: Distributed Computing Environment
DIA: Defense Intelligence Agency
DIN: Deutsches Institut fuer Normung
DIS: Draft International Standard
DISA: Defense Information Systems Agency
DNI: Detailed Network Interface
DPANS: Draft Proposed American National Standard
DoD: Department of Defense
DTD: Document Type Definition

ECMA: European Computer Manufacturers Association (name changed to "ECMA: Standardizing Information and Communications Systems"; ECMA is no longer an acronym in this context)
ECMA/TC33: ECMA: Standardizing Information and Communications Systems/Technical Committee 33
EDI: Electronic Data Interchange
EDIFACT: Electronic Data Interchange For Administration, Commerce, and Transport
EEI: External Environment Interface
EIA: Electronic Industries Alliance
EMPM: Electronic Manuscript Preparation and Markup
EPRI: Electric Power Research Institute
EWOS: European Workshop on Open Systems

FDDI: Fiber Distributed Data Interface
FIPS: Federal Information Processing Standard

GCA: Graphics Communication Association
GIS: Geographic Information System
GKS: Graphical Kernel System
GOSIP: Government Open System Interconnection Profile
GUI: Graphical User Interface

HCI: Human/Computer Interface
HTML: Hypertext Markup Language
HTTP: Hypertext Transfer Protocol

ICCCM: Inter-Client Communications Conventions Manual
IDRP: Inter-Domain Routing Protocol
IEC: International Electrotechnical Commission
IEEE: Institute of Electrical and Electronics Engineers
IGES: Initial Graphics Exchange Specification
IGOSS: Industry/Government Open Systems Specification
INTAP: Interoperability Technology Association for Information Processing
IRDS: Information Resource Dictionary System
IS-IS: Intermediate System-Intermediate System
ISDN: Integrated Services Digital Network
ISEE: Integrated Software Engineering Environment
ISO: International Organization for Standardization
ISO/IEC: International Organization for Standardization/International Electrotechnical Commission
ITL: Information Technology Laboratory (part of NIST)
ITU: International Telecommunication Union
ITU-T: International Telecommunication Union—Telecommunications Standardization Sector [ITUT] (formerly CCITT)

JISC: Japanese Industrial Standards Committee
JITC: Joint Interoperability Test Center
JTC1: Joint Technical Committee One
JPEG: Joint Photographic Experts Group

LAN: Local Area Network
LAPD: Link Access Procedure on the ISDN D channel
LIS: Language Independent Specification
LOC: Level of Consensus

MAC: Mandatory Access Control
MAN: Metropolitan Area Network
MAP/TOP: Manufacturing Automation Protocol/Technical and Office Protocols
MHS: Message Handling Service
MIME: Multipurpose Internet Mail Extensions
MPEG: Motion Pictures Expert Group

NASA: National Aeronautics and Space Administration
NBSIR: National Bureau of Standards Interim Report
NCC: National Computing Centre (UK)
NCGA: National Computer Graphics Association

NCITS: National Council for Information Technology Standards
NCSC: National Computer Security Center
NI-X: Bellcore National ISDN-X
NISO: National Information Standards Organization
NIST: National Institute of Standards and Technology
NISTIR: National Institute of Standards and Technology Iterim Report
NIU-Forum: North American ISDN Users' Forum
NIUF: North American ISDN Users' Forum
NTIS: National Technical Information Service
NVLAP: National Voluntary Laboratory Accreditation Program (NIST-sponsored program)

OIW: OSE Implementor's Workshop
OMG: Object Management Group
OSE: Open System Environment
OSE/RM: Open System Environment Reference Model
OSF: Open Software Foundation
OSI: Open System Interconnection

PCTE: Portable Common Tools Environment
PDES: Product Data Exchange using STEP
PDDF: Portable Document Delivery Format
PDF: Page Description Format
PHIGS: Programmer's Hierarchical Interactive Graphics System
PII: Protocol Independent Interfaces
POSIT: Profiles for Open Systems Internetworking Technologies
POSIX: Portable Operating System Interface (POSIX)—System Application Program Interface [C Language]
PRI: Primary Rate Interface
PRL: Problems/Limitations

RDA: Remote Database Access
RPC: Remote Procedure Call

SDIF: Standard Document Interchange Format
SDTS: Spatial Data Transfer Specification
SGML: Standard Generalized Markup Language
SHA: Secure Hash Algorithm
SNI: Simple Network Interface
SPDL: Standard Page Description Language
SQL: Structured Query Language
SSL: Secure Socket Layer
STEP: Standard for the Exchange of Product Model Data
SVID: System V Interface Definition

TEI: Text Encoding Initiative
TFA: Transparent File Access

UAC: User Advisory Council
UI: UNIX International
UNI: Ente Nazional Italiano di Unificazione
UN/ECE/WP.4: United Nations Economic Commission for Europe, Working Party Four on Trade Facilitation
USGS: U.S. Geological Survey

VAN: Value-Added Network
VPL: Validated Products List

W3C: World Wide Web Consortium
WAN: Wide Area Network
WYSIWYG: What You See Is What You Get

X3: Standards Committee X3—Information Technology
XML: Extended Markup Language
XTI: X/Open Transport Interface

APPENDIX 2. STANDARDS DEVELOPMENT ORGANIZATIONS

The following organizations are responsible for distributing standards for various standards-making organizations. Ordering and fee information for specific standards may be obtained directly from the addressees.

AAP
Association of American Publishers
EPSIG (Electronic Publishing Special Interest Group)
c/o OCLC
6565 Frantz Road
Dublin, OH 43017-0702
Phone: (614) 764-6000

ANSI
American National Standards Institute
11 West 42 Street, 13th Floor
New York, NY 10036
Phone: (212) 642-4900
ANSI International Publications

Information on standards from ISO and its member bodies (e.g., DIN, BSI, JISC), IEC, and CEN/CENELEC

Phone: (212) 642-4995

ANSI General Sales (National Standards)
Phone: (212) 642-4900

CCITT (renamed ITU-T)
International Telegraph and Telephone Consultative Committee
Place des Nations
CH-1211 Geneva 20
Switzerland

COSMIC
Computer Software Management and Information Center
The University of Georgia
382 East Broad Street
Athens, GA 30602
Phone: (706) 542-3265
FAX: (706) 542-4807

Department of Defense
Defense Printing Service
Standardization Documents Order Desk
700 Robbins Avenue
Building 4-D
Philadelphia, PA 19111-5094
Phone: (215) 697-1187 or (215) 697-2179

Any Federal organization or DoD contractor can order numerous types of standards, including FIPSes and MIL-STDs from the Defense Printing Service.

Data Interchange Standards Association
ASC X12 and PAEB Secretariat
1800 Diagonal Road, Suite 355
Alexandria, VA 22314
Phone: (703) 548-7005
FAX: (703) 548-5738

ECMA
European Computer Manufacturers Association
Rue du Rhone 114
CH-1204 Geneva
Switzerland
Phone: 011-41-22-735-36-34

EIA
Electronic Industries Alliance
2500 Wilson Blvd.
Arlington, VA 22201-3834
Phone: (703) 907-7500

Federal Information Processing Standards (FIPS)
U.S. Department of Commerce
National Technical Information Service (NTIS)
5285 Port Royal
Springfield, VA 22161
Phone: (703) 487-4650
FAX: (703) 321-8547

NIST publishes an index of FIPS that is available through NTIS. Request "NIST Publications List 58."

GCA
Graphic Communications Association
199 Daingerfield Road
Alexandria, VA 22314-2888
Phone: (703) 519-8160
FAX: (703) 548-2867

GPO
Government Printing Office
Superintendent of Documents
U.S. Government Printing Office
Washington, DC 20402
Phone: (202) 512-1800

IEC
International Electrotechnical Commission
3 Rue de Varembe
P. O. Box 131
CH-1211 Geneva 20
Switzerland
Phone: 011-41-22-34-01-50

IEEE (for accepted standards)
The Institute of Electrical and Electronics Engineers, Inc.
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Phone: (800) 678-IEEE or (800) 678-4333

IEEE (for draft standards)
1730 Massachusetts Avenue, N.W.
Washington, DC 20036-1903
Phone: (202) 371-0101

IETF
Internet Engineering Task Force
IETF Secretariat
c/o Corporation for National Research Initiatives
1895 Preston White Drive, Suite 100
Reston, VA 22091
Phone: (703) 620-8990
FAX: (703) 620-0913
Internet: ietfsecretariat@cnri.reston.va.us
WWW: http://www.ietf.cnri.reston.va.us/home.htm

ISO
International Organization for Standardization
Central Secretariat
1 Rue de Varembe
P. O. Box 56
CH-1211 Geneva 20
Switzerland
Phone: 011-41-22-34-12-40

ISOC
Internet Society
12020 Sunrise Valley Drive, Suite 270
Reston, VA 22091
Phone: (703) 648-6888
FAX: (703) 648-9887 or (800) 468-9707 (USA only)
E-mail: isoc@isoc.org

ITU-T (formerly CCITT)
International Telecommunication Union—Telecommunications Standardization
Sector
Place des Nations
CH-1211 Geneva 20
Switzerland

JTC1 TAG
Joint Technical Committee 1 Technical Advisory Group
Information Technology Industry Council (ITI)
Director, JTC1 TAG Secretariat
1250 Eye Street NW, Suite 200
Washington, DC 20005-3922
Phone: (202) 737-8888 (Press 1 twice.)
FAX: (202) 638-4922 or (202) 628-2829

National Computer Graphics Association
2722 Merrilee Drive, Suite 200
Fairfax, VA 22031
Phone: (703) 698-9600

National Computer Security Center
INFOSEC Awareness Division
ATTN: IAOC (X711)
Ft. George G. Meade, MD 20755-6000

National IGES Users Group (NIUG)
c/o NCGA, Suite 200
2722 Merrilee Drive
Fairfax, VA 22031-4499
Phone: (703) 698-9606 x330
E-mail: jzink@uspro.fairfax.va.us
FAX: (703) 560-2752

National Institute of Standards and Technology (NIST)
Building 225, Room B266
Gaithersburg, MD 20899
Phone: (301) 975-2821
FAX: (301) 926-3696
WWW: http://www.nist.gov

National Technical Information Service (NTIS)
U.S. Department of Commerce
5285 Port Royal
Springfield, VA 22161
Phone: (703) 487-4650
FAX: (703) 321-8547

Network Management Forum
1201 Mt. Kemble Avenue
Morristown, NJ 07960-6628
Object Management Group (OMG)
492 Old Connecticut Path
Framingham, MA 01701
Phone: (508) 820-4300
FAX: (508) 820-4303

OSF
Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142

SQL-Access
SQL Access Group
c/o Robert Crutchfield
Fransen and Associates, Inc.
2171 Campus Drive, Suite 260
Irvine, CA 92715
Phone: (714) 752-5942

T1 Standards
Standards Committee T1 Telecommunications
1200 G Street, N.W., Suite 500
Washington, DC 20005
Phone: (202) 434-8845
FAX: (202) 393-5453

U.S. Product Data Association (US PRO)
c/o NCGA, Suite 200
2722 Merrilee Drive
Fairfax, VA 22031-4499
Phone: (703) 698-9606 x308
FAX: (703) 560-2752
E-mail: uspro@uspro.fairfax.va.us

X3 (renamed National Council for Information Technology Standards)
American Standards Committee X3—Information Technology
Information Technology Industry Council (ITI)
Director, X3 Secretariat
1250 Eye Street NW, Suite 200
Washington, DC 20005-3922
Phone: (202) 737-8888 (Press 1 twice.)
FAX: (202) 638-4922 or (202) 628-2829
X/OPEN—X/OPEN Portability Guide (XPG)
1750 Montgomery Street
San Francisco, CA 94111

BIBLIOGRAPHY

C. F. Cargill, *Information Technology Standardization, Theory, Process, and Organizations,* Bedford, MA: Digital Press, 1989.

G. Fisher, *Guide on Open System Environment (OSE) Procurements,* NIST Special Publication 500-220, Gaithersburg, MD: National Institute of Standards and Technology, 1994.

International Organization for Standardization, *IEC/ISO Directives—Part 1, Procedures for the Technical Work,* 1989.

International Organization for Standardization, *IEC/ISO Directives—Part 2, Methodology for the Development of International Standards,* 1989.

International Organization for Standardization, 3, *IEC/ISO Directives—Part Drafting and Presentation of International Standards,* 1989.

Institute of Electrical and Electronics Engineers, *Guide to the POSIX Open Systems Environment, IEEE P1003.0,* 1995.

R. Soley and C. Stone, *Object Management Architecture Guide, 3rd ed.,* New York: Wiley, 1995.

RICHARD KUHN
THOMAS RHODES
CHRISTOPHER DABROWSKI
US Department of Commerce

SOFTWARE TESTING.

See PROGRAM TESTING.