-- --

IEEE P1003.0 Draft 14 – November 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All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change.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

4.9 Character-Based User Interface Services

Responsibility: Martial Van Neste

4.9.1 Overview and Rationale

This clause describes the system services that are related to character-based terminals. It describes both the application program interfaces to character-based terminals and also the look and feel of the interaction between the user and the user interface equipment.

Despite the attention paid to graphical window interfaces, the vast majority of applications are written with a character based user interface. In fact, character-based devices are best suited for applications where the constraints of cost, speed, and the clutter of a pointing device on the desk are a major concern.

It should be noted also that there are character-based window applications that may not have all the flexibility and ease of use of their graphic counterparts, but represent an alternative allowing the utilization of the large installed base of character terminals and still improve the ease of use.

This clause is one portion of the User Interface API and EEI as described in Section 3.

4.9.2 Scope

The scope of this clause is limited to the services and standards required to support character (non-bitmapped) terminals. The services described here do not preclude the use of block-mode terminals, even though most applications built on POSIX -compliant platforms historically have used character-stream terminals.

4.9.3 Reference Model

This subclause identifies the entities and interfaces specific to the character-based terminal services of an OSE.

As illustrated in Figure 4-19, the components of character-based interfaces are broken into two groups: those specifications that impact the application programming interface and those that impact the external user interface.

This reference model is consistent with and expands on the reference model in Section 3.

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

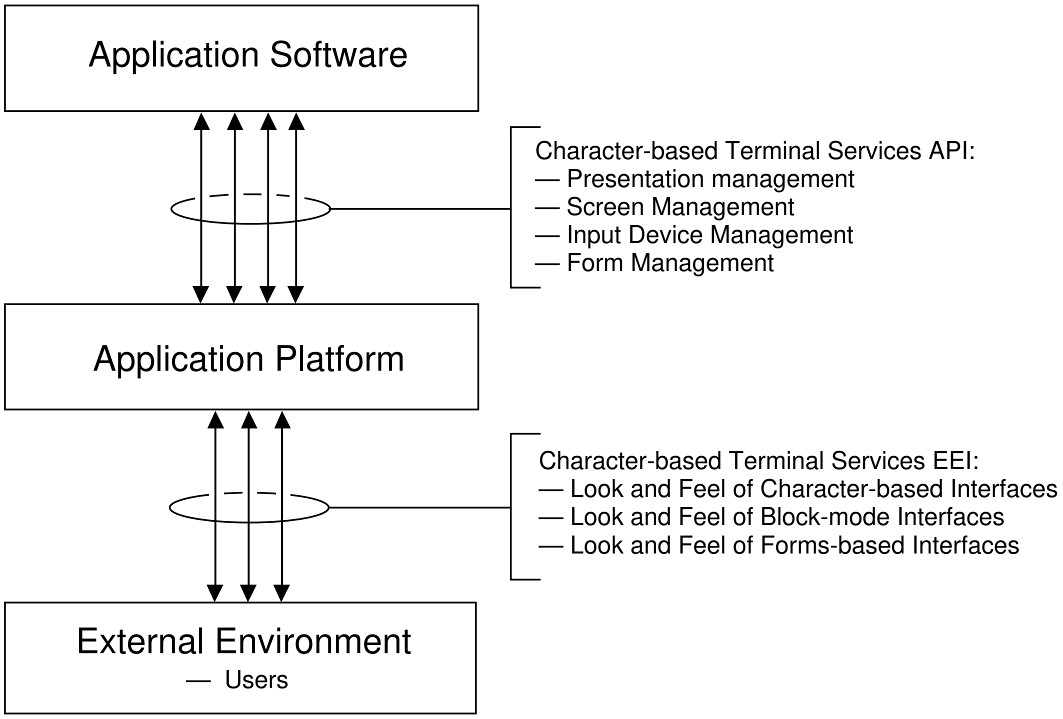

Figure 4-19 – Character-based Terminal Reference Model

4.9.4 Service Requirements

The fundamental service requirements for character-based terminals are to allow applications to be written that make use of the features of a wide variety of terminals using a single terminal-independent interface. The look and feel of user interactions should be consistent between applications to make moving between applications as simple as possible.

4.9.4.1 Application Program Interface Services

Application services include those made available to the application developer to separate the application function from the user interface functions as much as possible.

These standard services support requirements for application portability and terminal independence.

Presentation Management

Functions available for Presentation Management are:

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

- Placement of text on the screen using a consistent reference
- Positioning of the cursor for further output on the scree or for user input
- Control of attributes of displayed text such as highlighting, underscoring, and coloring, if available
- Clearing or refreshing the screen
- Getting the current cursor position

Screen Management

Functions available for Screen Management are:

- Control of the number and the width of the lines displayed
- Use of a protected status line
- Protection from writing or clearing in defined portions of the screen
- Auto-wrapping in defined portions of the screen

Input Device Management

Functions available for Input Device Management are:

- Configuration of the function keys, if available
- Keyboard locking
- Changing key mappings

Form Management

Functions available for Form Management are:

- Definition of a form with different output and input text fields
- Definition of the attribute input fields, such as text or different numeric formats
- Generic and customizable error handling procedures for incorrect input

4.9.4.2 External Environment Interface Services

The look and feel of user interactions with applications should be standardized to make moving between applications as simple as possible. The areas that require standardization are:

- Style of selecting commands
- Accessing online help
- Performing common functions such as page forward and page backwards.

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

— Selecting or moving between fields in a forms-based environment E

These interactions will differ slightly between different types of terminals because of limitations of the terminals.

4.9.4.3 Related Service Requirements

To be provided.

4.9.5 Standards, Specifications, and Gaps

4.9.5.1 Current Standards

None.

4.9.5.2 Emerging Standards

FIMS

ANSI CODASYL. A working draft is available for Forms Interface Management System (FIMS), which covers the interface between a programming language and any form-filling application on a computer or terminal screen.

This specification addresses some of the services requirements for a forms-based user interface.

4.9.5.3 Gaps in Available Standards

4.9.5.3.1 Public Specifications

E

Curses

Curses is a set of subroutines that provide a terminal-independent interface to applications. Many different types of character-based terminals are supported. Curses lacks complete support for flexible user input.

This specification satisfies some of the service requirements for character mode terminals. A recent specification for Curses can be found in volume 3 of X/Open's XPG 3. E

4.9.6 OSE Cross-Category Services

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

4.9.6.1 Security

To Be Provided.

4.9.6.2 Administration

It is important to allow the system management personnel to configure the system to designate where each terminal is connected. Also needed is the ability to add support for new terminals without affecting the application interface.

4.9.6.3 Configuration Management

The system could include a descriptive database of a current set of supported terminals, so that terminal-independent services can do the mapping for the different functions.

4.9.7 Related Standards

None.

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

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

4.10 User Command Interface Services

Responsibility: Wendy Rauch

4.10.1 Rationale and Overview

Although system-level services are necessary for application portability and interoperability, they are insufficient for many users' system needs. To maximize portability, users also require the commands, command interpreter (shell), compilers, editors, and other utilities that have been traditionally associated with many operating systems. These command interface services facilitate a successful port and help users to manage and maintain applications and to solve problems on an ad hoc basis. The standardization of these utilities allows users and programmers to move from platform to platform without having to relearn the command interface for each application platform.

4.10.2 Scope

This clause describes how a user interacts with an application platform by executing general purpose commands. This command interface is also available to applications so that applications also can execute commands. A standardized command interface provides a consistent, interactive environment across platforms for users and programmers.

Commands that are outside the scope of this clause are:

— System administration and installation commands

— Text formatting programs

— Database commands

— Networking and communications commands

— Graphical user interfaces

Networking commands and graphical user interfaces are described in other clauses of this guide.

4.10.3 Reference Model

The use of the command interface services presented in this clause is consistent with the reference model in Section 3. The POSIX OSE reference model for the command interface also is consistent with typical implementations for user command languages in traditional UNIX-based systems.

As Figure 4-20 shows, the command interface is available both to users (through the External Environment Interface) and to applications (through the Application Programming Interface). Any operating system implementation can reside underneath the APIs and EEIs.

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