-- --

IEEE P1003.0 Draft 13 – September 1991

2.2.2.48 System Internal Interface (SII): The interface between application platform service components within that platform; it may be standardized or non-standard.

2.2.2.49 system services: Firmware and software that provide an aggregation of network element functions into a higher level function; provide an interface to the data contained in the system.

2.2.2.50 System Services API: An interface providing access to services associated with the application's internal resources.

The System Services API has two parts: Language Specifications and Processing Services API.

2.2.2.51 system software: Application-independent software that supports the running of application software.

2.2.2.52 transaction: A unit of work consisting of an arbitrary number of individual operations all of which will complete successfully (or be of no effect) on the intended resources.

A transaction has well defined boundaries. A transaction starts with a request from the application program and either completes successfully (commits) or has no effect (abort). Both the commit and abort signify a transaction completion.

2.2.2.53 transaction application program: A program written to meet the requirements of a chosen Transaction Processing (TP) application.

Such programs allow a sequence of operations that involve resources such as terminals and databases. The transaction AP specifies transaction boundaries. The transaction AP as defined here is a logical entity and may involve an arbitrary number of processes.

2.2.2.54 validation: The process of evaluating a ported application, software, or system to ensure compliance with requirements.

2.2.3 Abbreviations

For the purposes of this guide, the following abbreviations apply:

2.2.3.1 API: Application Program Interface

2.2.3.2 EEI: External Environment Interface

2.2.3.3 POSIX. 0: This guide.

2.2.3.4 POSIX. *n*: An IEEE POSIX working group, where the number *n* represents the decimal notation in the IEEE P1003 series. Alternatively, when apparent from context, the latest standard issued, or under development, by that working group.

2.2.3.5 SII: System Internal Interface. D

Section 3: POSIX Open System Environment

Responsibility: Fritz Schulz

The POSIX Open System Environment (OSE) is a collection of concepts that provide a context for user requirements and standards specification. It provides a minimum, standard set of conceptual information system building blocks with associated interfaces and functionality. The POSIX OSE consists of a reference model, service definitions, standards, and profiles.

These OSE concepts are also intended to be conventional within computer science. The intention is not to break new ground, but to establish a minimum and unambiguous terminology and set of concepts for identification and resolution of portability and interoperability issues.

The POSIX Open System Environment is defined in five parts:

(1) General requirements are identified that apply to the POSIX OSE as a whole in 3.1.

(2) A reference model is developed that unambiguously identifies the system under consideration for purposes of specification. The POSIX OSE reference model described in 3.2 defines system elements to identify interfaces across which service requirements should be satisfied. The elements are chosen to expose those interfaces that are significant to the profile writer or user.

(3) Using the interfaces identified in the reference models, each subclause of Section 4 categorizes and describes the basic services available to users across each interface. The services are defined in a generic way, based on the reference model, user requirements, and current industry practice, rather than any given implementation.

Definition of the service requirements is not constrained by the availability of standards. Service requirements that are not currently satisfied via standards are discussed in either the Emerging Standards subclause, or under Gaps.

Each clause of Section 4 begins with a more detailed and specialized version of the reference model to provide a context for service specification. After defining the interfaces and services, each of the Section 4 clauses concludes with a discussion of standards that are related to the services.

(4) Section 5 discusses issues and requirements that directly affect all of the service categories, such as internationalization, security, and