IEEE P1003.0 Draft 13 – September 1991

Copyright © 1991 by the
Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017, USA
All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change. The publication, distribution, or copying of this draft, as well as all derivative works based on this draft, is expressly prohibited except as set forth below.

Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only, and subject to the restrictions contained herein.

Permission is hereby also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position, subject to the restrictions contained herein.

Permission is hereby also granted to the preceding entities to make limited copies of this document in an electronic form only for the stated activities.

The following restrictions apply to reproducing or transmitting the document in any form: 1) all copies or portions thereof must identify the document's IEEE project number and draft number, and must be accompanied by this entire notice in a prominent location; 2) no portion of this document may be redistributed in any modified or abridged form without the prior approval of the IEEE Standards Department.

Other entities seeking permission to reproduce this document, or any portion thereof, for standardization or other activities, must contact the IEEE Standards Department for the appropriate license.

Use of information contained in this unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
+1 (908) 562-3800
+1 (908) 562-1571 [FAX]

unauthorized copying of software, and regular software audits should include a check for any unauthorized copies.

5.3.4.3 Print Output and Distribution Services

Output and distribution packages control output production and distribution from the moment the output is planned to the time the user receives the print. The working criteria need to be set up first; e.g., define who receives the report and how much of the report the user gets.

The main functions are

— The report can be limited to parts wanted by the user.

— Multiple copies of the entire report, or of selected sections can be produced.

— Reports are grouped by recipient within delivery location.

— Reports for each job are spooled as a group when the job is complete.

— The number of whole reports and individual pages received by each user are recorded.

— Report production can be monitored and managed efficiently.

Output and Distribution packages should contain the following minimum characteristics:

— Printing and distribution of whole and part reports

— Status (queued, printing etc) of the report tracked

— Online viewing of reports

— Ability to archive report files

— Ability to support a wide range of printers

— Costing and charging functionality

— Security facilities

By using an output distribution package, the delivery of reports to the correct person at the correct location can be ensured. Paper, time, and IT resource are saved as the users receive only the parts of reports that they need, and can also view the reports online. The number of pages printed can be controlled. Reports can be tracked from the time they are created to the time they are delivered to the user, allowing good security monitoring.

5.3.4.4 Office Media Management and Backup/Restore

The main functions of magnetic tape and data cartridge management systems are:

— To provide automated support for tape housekeeping and maintenance including:

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

924 • Allocating tapes and releasing them for reuse helping D
925 • To ensure even patterns of use where appropriate D
926 • Constructing and triggering cleaning schedules D
927 • Maintaining the security of data D
928 — To help automate archiving (vault management) for offsite storage D
929 — To help identify growth requirements D

930 Vault management is concerned with controlling the movement of tape cycles D
931 from one storage location to another. As a tape cycle is used, the tape manage- D
932 ment system automatically logs a different vault identifier against each tape. D

933 A backup strategy is required to control the frequency of backups and the way in D
934 which they are created; e.g., whole volumes to cartridge or individual files to tape. D

935 The backups and restores of system and application software should be separate D
936 from the backups and restores of data. Software and library backups should be D
937 explicitly scheduled and the complete software item or library backed up. The D
938 schedule for backing up files must be fully documented, properly maintained and D
939 adequately safeguarded as the contents of the schedule are required for disaster D
940 recovery purposes. D

941 #### 5.3.4.5 Online Disk Management D

942 The operation of disk management systems requires that they take account of a D
943 range of factors such as retention period, recovery, space fragmentation, disk D
944 overflow, file and record activity levels, and channel use. Some systems merely D
945 report against values or thresholds set, but increasingly they invoke corrective D
946 action. Typically, the corrective action is file and disk reorganization or file and D
947 data archiving. D

948 If a disk management system is used, the constant monitoring and actioning of D
949 requests for disk space can be minimized. Disk space may be collectively pooled D
950 and unused space constantly reclaimed. D

951 #### 5.3.4.6 Job Scheduling D

952 Scheduling involves the continuous organization of jobs and processes into the D
953 most efficient sequence, maximizing throughput and utilization to meet the tar- D
954 gets set in service level agreements (SLA). Jobs are scheduled to ensure: D

955 — SLAs and user requirements are met; e.g., certain jobs need to be run by a D
956 certain time D

957 — Available capacity is used effectively; e.g., the workload run at any given D
958 time does not exceed the practical capacity. D

959 The minimum functionality of a scheduler should include: D

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

960 — A high upper limit for the number of relationships allowed between jobs D
961 — The ability to schedule by calendar and criteria D
962 — Workload balancing support D
963 — Levels of security D
964 — Ability to restart jobs D
965 — Operator override capability D
966 — Capability to model future workloads. D

967 **5.3.4.7 Processor Configuration Management** D

968 Configuration management consists of four basic functions: identification, con- D
969 trol, status accounting, and verification. D

970 Identification involves specifying and identifying all components of an IT infras- D
971 tructure. D

972 Control implies the ability to agree and ''freeze'' configuration items (CIs) and D
973 then to make changes only with agreement of the appropriate named authorities. D
974 Control is concerned with ensuring that none of the CIs shown is altered or D
975 replaced, and that no CIs are added without appropriate authorization. D

976 Status accounting involves the recording and reporting of all current and histori- D
977 cal data concerned with each CI. Status accounting maintains records of the D
978 current, previous and planned states and attributes of the CIs and tracks these D
979 states and attributes: for example, as the status of a CI changes from ''develop- D
980 ment'' through to ''test,'' ''scheduled to go live,'' ''live,''and through to ''archived.'' D

981 Verification consists of a series of reviews and audits to ensure that there is con- D
982 formity between all CIs and the authorized state of CIs as recorded in the D
983 configuration management database (CMDB). It is concerned with checking that D
984 the physical CIs actually match the authorized system as described in the CMDB. D

985 **5.3.4.8 Network Configuration Management** D

986 To ensure the viability of network services the configuration of systems and ser- D
987 vices must be controlled and managed. Effective configuration management will D
988 produce a minimum risk environment. D

989 Configuration management procedures must ensure that details are provided for D
990 network equipment and systems covering: D

991 — Configuration activities—how to configure the network equipment D
992 — Security controls D
993 — Access controls D
994 — Configuration history log D

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

995 — Configuration authority D
996 — Build details D
997 — Fall-back and test records D
998 — Management reporting requirements. D

999 **5.3.4.9 Distributed System Configuration Management**

1000 — authentication services for a distributed system environment
1001 — distributed Naming Service Configuration
1002 — distributed Time Service Configuration
1003 — X Window system configuration
1004 — Window/Session Manager configuration

1005 **5.3.4.10 Accounting** D

1006 An effective cost management system should: D

1007 — Provide assistance in developing a sound investment strategy which recog- D
1008 nizes and evaluates the options and flexibility available from modern tech- D
1009 nology. D

1010 — Provide targets for performance and measure that performance against tar- D
1011 gets. It should also enable management to monitor actual costs against D
1012 budget. D

1013 — Facilitate prioritization of resource usage. D

1014 — Ensure the sound stewardship of all the assets employed in the organiza- D
1015 tion, including the maintenance of appropriate records for control purposes. D

1016 — Assist management in their day to day decision making and enable them to D
1017 make considered decisions as regards forward plans with the minimum of D
1018 risk. D

1019 — Be flexible and capable of providing a fast response to changing business D
1020 circumstances. D

1021 The essential objectives of apportioning costs are to: D

1022 — Recover from users the full cost of (IT) services, including the cost of capital. D

1023 — Ensure that IT users are aware of the costs which they impose on the IT D
1024 systems. D

1025 — Ensure that IT providers have an incentive to deliver an agreed quality and D
1026 quantity of economic and effective service. D

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.3.4.11 Performance Management

— ensure that users have satisfactory performance, workload throughput, and terminal response times in accordance with their service level agreements

5.3.4.12 Capacity Management D

An effective and efficient capacity management function contains at least the following elements: D D

— Performance management to monitor and optimize the use of current systems. D D

— A capacity management database that contains current and historic data of technical and business related interest. This database forms the basis for the provision of both tactical and strategic reports on performance and capacity. D D D D

— Workload management to identify and understand the applications that make use of the system. The understanding of workloads has both a technical and business related nature. This involves application sizing to accurately predict the performance and required capacity of new applications. D D D D

— Capacity planning to accurately plan the required hardware resource and associated cost for the future and to predict the effect on performance and capacity of both tactical and strategic plans. D D D

5.3.4.13 Availability Management

— measures taken to ensure availability and reliability

— automated or manual measures taken to recover a failed system

5.3.4.14 Security Management

— configuration of appropriate ACLs for System, User Interface, Storage, Network, and application software services.

5.3.5 Standards, Specifications, and Gaps

To be provided.

5.3.6 OSE Cross-Category Services

— Security for remote print jobs

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.3.7 Related Standards

To be provided.

5.3.8 Open Issues

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.4 Fault Management

Responsibility: Rich Bergman

The trade-offs in this subclause involve:

— Testability and Verifiability vs. simplicity and time

— Confidence and reliability vs. cost vs. risk

— Downtime vs. availability

— Maintainability vs. higher system cost

These services allow the system to react to the loss or incorrect operation of system components at various levels (hardware, logical, services, etc.). The classical model of fault tolerance has a three-step approach. The three steps are fault detection, fault isolation, and fault recovery. Typically implementations divide these steps into multiple steps or integrate them into one or two steps. Additionally, fault diagnosis services support the other steps in the treatment of a fault.

Various fault tolerance strategies, such as checkpointing and voting, are implemented as a collection of services comprising one or more of the steps in the fault tolerance classical model. For example, services involved in implementing a three-node voting scheme will include a vote comparator service (fault detection), vote analyzer service (fault isolation/fault diagnosis), a service to pass the majority ‘‘answer’’ through (fault recovery) as well as a service to disable the faulty resource and reconfigure the voters (fault recovery/reconfiguration).

Fault Detection

Fault detection services are concerned with determining when a fault has occurred in the system. Fault detection services are both passive and active. Active services are those that attempt to determine the status of various system components by testing those components. Passive services, on the other hand, try to ascertain system components by passively gathering information and watching the behavior of the system.

Fault Isolation;

Fault isolation services attempt to determine the component at fault and segregate the faulty component from the rest of the system. Services may be shared between the fault detection and isolation service library in that they perform both functions.

Fault Recovery

Fault recovery services attempt to bring the system into a consistent state. These services may be very interrelated to the scheduling services, network services, and data base services, depending on the recovery scheme used.

Redundancy of resources is many times needed to support fault recovery. Resources may include data, process, processor, disk drive, etc.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

As parts of the system fail, it may no longer be possible to satisfy all the requirements of the application. Services to support graceful degradation may be used to ensure that critical activities do not fail.

Fault Diagnosis

These services deal with the system's ability to analyze the attributes of a system fault and determine its cause. These services tend to be very interrelated with fault detection and fault isolation services.

Fault Avoidance

These services involve the avoidance of faults before a failure in the system component occurs. If a system can detect that the operation of a component is approaching the edge of its operational range, a standby or backup component could be phased in to replace it. Another form of fault avoidance is logging of shocks, temperature extremes, etc., so that it can be predicted that a component will not meet its original expected service life.

Software Safety

These services involve the system's ability to keep application software from causing harm to the system's software, hardware, or user. For instance, a process may attempt to write into another process's memory space without permission.

A good example of a reliability method that may provide software safety is a bounds checker. The checker compares an answer supplied against the bounds. If it is not within the bounds, the bounds checker will not allow the answer to propagate, possibly causing damage to the system's integrity. Additionally, it may send a fault message (or security violation information, depending on the type of answers expected) to the proper service.

To enhance software safety, other services and processes should be only given the resources necessary to complete their job.

Status of System Components

These services involve the obtrusive and nonobtrusive diagnosis of the state of system components. For further explanation of these services, see Fault Detection and Fault Diagnosis services. These services may additionally need to record and/or display information concerning performance, configuration, and general system information.

Reconfiguration

These services allow the system to reconfigure its view of the world. This services allow the system to substitute different resources to perform system functions such as substituting a new physical I/O channel to support a logical channel. These services are part of the API but their use may be restricted to specially authorized programs such as those used by the target system operator.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Maintainability

Maintainability services provide support for the maintenance of a system. A major component of that support is the collection and logging of information about the operation of the system. Typical information to be logged is:

— Software and hardware errors during operation

— Processes that failed or almost failed to meet scheduled deadlines

— Performance metrics for system tuning

— Times when the system operated in extreme environmental conditions

— Errors reported during startup self-testing

— Attempts to violate rules of the system's security policy.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Section 6: Profiles

Responsibility: Fritz Schulz

This section targets those who want to know more about what profiles are and those who are in the process of developing their own profiles. The latter group consists of those developing formal ''Standardized Profiles'' and those developing less formal profiles for their industry group (e.g., a banking trade association) or their own company or enterprise for procurement or strategic planning purposes.

Those not involved in the development of profiles should read 6.2. Parts of 6.3 also may be useful, especially the earlier subclauses that give definitions of terms and explain concepts more precisely.

Developers of profiles that are not formal POSIX Standardized Profiles (POSIX SPs) should read all of Section 6.

Developers of profiles that are formal POSIX SPs should read all of Section 6 and Annex A.

6.1 Scope

The information presented here about profiles is limited in scope to assist those needing to understand profile concepts as they apply to the POSIX Open System Environment. Covered are profiles constructed from standards (and profiles) listed within this guide (that, by design, are consistent with POSIX.1).

The goal is to create a common approach and documentation scope and style for POSIX-oriented profiles. Annex A goes further by giving specific guidance to developers of formal POSIX SPs.

6.2 Profile Concepts

Responsibility: Bob Gambrel

Introduction

This guide is designed to assist in the selection of standards in the procurement process or as a target application environment. Profiles also assist in the selection of standards. A profile is a suite of base standards with specified options. Profiles can be created by software developers to describe the environment they

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

target or by buyers to identify their purchasing objectives.

Basic Terminology

See 2.2.2 for relevant terms that will be used in this section: base standard, application environment profile, profile, standardized profile, and POSIX standardized profile.

There are two general classes of standards documents:

— Base standards

— Profiles (including AEPs, standardized profiles, and POSIX SPs)

As used in this guide, base standards specify functionality, syntax, protocols, data formats, etc., in detail, while profiles do not. Instead, profiles identify which base standards are applicable. Since base standards often consist of a base or mandatory part and a number of selectable optional parts and values, profiles may also (or may not) choose, for each base standard, specific options or values. A profile may also identify other profiles, allowing the construction of ''larger'' profiles based on both base standards and other ''smaller'' profiles.

NOTE: Profiles go by many names throughout the world including ''functional standards,'' ''implementation agreements,'' and ''specifications.'' In the context of internationalization, the term ''national profile'' is frequently used and will be found, for example, in POSIX.1 {2} and POSIX.2 {*ref*}. Its meaning is consistent with the definitions in 2.2.2, but in many cases such profiles reflect national cultural conventions. For example, Denmark and Japan both have specified a national character profile. D

6.2.1 Relationships Between This Guide and Profiles

Key to the understanding of profiles is a discussion of the relationships that exist among profiles, this guide, and the base standards.

There exist many thousands of base standards, each addressing a particular, usually narrowly scoped, area of application portability or interoperability. Many of the base standards, developed over the years, are simultaneously narrow in scope (for example, a C binding of SQL), but broadly applicable (for example, applicable to Operating Systems that have 'POSIX''interfaces and those that do not.)

The base standards listed in 1.2 form the basis of the POSIX Open System Environment. The list is comprehensive, in that its coverage is broad enough to cover most modern day application development, and the base standards selected have been determined to be consistent with POSIX.1 {2}. D

While this guide does not list all base standards, it is still a large list, and in fact the list contains base standards that might not be consistent with each other (choose any two standards from the POSIX OSE and they might not be consistent with each other.) The process of profile writing addresses this.

The profile writer reduces even further the list of base standards to just the (relatively) few that are needed to provide portability and interoperability in a given

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

functional area. In the process, the profile writer grapples with the coherence of the selected base standards by choosing only those that will work together to get the particular job done. Profile writers should also deal with *harmonization,* which means making the profiles consistent with each other where they overlap. This can often be done among profiles even where the functional areas served differ greatly. Procurements specifying two profiles that have been harmonized by their authors have the benefit of knowing that the two will not conflict with each other.

By specifying compliance to a particular profile in a procurement, a consumer easily references a set of multiple base standards that have been determined to: serve a particular purpose; work together; and be consistent with POSIX .1 {2}.

The benefits and relationships do not end here, however. Since profiles can be constructed to reference profiles as well as base standards, future profile writing will be even easier.

NOTE: An analogy is in the construction of electronic equipment such as computers. The basic building blocks are ‘‘components,’’ such as memory chips and capacitors, which can be fabricated into larger building blocks such as printed circuit boards, which can be fabricated (with other components or printed circuit boards) into larger building blocks, such as standalone computers, which can be fabricated into larger building blocks such as department wide networks of computers, etc. Likewise, a few base standards (the basic building blocks), can be gathered together into ‘‘component’’ profiles, which can then be gathered together (with other base standards or component profiles) into larger ‘‘platform’’ profiles, which can be gathered together into larger ‘‘application area’’ profiles. (See 6.3.3.5.)

The development of profiles from the primary building blocks (base standards) results in larger building blocks (profiles) that can then be incorporated into future profiles and also into future versions of this guide.

The Importance Of Profiles

Profiles are important for a number of reasons:

— Profiles select one or more base standards or profiles and specify options and parameters within these. This provides a clear statement of specifications that describe the standards for the target functional objective(s).

— Profiles include information about the relationship between the standards included (i.e., coherency is an objective).

— Profiles are a clear method of communication about the specific standards needed for an application domain and can be used in procurement, in conformance testing, and as a target for applications development.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

6.3 Guidance to Profile Writers

Responsibility: Bob Gambrel

This clause expands the concept of profiling in the manner needed by profile writers and provides detailed guidance to those writers. It includes a description of the basis for this guidance, expands on the purposes served by profiles, and finishes with more detailed guidance specifically aimed at those writing profiles.

Using this guide as a basis, profile writers can develop their own informal profiles, suited to their own needs, or formal standards bodies can develop formal, balloted profiles. This clause details the requirements that should be met by developers of profiles whether they are POSIX SPs, standardized profiles, or less formal profiles. Standardized profiles are formal profiles that meet the requirements of a sponsoring standards body. Standardized profiles that also meet the requirements for POSIX-based profiles (rules established by IEEE) are called POSIX standardized profiles (POSIX SPs.) For more information about writing POSIX SPs, see Annex A.

6.3.1 Basis for This Guidance

Many of the ideas and concepts for profiling described in this section derive from the work of ISO/IEC JTC 1 SGFS as documented in ISO/IEC TR 10000-1 {*ref*}. Some items specified in that document that are not covered here include:

— International standardization considerations

— Conformance issues

— Processes and procedures

— Maintenance

— Taxonomy

Additionally, some consideration was given in this guidance above and beyond that given in ISO/IEC TR 10000 {*ref*}:

— Standardized profiles and POSIX standardized profiles as a conceptual extension to International Standardized Profiles (ISP).

— IEEE basis, not ISO basis, for formatting rules; see Annex A

Writers of profiles following the guidance of this clause should refer to Annex A if they intend to propose IEEE acceptance as a POSIX SP and to ISO/IEC TR 10000 {*ref*} if they intend to propose acceptance as an ISP.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

6.3.2 Purpose of Profiles

Profiles define combinations of base standards and profiles for the purpose of:

— Identifying the base standards, together with appropriate classes, subsets, options, and parameters, that are necessary to accomplish identified functions for purposes such as interoperability and portability.

— Providing a system of referencing the various uses of base standards that is meaningful to both users and suppliers

— Enhancing the availability for procurement of consistent implementations of functionally defined groups of base standards that are expected to be the major components of real application systems

— Promoting uniformity in the development of conformance tests for systems that implement the functions associated with the profiles

6.3.3 Detailed Guidance to Profile Writers

6.3.3.1 The Relationship to Base Standards

Base standards specify procedures and formats that facilitate application portability and interoperability. They provide options, anticipating the needs of a variety of applications and taking into account different capabilities of real systems and networks.

Profiles further promote portability and interoperability by defining how to use a combination of base standards for a given function or application area. Profiles, by definition, do not define new application interfaces.

In addition to the selection of base standards, a choice may be made of permitted options for each base standard and of suitable values for parameters left unspecified in the base standard.

Profiles should not contradict base standards, but should make specific choices where options and ranges of values are available. Profiles must include all of the items made ‘‘mandatory’’ by the standard. The choice of the base standard options should be restricted so as to maximize the probability of interworking between systems implementing different selections of such profile options, consistent with achieving the objectives of the profile.

A profile makes explicit the relationships between a set of base standards used together (relationships that are implicit in the definitions of the Base Documents themselves) and may also specify particular details of each base standard being used.

A profile may contain conformance requirements that are more specific and limited in scope than those of the base standards to which it refers. While the capabilities and behavior specified in a profile will always be valid in terms of the Base Documents, a profile may exclude some valid optional capabilities and optional behavior permitted in those base standards.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Thus, conformance to a profile implies, by definition, conformance to the set of base standards that it references. However, conformance to that set of Base Documents does not necessarily imply conformance to the profile.

6.3.3.2 Main Elements of a Profile Definition Document

The definition of a profile should comprise the following elements:

— A concise definition of the scope of the function for which the profile is created and of its purpose

— Reference to a set of base standards and other profiles, including precise identification of the actual texts of the base standards and profiles being used and of any approved amendments and technical errata, conformance to which is identified as potentially having an impact on achieving portability and interoperation using the profile

— Specifications of the application of each referenced base standard and profile, covering recommendations on the choice of classes or subsets and on the selection of options, ranges of parameter values, etc.

— A statement defining the requirements to be observed by systems claiming conformance to this profile, including any remaining permitted options of the referenced base standards and profiles, which thus become options of this profile

Systems that interoperate can perform different but complementary roles (e.g., an initiator-responder or a master-slave relationship). In such a situation the profile should identify the separate roles that may be adopted by a system, and these should be stated as either mandatory requirements or options of the profile, as appropriate.

6.3.3.3 Profile Objectives

Completeness

A profile should be complete with respect to its functionality objectives. This may well be an iterative process, since the understanding of the requirements and standards will evolve. Completeness means that all areas where standards should be applied have been identified and the requirements defined. Where standards exist, they have been included, and the options within those standards have been addressed. Where standards do not exist, but are needed, this has been documented in the profile.

It may be appropriate to document (probably in a nonnormative appendix) specifications and alternatives available in areas where standards have not been defined. The meaning of this concept will be relative to the forum for acceptance of the profile. If the profile is targeted at ISO acceptance, then ISO DIS and IS standards should be the reference point, where as a US Government profile might be focused on FIPS and ANSI standards. Within private industry, consortium and even vendor specific specifications could be incorporated, keeping these as

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

examples and not explicit requirements, which will simplify harmonization with formal standards as they emerge. Where standardized profiles are being developed and gaps are identified, the profile writer should identify the requirements that are not satisfied by a standard. If there is a preliminary specification available that addresses many of the requirements, that specification should be referred to informatively.

Clear Communications

A key objective for the profile is clear communications between the affected parties. Users, software developers, and platform suppliers all need to have the same terms and specifications. The application software developers and system vendors need a common set of specifications to target for their development efforts.

Harmonization

Harmonization means making the profiles consistent with each other where they overlap. This can often be done among profiles even where the functional areas served differ greatly. This assures that the maximum practical agreement exists between different profiles, maximizing the implementations of that common ground.

Validation

A profile addresses validation in two different ways.

Firstly, by selecting options and parameters within the profile, validation is potentially made simpler.

Secondly, by including more than one base standard, validation potentially becomes more difficult. Now validation extends beyond just insuring a single standard is being complied with into the area of insuring that the interactions between and among multiple base standards is also being complied with.

Coherence

The simple selection of a group of standards does not assure that they will work together on a platform in a predictable way. A profile should contain a matrix of all standard components compared to each other and state what relationship exists between them. A profile may be coherent if it states that between two standards no relationship needs to exist, that none shall exist, or that a specified relation shall exist. Not to speak to an intersection in the matrix would indicate that the issue of coherence has not been addressed.

Gap Identification

In the process of developing profiles, there may be gaps in coverage by standards that become apparent. These may exist in terms of the characteristics available with one standard that need to be made available from another, or missing standards, or additional functionality that is needed for a specific applications

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

activity. So, an additional objective for a profile effort is to document the requirements for such additional work and forward it to the appropriate standards effort. Profile groups in industry should consider providing expertise to the associated standards groups to assure that the resulting standards meet the needs of that applications area.

6.3.3.4 Defining a Profile

The process of creating a specific profile is broken down into three stages. These are the creation of a generic information system model, creation of an open system environment that supports application portability and interoperability, and the creation of a application-area-specific profile (see Figure 6-1).

In the first stage, a generic model for an information system is constructed by abstracting the properties and functionality common to all possible information systems and choosing specific system features based on the application requirements of an enterprise, department, or application. Such a model is neither new nor earthshaking; it has been used for more than twenty years to understand, design, and specify a system, as well as to test, redesign, and refine that system. What is new is the refinement of generic models to design open systems for specific application areas.

The second stage of the profile creation process requires the profile developer to identify and choose the functionality, standards or specifications, and interfaces required for application portability and interoperability. The profile developer should probably choose at most one base standard for each service identified. The criteria used for selecting base standards should take into account openness, completeness (maturity), stability, geographic coverage and publicness of the standard. The profile writer should also have in mind a precedence (e.g., international standards take precedence over national standards). There may also be other considerations, of course, such as performance, cost, design, etc. Examples of such considerations, that were used in selecting standards for this guide, can be found in 3.4.2.

To ensure that the open system environment produced is not just a paper standard, it is necessary to check that the component standards and specifications chosen are practical for users, marketable for vendors, integratable with each other, achievable in a short time, and have a good likelihood of widespread acceptance. The X/Open Common Application Environment and the IEEE POSIX. 0 POSIX Open System Environment have been (or are being) produced in this way.

The third stage of the profile-creation process requires making further choices of functionality, standards, mandatory options within standards, subsets of standards, and interfaces to meet the open-systems requirements of a specific application area. As with open system environments, standards for a profile should be chosen based on international standards, national standards, formal professional group standards, consensus-defined consortia specifications, and de facto specifications. Also, where no standards exist for a particular functionality necessary to the application area in question, the profile developer should come up with a plan to produce the missing standard. In addition, the profile's component

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

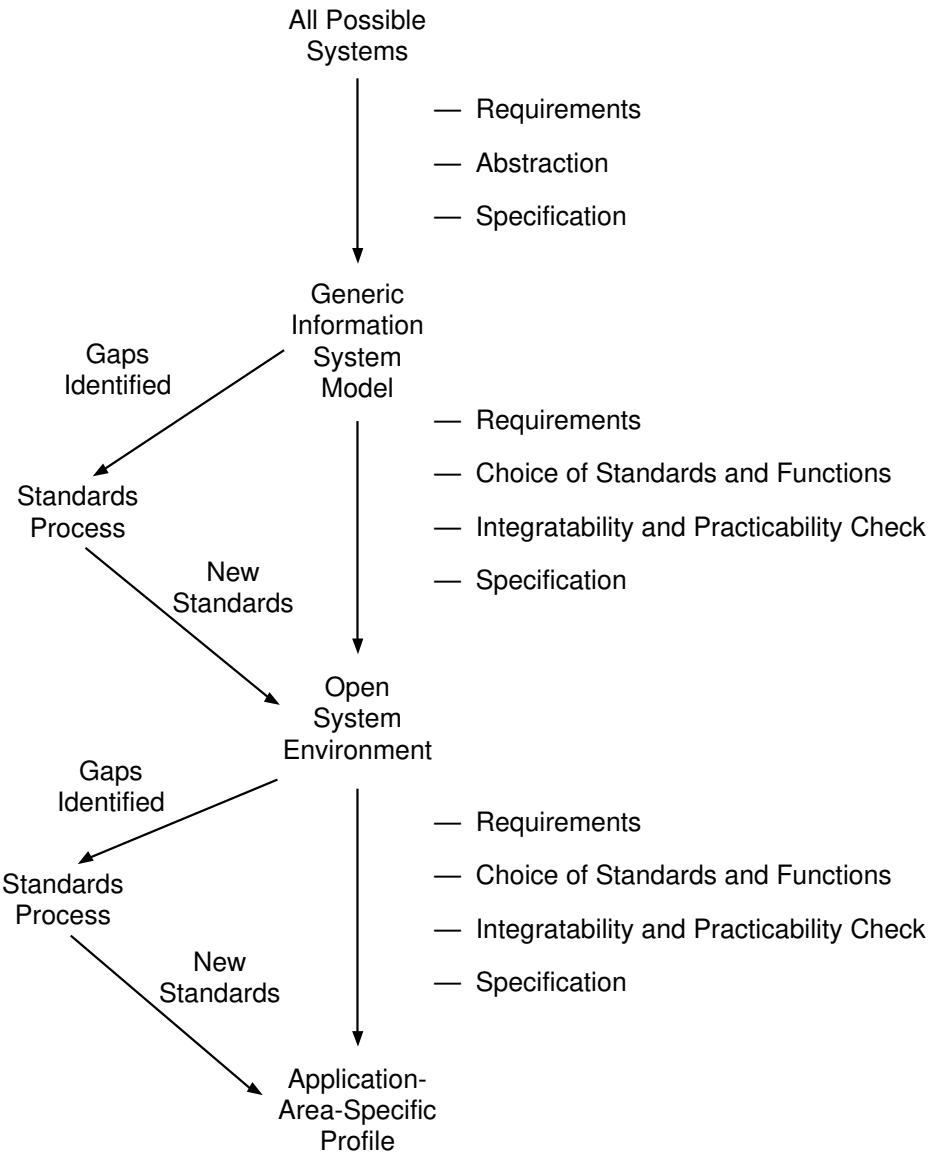

Figure 6-1 – Development of a Profile

standards should be selected bearing in mind practicality, marketability, integration of component standards with each other, and user and vendor acceptance.

The choice of functions and standards applicable to an application area results in a standardized application-area-specific profile. The MAP, TOP, and GOSIP profiles of OSI, as well as the POSIX. 10 (supercomputing), POSIX. 11 (transaction processing), and POSIX. 13 (embedded realtime) profiles have been (or are being) defined in this way.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

6.3.3.5 Types of Profiles

Three different types of profiles have been, or are being, defined by the procedures described above:

— Component Profiles

— Application Area Profiles

— Platform Profiles

A Component profile is mostly a subset of a single standard. The profile developers specify mandatory options for a specific domain, options that are not desirable for that domain, gaps in that parent standard, and, if necessary, specifications to fill that gap. Examples of such profiles are MAP, TOP, and GOSIP profiles and possibly the POSIX.13 embedded realtime POSIX profile if it continues to be based exclusively on functions chosen from the POSIX.4 realtime standard. D

An Application Area Profile is created from multiple standards that specify multiple, diverse types of functionality needed for a particular application area (e.g., database, networking, graphics, operating system). The application area profile developers specify all the diverse standards necessary for the application area in question. Within each standard, they identify mandatory options, functions and options that are not needed, gaps in the standards, and, if necessary, specifications to fill the gaps. Examples of application area profiles are the POSIX.10 supercomputing and POSIX.11 transaction processing profiles.

A Platform Profile focuses on the functionality and interfaces needed for a particular type of platform. The platforms could be traditional platforms (such as time sharing systems) or relatively new or emerging platforms (e.g., workstations, personal computers, or symmetric multiprocessing systems). A platform profile could be created from one or multiple diverse standards. As with other types of profiles, the profile developers have to specify the standards, options, standards gaps, and if necessary, specifications to fill the gaps. Examples of platform profiles are the POSIX.18 Platform Profile for Traditional Multiuser UNIX systems and the POSIX.14 Multiprocessing profile.

All three types of profiles can be seen in the next section.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.