--

--

IEEE P1003.0 Draft 14 – November 1991

5.2.5.2 Emerging Standards

Information Retrieval, Transfer and Management For OSI—Draft Access Control Framework, ISO / IEC SC 21/ WG 1.

Draft Addendum to ISO 8613 On Security

The P1003.6 scope is limited to security extensions for those interfaces defined within the base POSIX interface specification (POSIX .1 {2}). Issues not addressed within the P1003.6 scope include noninterface-specific architectural assurance issues and communications security.

5.2.5.3 Gaps in Available Standards

The Information Technology Security Evaluation Criteria, Version 1.2, 28 June 1991.

US DoD, DOD 5200.28- STD, *Trusted Computer System Evaluation Criteria.*

Trusted Network Interpretation

Trusted Database Interpretation

Computer Security Subsystem Interpretation

5.3 Information System Management

Responsibility: Don Folland, Neil Croft

5.3.1 Overview and Rationale

Information System Management issues are considered in this clause. The subject is concerned with the effective management and control of the complete set of resources that comprise an information system. The tools in support of the services required by system managers need to reflect the portability and interworking attributes of open systems and fit the Open System Environment Reference Model (Figure 3-3). It is necessary to consider a variety of system management support scenarios (central management, dispersed management, or hybrid), addressing both distributed systems and standalone systems. The issues apply to application software or software components of the application platform. It is necessary to support automated management and operation of the IT infrastructure and address a wide variety of licensing scenarios.

5.3.2 Scope

This category includes services and policies that address the administration of the overall information system required by any organization, including:

- Information Management
- Processor Management (e.g., Add new user)
- Network Management
- Configuration Management
- Security Management (e.g., Authentication, Key Management)
- Accounting Management
- Performance Management

Administration services accessible from the API may have Programming Language or Language Binding service specifications associated with them.

These services are defined to provide system and network administrator portability.

5.3.3 Reference Model

The Reference Model for system management is the same as the model shown in Figure 3-3. System management impacts all of the APIs and EEIs in the POSIX Open System Environment Reference Model.

5.3.4 Service Requirements

The following services should be provided: E

5.3.4.1 Processor Configuration Management

Configuration management consists of four basic functions: identification, control, status accounting, and verification.

Identification involves specifying and identifying all components of an IT infrastructure.

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

Status accounting involves the recording and reporting of all current and historical data concerned with each CI. Status accounting maintains records of the current, previous and planned states and attributes of the CIs and tracks these states and attributes: for example, as the status of a CI changes from ''development'' through to ''test,'' ''scheduled to go live,'' ''live,''and through to ''archived.''

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

5.3.4.2 Network Configuration Management

To ensure the viability of network services the configuration of systems and services must be controlled and managed. Effective configuration management will produce a minimum risk environment.

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

- — Configuration activities—how to configure the network equipment
- — Security controls
- — Access controls
- — Configuration history log
- — Configuration authority
- — Build details
- — Fall-back and test records
- — Management reporting requirements.

5.3.4.3 Distributed System Configuration Management

The services here consist of the following:

- Authentication services for a distributed system environment
- Distributed Naming Service Configuration
- Distributed Time Service Configuration
- X Window system configuration
- Window/Session Manager configuration

5.3.4.4 Software Installation and Distribution

The main types of software to be installed and distributed are application programs developed in-house, bought-in applications, and utility software and personal computer software packages. All software needs to be managed effectively from development or purchase through to the live environment. Unless the distribution and implementation process can be controlled automatically, or from the center using software tools, procedures must be in place to ensure that distributed software arrives when expected and is checked for authenticity in whatever way is practical, and that the software is brought into use when required. The main procedures involved in software distribution and installation are:

- System management staff at the center to inform remote staff when to expect distribution software to arrive.
- Recipients to report to system management staff when the distributed software has arrived successfully.
- System management staff to check that all software is received as expected at locations.
- System management staff to issue clear instructions about when the software is to be implemented.
- Location staff to report to system management at the center when the software has been implemented. The release record on the Configuration Management Database will state which installations are to receive the release. This database must be updated to reflect the receipt and implementation of the release at each site.

5.3.4.5 License Services

The terms and conditions relating to the supply of software may place legal restrictions on the organization (e.g., no unauthorized copies to be made). It is particularly important therefore that the Configuration Management Database is updated with details of who holds copies of software items. This assists the organization in discharging its legal obligations and assists auditors in checking for the existence of unauthorized copies.

All authorized copies of licensed or purchased software that are made by system management staff should be allocated a unique copy number and recorded in the Configuration Management Database together with where they are located and who is responsible for them. Procedural restrictions should be introduced to prohibit the unauthorized copying of software, and regular software audits should include a check for any unauthorized copies.

5.3.4.6 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 include the following:

- — 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.

5.3.4.7 Office Media Management and Backup/Restore

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

— Provide automated support for tape housekeeping and maintenance including:

- Allocating tapes and releasing them for reuse helping
- To ensure even patterns of use where appropriate
- Constructing and triggering cleaning schedules
- Maintaining the security of data

— Help automate archiving (vault management) for offsite storage

— Help identify growth requirements

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

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

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

5.3.4.8 Online Disk Management

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

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

5.3.4.9 Job Scheduling

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

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

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

The minimum services of a scheduler should include:

- A high upper limit for the number of relationships allowed between jobs
- The ability to schedule by calendar and criteria
- Workload balancing support
- Levels of security
- Ability to restart jobs
- Operator override capability
- Capability to model future workloads.

5.3.4.10 User Administration

The services here consist of the ability to:

- Create a new user or group of users
- Delete a user or group of users
- Allocate system resources to a user or a group of users

5.3.4.11 Accounting

An effective cost management system should contribute to the development of a sound investment strategy that recognizes and evaluates the options and flexibility available from modern technology. The services here should provide the ability to:

- Establish targets for performance
- Measure performance against targets
- Measure and prioritize resource usage
- Monitor assets and maintain records for control purposes
- Apportion costs of IT services to users
- Report costs to management and users

5.3.4.12 Performance Management

The services here should provide the ability to:

- Monitor hardware, software, and network performance
- Monitor workload and throughput
- Set and adjust system parameters to tune performance

— Monitor terminal response time E

5.3.4.13 Capacity Management

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

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

5.3.4.14 Fault Management E

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.

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.

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.

5.3.4.15 Security Management

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

5.3.5 Standards, Specifications, and Gaps

There are a number of international and national initiatives to develop standards for system management. E E

Note to reviewers: This subclause will be expanded in a later draft. E

5.3.6 OSE Cross-Category Services

— Security for remote print jobs

5.3.7 Related Standards

None. E

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 target or by buyers to identify their purchasing objectives.

Basic Terminology

There are two general classes of standards documents:

- Base standards
- Profiles, including application environment profiles (AEP), standardized profiles, and POSIX standardized profiles

See 2.2.2 for format definitions of these terms. As used in this guide, base standards specify functionality, syntax, protocols, data formats, etc., in detail, while profiles do not. Instead, profiles (sometimes called ''functional standards'') 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: 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. 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.

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 comply with POSIX specifications 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}.

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 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*,[3] 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 and work together. E

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.

3) This should not be confused with *international harmonization*, which refers to a specific process that must be followed in the approval process for International Standardized Profiles (ISPs). E E

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.

Note to reviewers: Annex A has important information in relation to this section that should be reviewed.

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. 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:

— 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 if they intend to propose acceptance as an ISP.

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.

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 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 [4] 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.

---

4) Refer to the earlier footnote on *international harmonization*. E

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 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 Methods for Developing Profiles E

To Be Determined . E

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.

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