Contents

Chapter 1

The Common Applications Environment

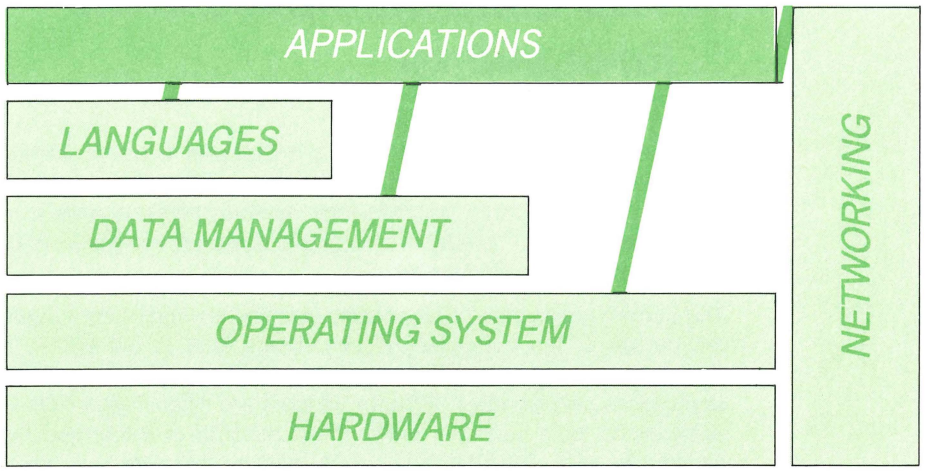
The formation of the X/OPEN Group represents a major initiative by European suppliers of computer systems to create a free and open market, offering Independent Software Vendors (ISVs) as wide a market as possible for their products and giving users an increased return on investment in application software.

The current dominance of proprietary machine environments is restricting the growth of the computer industry. Users tend to get locked into a particular proprietary system by the investment they have made in the applications. Independent Software Vendors are discouraged from writing applications for a particular environment because of the limited markets caused by this fragmentation. This means that there is very little generally available software for each type of system, thus increasing the size of investment needed by each user. All this in turn limits the sales potential of machines from the computer suppliers.

The objective shared by the members of the X/OPEN Group is to establish a Common Applications Environment to the mutual advantage of users, Independent Software Vendors and computer suppliers. Applications written to operate in this environment will be portable at the source code level to a wide range of machines, thereby releasing the user from dependence on a single supplier, reducing the necessary investment in applications, considerably increasing the market for independent software and opening up the market for systems suppliers.

The existence of these "Open Systems" allows users to mix and match systems from different suppliers, and to move applications between machines to meet changing requirements as business grows, thereby giving protection of investment in applications software into the future.

The great increase in the potential market encourages the Independent Software Vendors to produce a wealth of general applications packages, and the availability of this further reduces the investment needed by the users. The whole situation is thus mutually reinforcing.



The foundations of the Common Applications Environment are the interfaces of the UNIX System V operating system, as defined in the AT&T "System V Interface Definition", and the C language.

To define a complete environment for portable applications, it is also necessary to satisfy the requirements for data management, integration of applications, data communications, distributed systems, the use of high level languages and the many other aspects involved in providing a comprehensive applications interface. The X/OPEN Group intends, therefore, to publish progressively definitions covering these areas. This first issue of the Portability Guide gives the definitions for record access to indexed files (ISAM) and for the COBOL and FORTRAN languages.

The systems of the X/OPEN Group members that support interfaces derived from UNIX operating systems will do this according to the X/OPEN definitions and will support the full Common Applications Environment.

A specific Common Applications Environment feature may not, however, be present if it is not relevant in the market area in which a particular system is sold. For example, a system sold only in a scientific context might not support COBOL. Conversely, a particular system may support features over and above those of the Common Applications Environment, some of which may partially overlap. An example of this could be that an alternative dialect of COBOL is supported in addition to that of the Common Applications Environment.

The X/OPEN Group is concerned with standards selection and adoption. It is not a standards creation body. The general policy is to use International Standards, where they exist, and to adopt "de facto" standards in other cases.

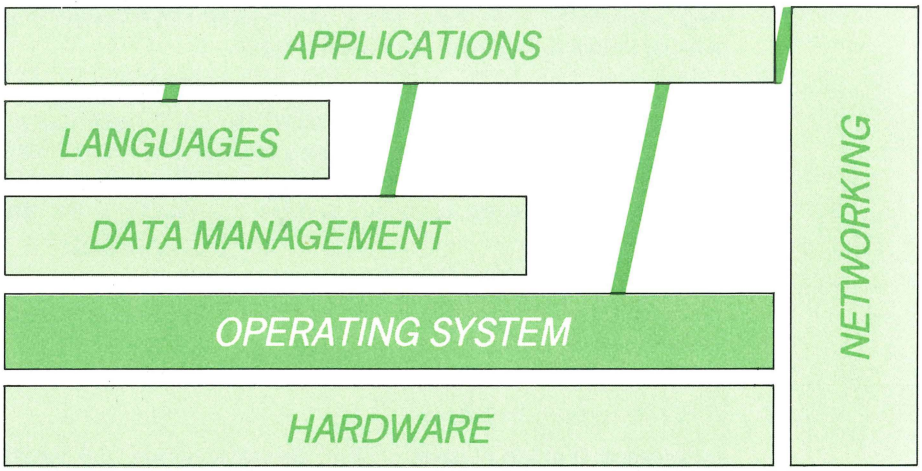
It is important that the defined elements of the Common Applications Environment be readily achievable on member systems, and have wide acceptance. For this reason, the definitions, in general, fall within the capabilities of at least one currently available popular product.

In this guide, certain aspects of the Common Applications Environment are defined with reference to the interfaces offered by specific products. This does not mean that member systems will necessarily contain these products, but that the defined interfaces will be supported. Indeed the method of support for an interface on a particular system may change with time.

The chapter entitled "Future Directions" gives an indication of the areas in which the X/OPEN Group is currently working towards extensions to the Common Applications Environment.

Chapter 2

System V



2.1 INTRODUCTION

This chapter addresses the operating system interfaces, the foundation of the Common Applications Environment. The X/OPEN System V Specification is derived from a series of standards activities, culminating in the production of the AT&T System V Interface Definition. The evolving standard is briefly addressed, and the relationship between the X/OPEN System V Specification and the System V Interface Definition and other standards is explained.

2.2 THE EVOLVING STANDARD

2.2.1 Origins

The UNIX operating system was developed by Ritchie and Thompson at Bell Laboratories in the early 1970s. The current AT&T System V version may be traced back directly to that first system.

For many years, it remained basically an academic product. More recently, computer suppliers have adopted the UNIX system as a multi-tasking, multi-user and portable operating environment. They have based their systems on one of several releases, variants or look-alikes. Of these, the most widely used are Version 7, System III, the Berkeley system and XENIX.

Although these systems have much in common, the degree of compatibility at the application interface level is insufficient to permit the development of totally portable applications.

2.2.2 The /usr/group Standard

/usr/group, a group of users of UNIX derivatives in the USA, established a committee with the objective of proposing a set of standards for application level interfaces. After publishing its standard, together with a reviewer's guide, the group decided to seek IEEE status for the standard. In late 1984, the /usr/group standards committee closed its activities in its own name and its members were encouraged to become involved in the IEEE group, known as P1003.

2.2.3 The AT&T System V Interface Definition

The "System V Interface Definition" (SVID), published by AT&T in the spring of 1985, represents a major standards initiative. AT&T were prominent in the activities of /usr/group and the influence of this standard can clearly be seen in the SVID. The stated purpose of the SVID is to define common interfaces for all System V implementations.

The structure of the SVID differs from that of any previous AT&T UNIX system documentation and from the /usr/group standard. System calls and libraries are redistributed into new sections, referred to as *System Services*, *Other Library Routines* and *Extensions*.

The definition groups interfaces into a mandatory *base* plus a series of *extensions*. The *base* interfaces must be present in any implementations of System V. If any interface from an *extension* is supported, it must adhere to the definition.

2.3 THE X/OPEN SYSTEM V SPECIFICATION

The X/OPEN System V Specification (XVS) is based upon the AT&T System V Interface Definition, but also taking into account the standard published by /usr/group and the capabilities of UNIX System V Release 2.0.

The X/OPEN Group has extended the SVID in a number of areas:

- the Group has taken certain changes into its specification, which the SVID denotes as future directions
- the use of symbolic names to replace numeric constants, introduced by AT&T in their SVID, has been extended
- clarification of existing wording has been introduced in a limited number of places to "tighten" the specification
- the opportunity has been taken to correct a small number of clerical errors in the SVID
- the Group has included definitions of a number of further UNIX System V Release 2.0 functions which are in widespread use by application developers.

Differences between the XVS and the SVID are clearly annotated. The whole of the SVID *base* definition is included as mandatory with the exception of *termio*, which is not mandatory in the XVS because of known implementation difficulties, and the maths group, which is not mandatory for systems sold into markets where it is not relevant.

The policy of X/OPEN is to maintain compatibility between the AT&T SVID and the X/OPEN Portability Guide such that a system which conforms to the X/OPEN guidelines will pass the AT&T System V verification test. X/OPEN will collaborate closely with AT&T to ensure that future changes and extensions to the AT&T SVID are reflected in the X/OPEN specification.

The XVS incorporates all the interfaces within the SVID *kernel extension set* (K_EXT) with the exception of those relating to:

- shared memory
- semaphores
- message passing

Shared memory functions are machine specific and the Group believes that a more generalised approach to the whole subject of Inter Process Communication is required.

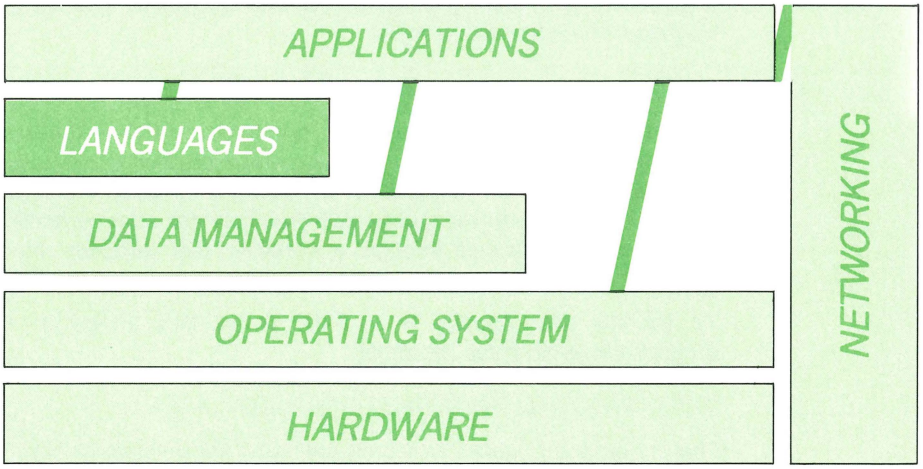
In the XVS, interfaces defined as *optional* will be available on most but not all X/OPEN systems; use of them could restrict portability. Any *optional* interface supported on an X/OPEN system will conform to the X/OPEN specification.

The XVS defines interfaces in terms of their interface syntax and run-time behaviour, without constraining the method of their implementation. The names ''system calls'' and ''subroutines'' are retained purely for compatibility with other documentation.

PART II of this guide contains a full definition of the interfaces included within the X/OPEN System V Specification.

Chapter 3

C Language



3.1 INTRODUCTION

This chapter addresses the C language and guidelines for portability when writing C code.

Currently the ANSI committee, X3J11, is working towards a standard for the C programming language. The X/OPEN Group is represented on that committee by member companies and intends to adopt the standard, once it has been established as a practical reality.

Meanwhile, the X/OPEN definition will be based upon that of the C language given in Chapter 2 of the " System V Programming Guide", Release 2.0, published by AT&T. This is included in PART III of this guide.

3.2 C LANGUAGE PORTABILITY GUIDELINES

Whilst the C language provides the basis for applications portability, it is easy to write statements, using valid C constructs, that are machine specific. Care has to be taken when writing programs that are intended to be portable across a range of systems. Chapter 3 of PART III includes advice towards ensuring portability.

3.3 INTERNATIONALISATION ISSUES

Internationalisation is a key topic in X/OPEN plans and will be treated in depth in a future issue of this guide. Meanwhile, Chapter 3 of PART III includes a section giving basic advice on the structuring of C programs to minimise development costs, whilst facilitating their use in different language/character set contexts.

3.4 THE ANSI X3J11 DRAFT STANDARD

The ANSI X3J11 standard has been published in a draft form but will almost certainly change before it is approved. However, it is already clear that the standard will impose certain restrictions such that programs written to the current C language definition may not work correctly, if the source is later passed through a compiler that supports the ANSI standard.

To address this, Chapter 3 of PART III includes advice on writing programs to avoid these problems.

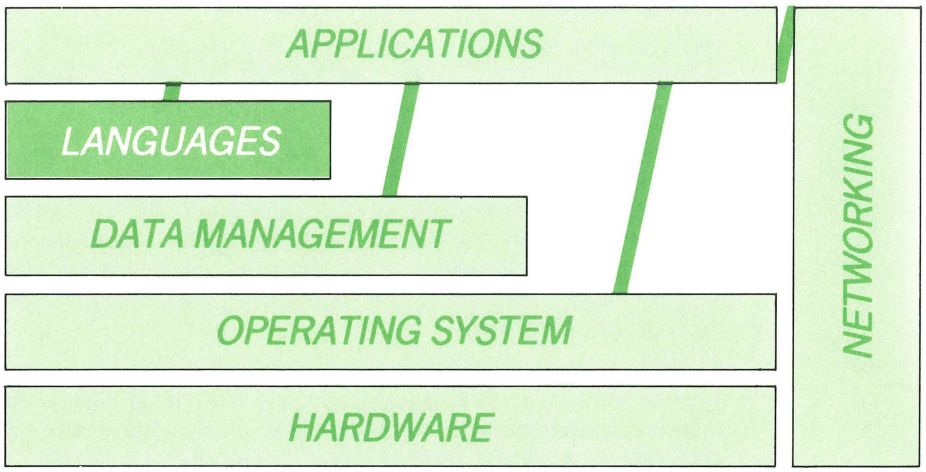
3.5 THE C PROGRAM PORTABILITY CHECKER (*lint*)

The *lint* program checks C source programs for violation of most of the portability rules. It also gives a more stringent enforcement of the type rules of C than is provided by most C compilers. A further option detects a number of wasteful or error-prone constructions which nevertheless are syntactically correct.

Use of the *lint* program is recommended: it is fully described in Chapter 4 of PART III.

Chapter 4

Other Programming Languages



4.1 INTRODUCTION

This chapter addresses the requirements for programming languages other than C on X/OPEN systems. It covers the inclusion of the principal high level languages in the Common Applications Environment.

To date, The X/OPEN Group has established definitions for COBOL and FORTRAN.

4.2 COBOL

The X/OPEN COBOL definition identifies a common set of language facilities that will be supported by COBOL compilers on all member systems. Applications written to this definition will be portable to any X/OPEN system.

The accepted standard for COBOL is that defined in the American National Standards document "ANSI X3.23 - 1974", to which most current COBOL compilers substantially conform.

The ANSI standard is incomplete in the area of facilities for interaction with the on-line user. To overcome this deficiency, most COBOL compilers provide extensions to the *ACCEPT* and *DISPLAY* verbs, but they do this in incompatible ways. Since the majority of applications now include interactive operation, it is necessary for a standard form of *ACCEPT* and *DISPLAY* to be defined in the X/OPEN Common Applications Environment.

In order to have an X/OPEN definition that is achievable on member systems within a short timescale, and one that would have immediate widespread acceptance, it has been based on the definition of COBOL embodied in a popular product: Micro Focus LEVEL II COBOL, which itself conforms to the ANSI standard.

The Micro Focus LEVEL II COBOL language specification includes a number of other extensions beyond the ANSI standard, in addition to those to *ACCEPT* and *DISPLAY*. None of these are currently included in the X/OPEN definition. The X/OPEN definition also applies a few restrictions to the ANSI-based parts of the LEVEL II definition.

Whilst the X/OPEN COBOL definition is based on the specification of a particular product, the means of implementation across the systems of the X/OPEN members may vary. Any particular system may support extensions beyond the facilities identified, but their use is likely to impede portability.

The X/OPEN COBOL definition is given in detail in PART V and its relationships to both the ANSI 74 standard and LEVEL II COBOL are identified.

The definition is given in terms of the command syntax. It is derived from the Syntax Summary from Appendix F of the LEVEL II COBOL Reference Manual, with the specification of the elements that have been included in the X/OPEN definition clearly marked. The semantics of the language are defined as being those of the ANSI 74 standard as documented in the Micro Focus LEVEL II COBOL language specification.

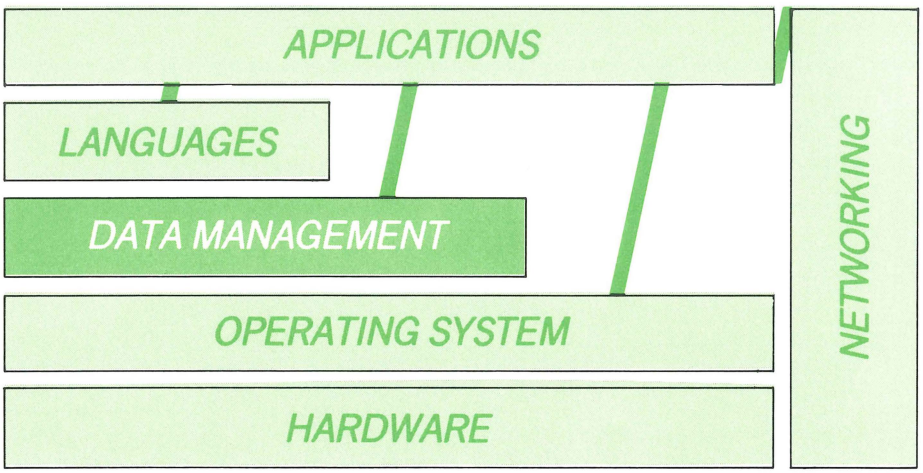
4.3 FORTRAN

The X/OPEN definition for FORTRAN is the formal definition given in the American National Standards document "FORTRAN 77, ANSI X3.9 - 1978". This has had wide-scale acceptance throughout the world and there are many certified compilers available.

The majority of FORTRAN compilers, while adhering to the basic FORTRAN 77 standard, also offer extensions beyond that standard. There is little compatibility in these extensions between compilers and they do not form part of the X/OPEN definition. Developers are warned that use of these extensions will affect the portability of FORTRAN programs.

Chapter 5

Data Management



5.1 INTRODUCTION

The input/output facilities supported by System V consist only of byte-stream read and write operations on files. No facilities are provided for operating on files as sets of records. This leads to application writers having to make their own arrangements for record handling, resulting in both a multiplication of effort and a proliferation of non-standard methods.

Data Management is a key element in the integration of applications. Applications written in a variety of languages must be able to work on the same basic data in the same form, and data must be passed easily and efficiently between applications.

As a first step towards addressing these issues, the X/OPEN Group defines an interface for the creation, management and manipulation of indexed files, generally known as the Indexed Sequential Access Method (ISAM). The availability of this interface on X/OPEN systems will not only provide application portability, but will ease and encourage integration.

The X/OPEN Group also recognises the importance of relational database facilities. As a further step towards applications portability and integration, the X/OPEN Group intends to support the emerging SQL standard for access to relational data bases. For applications portability, this refers to the use of SQL facilities within programs (embedded SQL).

5.2 INDEXED SEQUENTIAL ACCESS METHOD (ISAM)

The X/OPEN definition for ISAM, which is contained in PART IV of this guide, is a major subset of the specification of the C-ISAM product, Version 2.10, published by Relational Database Systems Inc.

The full specification of C-ISAM contains implementation details specific to that product, in addition to the definition of the interface available to applications. Only the applications interface forms part of the X/OPEN definition; implementation details specific to C-ISAM have been omitted. Indeed, there may be alternative implementations available on particular member systems.

Chapter 6

Source Code Transfer Between Machines

6.1 INTRODUCTION

One of the major problems inhibiting the porting of applications between UNIX system derivatives is that of incompatible media standards and the physical problems of transferring source code in machine readable form.

The X/OPEN Group takes this problem seriously and has agreed common standards for the transfer of source code. Detailed standards are defined in PART VI.

Standards are defined for transfer of 5 ¼″ floppy discs and ½″ magnetic tape between machines. Because of the different nature of X/OPEN systems, ranging from single user work stations to large mainframes, it is not possible to define formats which are portable across the whole range. Defining standards for both floppy discs and ½″ magnetic tape gives the highest practical coverage of systems.

Current differences in the physical recording formats between cartridge tape devices prevents the definition of a standard for this popular medium.

6.2 FLOPPY DISC STANDARD

As exchange media, the X/OPEN group defines standards for 40 and 80 track floppy discs. It is intended that the prime format should be 80 track, with 40 track retained for compatibility with personal computers. X/OPEN systems equipped only with 80 track disc drives will offer the facility to read 40 track floppy discs by skipping alternate tracks.

6.3 MAGNETIC TAPE

The X/OPEN standard for magnetic tape covers ½″ magnetic tape, with a number of different recording formats and densities. The prime format is 9 track Phase Encoded at 1600 bits per inch.

6.4 UTILITIES

Part VI includes the definition of two alternative utilities for the archiving of files to the transfer medium and their subsequent retrieval, *tar* and *cpio*.

In addition, guidelines are given on the use of direct machine to machine connection and the *uucp* utility, as a means of transferring files between X/OPEN systems.

Chapter 7

Future Directions

7.1 INTRODUCTION

The X/OPEN Group recognises that the work of defining the Common Applications Environment is incomplete, and has planned a comprehensive programme of work to enhance and extend the guidelines given in this guide.

This chapter includes statements of future direction in several important areas.

7.2 INTERNATIONALISATION

X/OPEN members market systems in many countries. Our customers and users speak many different languages and conform to different cultural conventions and business practises. In many cases, a strong requirement also exists to cope with these variations on the same system, and even within the same document. An example is within the administration of the European Economic Community.

Currently, UNIX System V and most systems derived from it are based on the 7-bit ASCII character code-set. There are no facilities for dealing with other code-sets, nor for supporting different national languages and conventions.

The requirement for effective mixed language working brings with it the need for character sets larger than can be accommodated in a 7-bit character, as does the requirement to support the more complex languages. At the same time, there is a trade-off between the ability to handle larger character sets, and the inefficiency that can result in the amount of storage required to hold the data. For most requirements not involving a complex language, a single byte (8-bit) system provides the correct balance. For the major Eastern languages (such as Chinese and Japanese), a two byte (16-bit) system is necessary even to support a single language.

To satisfy these primary requirements enhancements must be made to provide full byte transparency to applications, allowing complete flexibility in the code-sets employed. Additionally, the general utilities that handle ''characters'' in a transparent way should know whether single byte or double byte characters are being handled, and modify their behaviour accordingly.

The X/OPEN Group regards internationalisation as being a subject of considerable importance and is working on both the basic code-set issues and the many other issues involved in the production of systems appropriate to different countries and cultures. An awareness of current work in this area by the Japanese computer industry is having the effect of harmonising the methods of dealing with single byte and double byte requirements. The objective is that fundamental facilities for international working will be available at the earliest opportunity.

Although the primary requirement is to provide a system that is flexible, allowing the use of different code-sets, decisions have to be made on the actual codes to be used. For this, it is X/OPEN policy to conform to agreed national or international standards. For example, (draft) ISO standard DIS 8859/1, Latin alphabet no. 1 is likely to be adopted to cover the major Western European language requirements within a single code-set.

7.3 NETWORKING AND COMMUNICATION

7.3.1 Introduction

The general target for computer data communications is interworking between systems of different types from different suppliers. Future X/OPEN definitions for such open systems interworking can be expected to embrace the ISO OSI standard.

Two services specific to systems supporting the System V Interface have been identified. These are ''Generalised Inter Process Communication'' (IPC) and ''Distributed File System''.

Many current commercial applications are supported via interactive transaction processing systems. X/OPEN definitions in this area can be expected to follow the work being done by ECMA TC32.

A number of proprietary networks for the interconnection of personal computers already exist. X/OPEN may define interworking with leading de-facto standard PC networks as these emerge in the future.

7.3.2 Open Systems Interconnection

For interworking to be possible, systems must have common methods of describing both tasks and data, and must be capable of functioning in a defined manner. To describe interconnection, an architectural model is required. The International Organisation for Standardisation, ISO, has developed the ''Reference Model for Open Systems Interconnection'' (IS7498). This is often referred to as the ''ISO OSI model'' or the ''ISO 7-layer model''. This model sets a framework into which protocol and service standards can be set.

The ISO OSI target is to have a complete set of protocol and service definitions which comply with the 7-layer model and which are internationally agreed and published as ISO standards.

A complete set of ISO OSI standards does not yet exist. Where standards are available, they contain options. For practical systems to be built, it is necessary to have a very clear definition of standards to be adopted and the options to be used.

To provide a clear statement of the standards to be used, a specialist working group has been formed by the 12 major European vendors who propose the technical objectives for ''ESPRIT'', the ''European Strategic Programme for Research into Information Technology''. This is called the ''Standards Promotion and Application Group, SPAG''.

Where ISO standards do not yet exist, interim standards from one of the national or international standards bodies are adopted by SPAG. Such standards are expected to form the basis of future ISO standards. SPAG is not itself a standard making body. Its recommendations will reflect evolving standards.

X/OPEN member companies are committed to the ISO OSI target and the adoption of ISO standards. The group will monitor SPAG recommendations.

X/OPEN intends to define application interfaces for access to OSI services to ensure the portability of applications and library routines.

7.3.3 Generalised Inter-Process Communication, IPC

UNIX operating systems provide limited IPC capabilities in the form of "pipes" and "fifos". Kernel extensions within the AT&T System V Interface Definition provide some further IPC mechanisms for the passing of messages between processes in the same memory address space. These extensions have been omitted from the X/OPEN definition because it is believed that a much more generalised IPC capability is needed.

Generalised IPC is the ability to exchange data between processes, either in the same physical machine or in different machines connected via some local communications medium such as a bus or local area network. In this context, the following characteristics apply:

- The IPC interchange is identical in all cases. The data exchange is bi-directional and fully transparent with no protocol rules imposed by the IPC mechanism.
- Access to IPC endpoints should follow the normal UNIX filestore naming conventions and protection mechanisms.
- Message passing is asynchronous (i.e. non-blocking read/write; the ability to wait, subject to timeout, for data availability from one of a number of sources; the option to request that an asynchronous event be generated to indicate data availability).

The X/OPEN Group intends to define an IPC service for applications. It is expected that this will be based upon the STREAMS mechanisms proposed by AT&T.

7.3.4 Distributed File System

There is an increasing requirement to be able to access data contained within UNIX File Systems on machines connected together by a local area network from any system on that network. The totality of data accessible in this way can be regarded as a ''distributed file system''.

The X/OPEN Group therefore intends to define facilities for the support of distributed file systems, the characteristics of which are likely to be:

- Access to the distributed file system should be via the standard System V input/output system calls, and should be identical for local and remote files.
- No changes should be necessary to existing applications. Binary copies of existing applications should be able to access a distributed file system, subject to the requirement that the data within a file is in a compatible format.
- Naming of remote items should follow the same syntax as for local items and no new naming conventions should be required.
- Tools should be provided to facilitate the administration of distributed file systems. These should include facilities to enable items within a distributed environment to be located.
- The file locking mechanism will be extended to apply across the network.

7.3.5 Distributed Transaction Processing

Many commercial applications are supported via interactive transaction processing systems. Currently, there are no international standards in this area, but work by the ECMA TC32 group is expected to lead to an ECMA standard for a connection-oriented protocol for distributed transaction processing (Reference: ECMA/TC 32/85/89) by the end of 1985.

The proposed standard will be compatible with existing ISO OSI standards, and also with the ''basic conversation protocol boundary'' of the IBM SNA LU6.2 (Logical Unit 6, type 2), which is a de-facto standard for present-day distributed transaction processing.

It may be expected that interfaces to transaction processing implementations on X/OPEN systems will conform to the ECMA standard and to any eventual corresponding ISO standard.

7.4 RELATIONAL DATA BASE

The X/OPEN Group recognises the increasing importance of relational data base systems and their associated tool kits to the software developer.

The emergence of SQL as a standard for the applications interface to relational data base systems is being closely followed by the Group, as is the emergence of an increasing number of products which claim SQL compatibility.

As the draft American National Standards document, ANSI X3H2 - 1984, becomes more stable, the X/OPEN Group intends to define SQL interfaces based upon that standard.

7.5 USER INTERFACE

In common with /usr/group, the X/OPEN Group have found it impossible to define a totally portable set of user interface routines. The *curses* library, available in a number of different forms from different suppliers, allows application programs to perform terminal-handling functions in a way that is independent of the type of terminal actually in use.

The different versions of the *curses* library are not fully compatible and a future edition of this Guide will define a portable subset of *curses*. In the interim, part II contains a section giving some guidance regarding the portability of routines within the *curses* libraries.

The X/OPEN Group will be working on enhanced user interface definitions based on advanced human factors principles. These will provide the capability of handling windows, menus, icons and graphics which can be accessed by a keyboard or other input device, such as a mouse. Applications written to these defined interfaces will have a uniform and easily used human interface.

7.6 GRAPHICS

Future X/OPEN definitions will track current industry efforts to define standards for graphics functions. One area under active consideration is the Graphics Kernel Subsystem (GKS).

7.7 X/OPEN SYSTEM V SPECIFICATION

The guide will be extended and enhanced in future editions to include:

1. A common definition of commands.
2. The definition of a common environment for development, including the portability of ''make scripts''.

7.8 ADDITIONAL LANGUAGES

Definitions for other languages (such as PASCAL) will be published in future editions of the guide.

7.9 SOURCE CODE TRANSFER

In recognition of existing practical difficulties in the transfer of source code between systems, the X/OPEN Group will continue to search for common media transfers and will work to develop easier-to-use routines to control the transfer of files via a direct machine to machine link.