-- --

IEEE P1003.0 Draft 13 – September 1991

4.8 Character-Based User Interface Services

Responsibility: Charles Severance

4.8.1 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.

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

4.8.2 Scope

The scope of this clause is limited to the services and standards required to support character (non-bitmapped) terminals.

4.8.3 Reference Model

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

As illustrated in Figure 4-18, 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.

4.8.4 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 similar between applications to make moving between applications as simple as possible.

4.8.4.1 Application Program Interface Services

The service requirements at the Application Program interface are as follows:

— Ability to support a wide variety of character-based terminals using a single terminal-independent API.

— Ability to support a wide variety of block mode terminals using a single terminal-independent API.

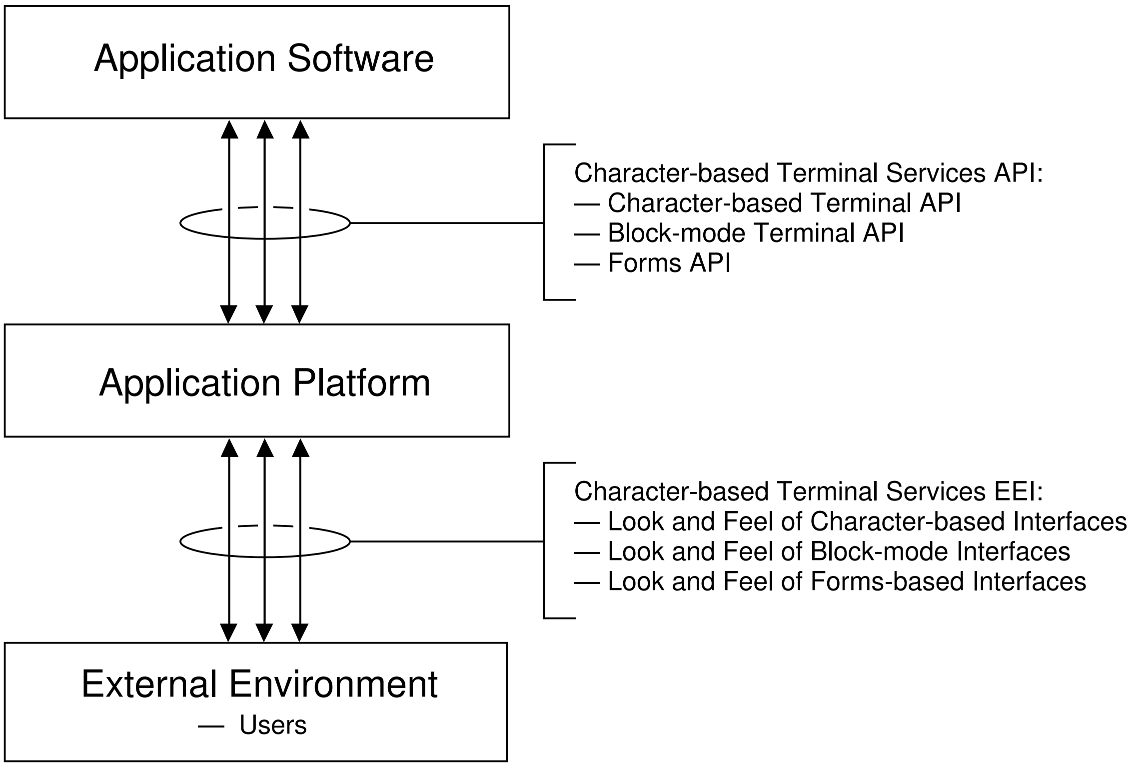

Figure 4-18 – Character-based Terminal Reference Model

— Ability to support all types of terminals using a forms-based API.

These interfaces should be complete enough to allow a wide range of applications to be implemented using the interfaces.

4.8.4.2 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:

— Placement of text on the screen

— Style of selecting commands

— Ways of using online help

— Ways to do common functions such as page forward and page backwards.

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

4.8.4.3 Related Service Requirements

To be provided.

4.8.5 Standards, Specifications, and Gaps

4.8.5.1 Current Standards in the POSIX OSE

None.

4.8.5.2 Emerging Standards in the POSIX OSE

FIMS

ANSI CODASYL. Working draft 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.

CRS : Parts of POSIX. 2 might belong here.

4.8.5.3 Gaps in Available Standards

4.8.5.3.1 Standards/Specifications Outside the POSIX OSE

4.8.5.3.1.1 Government/Legal Standards

None.

4.8.5.3.1.2 De Facto Standards

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 standard satisfies some of the service requirements for character mode terminals.

SAA

SAA is an interface specification to a large class of block mode terminals. This standard satisfies some of the service requirements for controlling block mode terminals.

4.8.6 OSE Cross-Category Services

D

4.8.6.1 Security

4.8.6.2 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.

4.8.6.3 Configuration Management

4.8.6.4 Fault Management

4.8.6.5 Performance

4.8.6.6 Accounting

4.8.6.7 Training

4.8.6.8 Systems Integration Interface Requirements

None.

4.8.7 Related Standards

None.

4.8.8 Open Issues

None.

4.9 User Command Interface Services

Responsibility: Wendy Rauch

HLJ : Wendy has provided a full clause replacement for Draft 13; it is not further D
diff marked. D

4.9.1 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.

4.9.2 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 the POSIX Open Systems Environment Guide.

4.9.3 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 4-19 shows, the command interface is available both to users (through the External Environment Interface) and to applications (through the Application