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All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change.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]

Annex E
(inform ative)

Additional Material

E.1 Software Development Environments

Responsibility: Don Folland

E.1.1 Overview and Rationale

Software Development Environments are dealt with as a particular application area needing special attention for the following reasons:

— The domain of Software Development Environments is one of prime importance. Software development is a major area of expenditure for government and large commercial organizations.

— The need for standardization is being driven not only by the SDE vendors and users, but also by the Independent tool developers who want to get their tool products on as many vendor platforms as possible.

— The SDE domain calls not only for portability, but also for particular integration and interoperability requirements.

— The domain is primarily of interest to that user community that has large complex software development requirements, but it is also of interest to all application areas as software development is an enabling technology for all applications.

Software engineers seek more powerful assistance to improve productivity and quality in the software development process. Considered opinion currently favors Project Support Environments (PSE) underpinned in such a way that the facilities are capable of being implemented on different machines. A PSE needs a base holding information such as specifications, designs, code, schedules, configuration plans, tests, etc., to support the developers. The interface between the base and the tools must ensure portability of the tools. Again, these tools will be supported by relevant language standards.

Certain design methodologies themselves have been modeled formally to establish their degree of rigor and self-consistency. Function Point Analysis is one method of measuring software systems and computing productivity that is increasing in use. It measures inputs, outputs, and entities accessed to determine transaction

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

size; it gauges technical complexity by reference to 19 characteristics. These are combined to give a measure of systems size. Productivity is the ratio of system size in function points to the effort required to produce or maintain the system.

Generally, software support for the development process is in its infancy and effective metrics have not yet been developed.

E.1.2 Scope

The problem domain is complex software development, from the generation of an idea to the delivery and ongoing support of a solution product set.

Thus, an SDE may include some or all of the following:

(1) Software Development Life Cycle
  (a) Requirements analysis
  (b) Logical design
  (c) Physical design
  (d) Functional and interface specification
(2) Activity support
  (a) Prototyping
  (b) Program development and testing
  (c) Quality assurance and regression testing
  (d) Generation of user documentation
  (e) User training
  (f) Problem report tracking and maintenance
  (g) Maintenance and tracking of schedules
(3) Configuration Management
  (a) Automatic version management
  (b) Integrity management
  (c) Traceability
(4) Project Management
(5) Data Administration
  (a) Access control

In the context of developing software for a POSIX Open System Environment, design will take account of portability and interoperability requirements. The SDE tools themselves should be portable. The software development activities may be provided with a large set of tools and applications. The SDE must provide the necessary support for the integration of all of these tools.

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

E.1.3 Reference Model

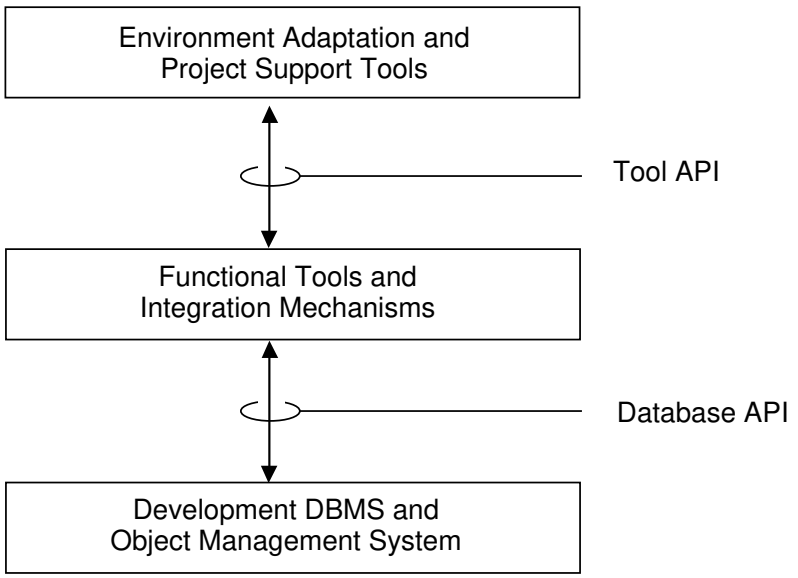

Figure E-1 – Software Development Model

In this clause the conceptual view of software development is related to the POSIX Reference Model (Figure 3-1). The software developer's view is shown in Figure E-1. The tools used to develop software can be viewed as applications in their own right in the context of the POSIX Reference Model, requiring the same services from the platform as for Database Management.

In the Software Development Model, the Environment Adaptation and Project Support Tools ''layer'' provides the essential link between the programmer, designer or analyst, the design method, and the development infrastructure. At this level are provided the tools and applications that are unique to the project or methodology; e.g., project management workbench. It requires support from a consistent human-computer interface to the Functional Tools.

The Functional Tools and Integration Mechanisms embrace the essential tool set to enable software developers to build software. It includes simple tools such as editors, tools for tool-building, and integration mechanisms. There will be tools for Configuration Management, Version Management, and System Administration. It is not within the scope of this guide to discuss these in detail.

The whole software development environment is underpinned by essential management systems, such as object management system, a data dictionary, a user interface management system, and environment management. A database will frequently be established to hold specifications, designs, configuration plans, etc.

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

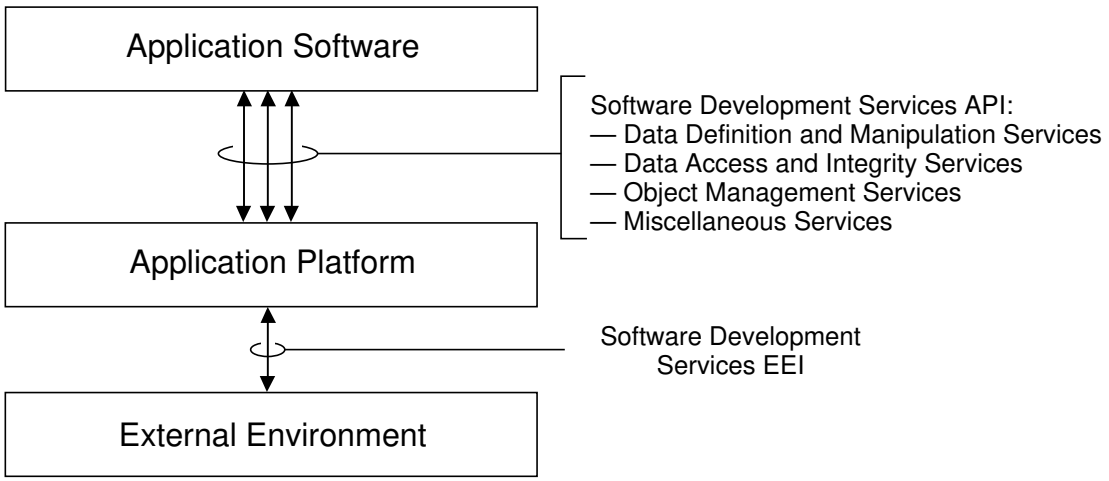

Figure E-2 – Software Development Reference Model

In the POSIX Open System Environment, the software development model can be incorporated into the POSIX Reference Model as in Figure E-2. The model shows that the tools and services required by the software developer are part of the POSIX Open System Environment and are available through the POSIX OSE API.

E.1.4 Services Requirements

Software developers, i.e., designers, analysts, and programmers, use software applications to facilitate the complex task of software development. A tool will require services from the application platform and will frequently require support from another application in the application set. There are many possible implementations of tool sets. Descriptions of these are beyond the scope of this guide.

E.1.4.1 Application Program Interface Services

The services required at the API are essentially similar to those described for Database Management in 4.4.4.1; i.e., Data Definition and Manipulation, Data Access, Data Integrity, and such Miscellaneous Services as Data Dictionary.

E.1.4.2 External Environment Interface Services

A consistent human-computer interface to the tool set is required. Some of the programmer's tool set will be explicitly focused on windowing services (such as 4.7 and 4.8) and provide assistance to develop software with improved usability.

Resource data formats must be specified in order to ensure effective information interchange [for example, CASE Data Interchange Format (CDIF)], for which standards are currently under development under the aegis of the CDIF Technical

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

Committee (see also E.1.5.2 and 4.5).

Protocol services are required for the transport of data (see 4.3).

E.1.4.3 Interapplication Software Entity Services

Many of the tools depend for interface between one another upon the data dictionary/repository, which is a key software component and which may conceptually be regarded as part of the Applications Platform. Included in this category will be utilities for servicing the DBMS, such as recovery, reorganization, and restructure:

— Object management system

— User interface management system

— Database management system

— Transaction processing management system

Details of these management systems may be recorded in the data dictionary/repository.

E.1.4.4 Software Development Resource Management Services

These services are generally not visible to the programmer or software developer at the Tools API, usually being provided by the tool building and other software development utilities.

E.1.5 Standards, Specifications, and Gaps

This subclause describes current accepted standards that are relevant to this area in addition to the language standards in 4.1.5 and the database standards in 4.4.5.

E.1.5.1 Current Standards

This subclause briefly identifies the current standards in this area.

The following provides place holders for further text to be inserted – assistance required please.

E.1.5.1.1 International Standards

Labeling and File Structure of Magnetic Media

The following standards refer to the labeling of magnetic media and for the file structure on such media to facilitate information interchange:

Labeling of magnetic tape	ISO 1001
Labeling of cassette and cartridge	ISO 4341

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

Table E-1 – Software Development Standards

Service	Specification	Subclause
Miscellaneous Services:		
Labeling of magnetic tape	ISO 1001	4.11.5.?
Labeling of cassette and cartridge	ISO 4341	4.11.5.?
Labeling of flexible disks	ISO 7665	4.11.5.?
Volume and file structure for flexible disks	ISO 9293	4.11.5.?
Volume and file structure for CD-ROM	ISO 9660	4.11.5.?
Documentation symbols and flowchart conventions	ISO 5807	4.11.5.?
Documentation of applications	ISO 6592	4.11.5.?
Program flow for sequential files	ISO 6593	4.11.5.?
Data descriptive file for information interchange	ISO 8211	4.11.5.?
Program constructs and conventions	ISO 8631	4.11.5.?
User documentation	ISO 9127	4.11.5.?

Labeling of flexible disks	ISO 7665
Volume and file structure for flexible disks	ISO 9293
Volume and file structure for CD-ROM	ISO 9660
Data descriptive file for information interchange	ISO 8211

The above-mentioned standards might be more suitably called out in Richard Scott's section 4.5.

Software Documentation

There are several standards dealing with documentation to assist with the task of software development, and therefore potentially facilitating programmer and designer portability, as well as user documentation.

Documentation symbols and conventions for data, program and system flowcharts, program network charts, and system resources charts	ISO 5807
Guidelines for the documentation of computer-based application systems	ISO 6592
Program flow for processing sequential files in terms of record groups	ISO 6593
Program constructs and conventions for their representation	ISO 8631
User documentation and cover information for consumer software packages	ISO 9127

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

E.1.5.1.2 Regional Standards

ECMA has approved ECMA-149 as the standard for the Portable Common Tool Environment (PCTE).

E.1.5.1.3 National Standards

To Be Provided

E.1.5.2 Emerging Standards

This subclause describes the activities currently in progress to further standardize this area.

E.1.5.2.1 International Standards

To Be Provided

E.1.5.2.2 Regional Standards

To Be Provided

CASE Data Interchange Format (CDIF)

The CDIF Technical Committee is developing a data interchange format to serve as an industry standard for exchanging information between Computer-Aided Software Engineering (CASE) tools. CDIF is an EIA-endorsed initiative. It assumes that two or more tools may interface asynchronously with each other and will transfer information from one to another via ‘CDIF files.’’ The types of information that may be contained in these files is defined by the CDIF Conceptual Models.

Portable Common Tool Environment (PCTE)

ECMA TC33 has responsibility for the development and maintenance of PCTE. The committee formed a Task Group in 1988 to develop a Reference Model which would assist the standardization process. Such a model has been developed totally independent of PCTE, and is described in ECMA Technical Report 55. The model provides a way to describe, compare, and contrast CASE environment frameworks.

E.1.5.2.3 National Standards

To Be Provided

E.1.5.2.4 National Standards

To Be Provided

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

E.1.5.3 Gaps in Available Standards

E.1.5.3.1 Public Specifications

To Be Provided

E.1.5.3.2 Unsatisfied Service Requirements

To Be Provided

E.1.6 OSE Cross-Category Services

Not applicable.

E.1.7 Related Standards

To Be Provided

E.1.8 Open Issues

— Relationship between methodology and formats

[PCTE and CAIS-A have been moved here largely because it is not clear what to do with them. They are not adequately accommodated by this model. They are both hybrids of operating system and database management system capabilities that seem to belong either everywhere or nowhere. They could both well be used in conjunction with a P1003.1 implementation, but they could also be implemented on other base operating systems, or implementations could even expand their capabilities to provide full operating systems. P1003.0 must decide what to do with them.]

PCTE

An effort by the European Computer Manufacturers Association (ECMA) has resulted in the definition by Technical Committee 33 of the Basis for the Portable Common Tools Environment (PCTE). This is now an ECMA standard and is referred to as Standard ECMA-149.

CAIS-A

MIL-STD-1838A (CAIS-A) was developed by the US Department of Defense to provide a common foundation for Ada Programming Support Environments. Similar in nature to PCTE (see above), it too covers many of the system services covered by 4.2.4. In addition, it provides data management services such as those discussed in 4.4 and data interchange services (specifically, a Common External Form) similar to those discussed in 4.5.

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

Alphabetic Topical Index

A

Abbreviations ... 12
ABS ... 247
Accounting ... 188
ACID ... 101, 103, 107
ACL ... 119, 180
ACSE ... 80
Ada ... 39, 42
Administration ... 149
AD1 ... 76
AEP ... 7, 194
AES ... 55, 58, 158, 247
AFNOR : Association Francaise de Normalization ... 229
AFNOR ... 229
Alphabetic Topical Index ... 263
ANSI : American National Standards Institute ... 230
ANSI X3.122 ... 96-97
ANSI X3.133 ... 90
ANSI X3.135 ... 110
ANSI X3.138 ... 89, 91
ANSI X3.168 ... 89-90, 110
ANSI /ISO ... 136
ANSI ... 42-43, 90-91, 96-97, 109, 123-124, 129, 136, 139-143, 198, 207, 209, 225, 230, 232-233, 235-237, 239, 242, 245, 249
API Service Requirements ... 169
API
definition of ... 12
APL ... 39, 42
APL ... 39, 41-42, 109, 138, 240
Application Environment Profile (AEP)
definition of ... 7
Application Platform Decomposition III — Redirection ... 31
Application Platform Decomposition II — Layering ... 31
Application Platform Implementation — Subdivision ... 30
Application Platform Implementation Considerations ... 28
application platform
definition of ... 7
Application Program Interface (API) Elements ... 115
Application Program Interface (API) ... 17, 22
application program interface (API)
definition of ... 7
Application Program Interface Services ... 47, 69, 86, 95, 105, 116, 135, 146, 153
Application Program Interface ... 20
Application Program Services ... 38
Application Programming Interface Services ... 179
Application Software Elements ... 114
application software entities ... 22
application software
definition of ... 7
Application to Application Service ... 71
Application to System Services ... 69
application
definition of ... 6
Approved POSIX Standardized Profiles ... 203
APP ... 237-238
APT ... 240
ASCE ... 67, 76
ASCII ... 163
ASE ... 80
ASN.1 ... 76, 81, 92, 96, 99, 176
AT&T ... 58, 158, 250-251

B

background ... 2, 6, 102, 155, 166, 220
base standard
definition of ... 7
Base Standards Working Groups ... 217
Basic Terminology ... 194
Basic Window Services ... 116
BASIC ... 40, 42-43
BASIC ... 39-43, 109, 138, 225
Basis for This Guidance ... 196

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