-- --

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All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change.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

the 1988 CCITT X.400 series of recommendations. The X.400 API and Object Management API are separate documents.

The purpose of the X.400 API is to provide standard interfaces supporting the development of applications that are users of the message transfer system, and gateways that incorporate or use X.400 mail functionality; this includes gateways between X.400 mail networks and proprietary mail systems.

The purpose of the companion OSI Object Management API is to provide a standard interface supporting the manipulation of complex arguments and parameters used by the X.400 and Directory Services APIs. The scope of the OSI Object Management API is to define an ASN.1 Object Management API for use in conjunction with, but otherwise independent of, the X.400 and Directory Services APIs that are currently being standardized.

4.3.5.3 Gaps in Available Standards

This subclause describes the standards that are cited to satisfy identified service requirements that are not satisfied by any existing or emerging standard.

Interoperable Networking Standards

This set of protocol standards is the traditional TCP/IP suite of standards, which are currently widely available on many computer platforms and operating systems.

This group of specifications includes:

TCP/IP	MIL-STD-1777, MIL-STD-1778
TELNET	MIL-STD-1782
FTP	MIL-STD-1780
SMTP	MIL-STD-1781

While these protocols are not expected to be standardized at any higher level than the MIL-STD level shown, it will be necessary for open systems to interoperate with these standards in a backwards-compatibility mode for some time.

Low Cost Wide Area Networking

The UUCP (UNIX-to-UNIX Copy Protocol) services and commands, for electronic mail and file copying, which are traditionally included in UNIX and UNIX-like systems are not addressed by any standards effort. Among other reasons, UUCP is not currently being addressed because of the inability of the POSIX groups to decide whether the UUCP services and commands should be standardized in the POSIX.2 Group (since UUCP is a traditional UNIX service with traditional command interfaces) or in the networking groups (since UUCP is an electronic mail and file copying facility that works on networks).

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

4.3.6 OSE Cross-Category Services

These EEI Services allow remote systems to be managed and monitored. Network management services include the ability to:

- — Get network status information
- — Get network configuration information
- — Test network functionality
- — Make network configuration changes

The security services allow the system management to control access to system resources and system information. Security services include:

- — Protect the system from intruders
- — Provide selective access to sensitive system resources
- — Manage the network security

See also 5.3.

4.3.7 Related Standards

ISO 8587, Distributed Transaction Services; see 4.6.

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

4.4 Database Services

Responsibility: Sandra Swearingen

4.4.1 Overview and Rationale

This subclause describes an overview of an architectural framework for discussing database management. It also describes the services provided to application programs and users, and it describes standards, current and emerging, that standardize those database services.

Database management is an important component of the POSIX Open System Environment; in a large class of application programs, especially those used in business, database access through a database management system plays a key role. For portability and interoperability, an application using a database must be isolated from the hardware and software retrieval methods as much as possible. Otherwise the application must have the data manipulation capability coded in its own programs. This might be done if performance is a key issue and the data is very unique. The cost is portability and interoperability.

E

4.4.2 Scope

Included within the component of database management are various structured ‘‘data models,’’ including indexed files and network, relational, semantic, and object-oriented databases. Specifically excluded from consideration are services for accessing data that is not part of a database. This subclause discusses database management services from both the application program and user points of view.

Database services provided to application programs by this component, for example, include the ability to create, alter, or drop tables, records, and fields and the ability to insert, select, and update data included in the structures above.

Included within this component are also utility capabilities such as database administration services.

E

4.4.3 Reference Model

4.4.3.1 Reference Model

In this subclause, the conventional view of Database Management is related to the POSIX reference model described earlier.

The application programmer’s view of the database model is introduced first. Quite simply, an application program, through a *Database API*, requests database

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

services. For convenience in the following discussion, the agent responsible for providing those services is called the *Database Manager* . The database manager is responsible for providing the application access to the *Database* . See Figure 4-10.

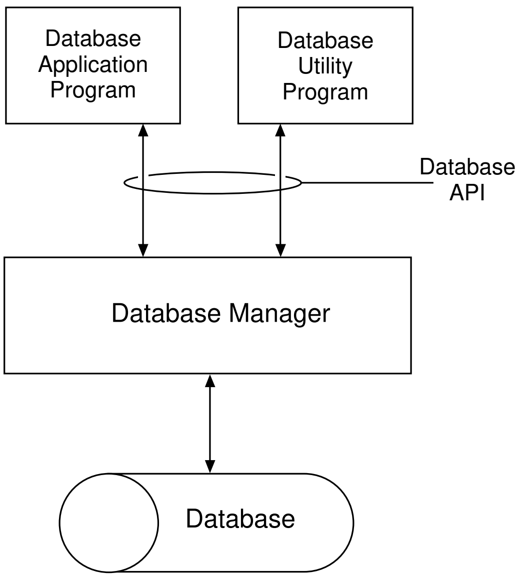

Figure 4-10 – The Traditional Database Model

This figure also demonstrates the concept of a *Database Utility Program* : one or more special application programs, usually provided by a database vendor, that perform utility services on the database. Such utilities might reorganize the database, recover the database after a system failure, etc.

The traditional database model can be incorporated into the POSIX reference model, as in Figure 4-11. This depiction of the model shows that the database manager is really just part of the overall POSIX Open System Environment and is available to the application through the POSIX OSE API.

The model depicted in Figure 4-11. is sufficient to describe an application developer's view of the POSIX OSE API in general, and the database API specifically. The four lines labeled ''Database API''represent the Database Applications Program Interface services, which are discussed in 4.4.4.1.

4.4.3.2 Implementation Aspects

Some real world considerations of the POSIX Reference Model were discussed in 3.6. One of the real-world approaches described is ''layering.'' Note that in the marketplace, Database Managers are often independently purchasable components that are effectively implemented as layers. Another consideration is Database Manager portability. It is not a requirement that a a database manager

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

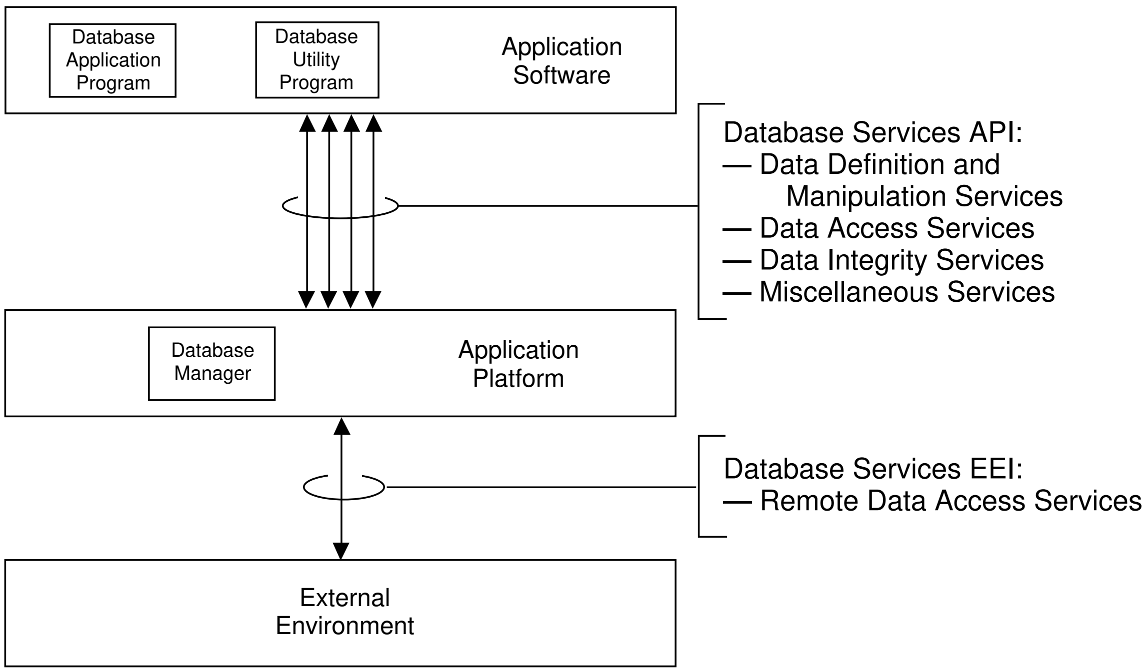

Figure 4-11 – POSIX Database Reference Model

sit on top of a POSIX OSE API, but there is very important value to the user in terms of portability if the database manager implementation does indeed sit on a POSIX API. This means that the database manager itself is portable. It should be noted that there will probably be implementations available of database managers that do not, in fact, sit on top of a POSIX API (or sit only partially on top of a POSIX API), that nonetheless provide the user the same database API. Such an implementation, using both POSIX API services and non-POSIX API services was described as ‘‘expansion’’ (see 3.6.1). Note that even though the model is drawn with only a single database manager, that does not imply that there may only be a single database manager available to an application. In fact, the coexistence of several database managers on the same system is consistent with this model, as is the ability of a single application to access two or more different database managers. The following extensions to the above model are specifically accommodated:

— There may be more than one database API. For example, there may be an ‘‘SQL’’API and an ‘‘ISAM’’API.

— There may be more than one database manager implementation for each different API. (For example, by two competing database vendors.)

— Applications may access more than one database manager.

This document has not described how a database manager is implemented in a POSIX Open System Environment, nor is it within the scope of this document to

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

do so. It should be noted, though, that this model is very general and does not constrain the implementor. This model supports a number of varying real world implementation techniques, including:

— Application and database manager linked into a single process.

— Database manager consisting of more than one process.

— Database manager consisting of a client part linked into the application process and a server part running as a process on the same or another system.

4.4.4 Service Requirements

The Database Manager described in the previous subclause provides services to the Application Program via the Database API, and the Database Utility Programs provide other services (e.g., to human users such as a ‘‘Database Administrator’’). This subclause describes the service requirements of all service users of the system. It is intended to be a complete list of service requirements rather than examples. Database Services are the specialized data services required to create, access, and manage databases located on a processor node. Users of these services include end users and those charged with the ongoing management of the information processing and database infrastructure.

4.4.4.1 Application Program Interface Services

This subclause describes the major categories of database services available at the POSIX Application Program Interface (API). These services include:

— Data Definition and Manipulation Services

— Data Access Services

— Data Integrity Services

— Miscellaneous Services

The following paragraphs clarify that these services should be provided for a large class of objects, access methods, and types of database systems.

Types of Data Objects
: Ability to perform the above operations on a variety of types of data objects, such as text, graphics, image, documents, and voice.

Types of Access Methods
: Ability to perform the above operations using a variety of access methods, such as indexed sequential access, nonindexed sequential access, and direct access.

Types of Database Management Systems
: Ability to perform the above operations on a variety of types of file and database management systems, and database management systems, such as relational, network, semantic, and object oriented

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

databases, and heterogeneous combinations of these database management systems.

4.4.4.1.1 Data Definition and Manipulation Services

These services relate to the ability of application programs to define and manipulate data. These services are:

— Data definition — ability to create, alter, or drop tables, views, records, fields, and/or data

— Data Manipulation — ability to insert, select, update, and delete tables, views, records, fields, and data

4.4.4.1.2 Data Access Services

These services relate to the ability of application programs to interrogate databases. These services are:

— Data Query Facilities — ability to specify search conditions, consisting of a combination of select lists, predicates, and comparison operators

— Data Transparency — ability to transparently access data regardless of the location of that data.

— Remote Data Access — ability to access and update remote data

4.4.4.1.3 Data Integrity Services

These services relate to the ability of database management systems to protect the databases from hardware and software malfunctions.

— Locking — ability to specify locking of data to some degree of granularity

— Consistency — ability to specify and execute check and referential constraints that help ensure data correctness

— Transaction Control — ability to specify commit and rollback commands and guarantee serializability for database transactions E

— Synchronous Writes (Durability?) — ability to force the writing of data to nonvolatile storage

4.4.4.1.4 Miscellaneous Database Services E

Miscellaneous database services include: E

— Privilege Administration — ability to grant and revoke privileges for accessing and administering data

— Exception Handling — ability to have applications that are interrupted and notified of exception conditions, to receive control of the machine and take action in response to these exception conditions—even if the action is to ‘‘continue’’

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

- Screen Definitions — ability to create screen definitions, and define, display, and/or paint screens to communicate information about databases
- Reporting — ability to create formatted reports.
- Dynamic Facilities — ability to temporarily turn control of a database to the end user for interactive access and manipulation of data, and then return control to the application.
- Data Dictionary Services — ability to get data about the data (i.e., meta-data) stored in the database. This allows users and applications to use the database contents in a much more flexible way. These services allow a user to create, access, and manage this metadata much in the same way as other databases are maintained.

4.4.4.2 External Environment Interface Services

External Environment Interface services are required for distributed database management systems. Also, to enable two or more databases to communicate with each other, a common interchange format is required. See 4.5.

4.4.4.3 Database Resource Management Services

These services are not visible to the application programmer at the Database API. These services are usually provided by Database Utility Programs. These services include:

- Database Administration Services
- Database Recovery Services
- Distributed Database Management Services
- Heterogeneous Environment Support Services

4.4.4.3.1 Database Administration Services

Database administration services refer to the ability for a designated data administrator to structure and configuration manage a database as a whole. The administrator allocates resources and monitors utilization to assure that authorized users receive the proper services. Archive functions, journaling, and logging services are available to the user and administrator on a selective basis.

4.4.4.3.2 Database Recovery Services

Database recovery services refer to the ability to decide that there has been a failure, allow recovery from failure, and permit a slave copy to become a master copy.

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

4.4.4.3.3 Distributed Database Management Services

Distributed database management services support the partitioning and partial replication of the databases.

4.4.4.3.4 Heterogeneous Environment Support Services

Heterogeneous environment support services permit local database systems to be of different types (e.g., inverted list, hierarchical, network, relational) by providing translators between the local database form and a general ‘‘network language.’’

4.4.4.3.5 Flagger

A flagger is software that alerts programmers about any code that does not conform to the standard in question, such as the Structured Query Language standard.

4.4.5 Standards, Specifications, and Gaps

There are currently four database standards, either completed or under development. These are the relational data language SQL, a network data language called NDL, the Information Resource Dictionary System (IRDS) for data dictionary work, and a Remote Data Access (RDA) protocol. Table 4-5 summarizes the service requirements provided by the various standards.

Table 4-5 – Database Standards

Service	Type	Specification	Subclause
Data Definition and Manipulation Services Data Access Services Data Integrity Services	S	SQL: ISO 9075 ANSI X3.168	4.4.5.1
Data Definition and Manipulation Services Data Access Services Data Integrity Services	S	NDL: ISO 8907	4.4.5.1
Miscellaneous Services (Data Security and Integrity, Exception Handling, Screen Definitions, Reporting, Dynamic Facilities, Data Dictionary Services) Database Resource Management Services (Database Administration, Recovery From Failure)	E	IRDS: ISO DP 10027 N2642 (IRDS Framework), ISO DP 8800 N2132 (IRDS Interfaces), ANSI X3.138	4.4.5.2
External Environment Interface Services	E	RDA: ISO/IEC DP 9759	4.4.5.1

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

4.4.5.1 Current Standards

This subclause describes the current accepted standards that apply to this area.

SQL Standard Database Language

ISO 9075 (FIPS 127)

ANSI X3.168

ISO 9075 provides for many of the services described in 4.4.4, including Data Definition, Manipulation, and Integrity. It provides for two levels of compliance: the weaker Level 1 and the more capable Level 2. While ISO 9075 deals with SQL independently of programming language, X3.168 binds, or embeds, SQL within the programming languages COBOL, FORTRAN, Pascal, PL/1, C, and Ada.

Work is currently planned by ANSI and ISO to include ''generalized triggers,'' ''generalized assertions,'' ''recursive expressions,'' ''escape from SQL,'' subtables, and support tools for object oriented and knowledge-based systems.

NDL Standard Database Language

ISO 8907

ANSI X3.133

This standard, developed in 1981-1986 by the ANSI X3H2 Database Committee, specifies a data definition language (DDL) and data manipulation language (DML) for network model databases. This work is an outgrowth of the 1978 CODASYL specifications.

This standard provides for many of the services described in 4.4.4, including Data Definition, Manipulation, Access, and Integrity. The above services apply only to network databases (i.e., not to relational or semantic databases.)

No follow-on NDL activities are being conducted by ISO or ANSI.

4.4.5.2 Emerging Standards

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

Remote Data Access (RDA) Protocol

ISO DP 9579-1 *Generic Remote Database Access* — DP 2

ISO DP 9579-2 *SQL Specialization* — DP 1

This standard, developed by the ECMA Technical Committee on Database Standards, TC22, submitted to ISO in 1985, specifies a protocol that allows remote access and updating, via OSI communications protocols, of relational databases or of database systems that support SQL.

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

This standard provides for the Data Transparency, Remote Data Access, and Support for Heterogeneous Environment requirements described in 4.4. This protocol is aimed at relational databases and other database types that provide access via relational interfaces such as SQL.

Much work is planned on in this area by the ISO committee ISO TC97/SC21/WG3. A specific area of current interest is a generic RDA that uses a nonspecific database language (i.e., not SQL.)

Information Resource Dictionary System (IRDS)

ANSI X3.138	FIPS Pub 156, April 5, 1989
ANSI X3H4/90-28 (draft, 4 Apr 90)	IRDS Export/Import File Format
ISO DP 10027 N2642 (1988)	IRDS Framework
ISO DP 8800 N2132 (1988)	IRDS Services Interfaces

These standards are being developed by the ANSI X3H4 Database Group and the ISO/IEC /JTC 1/SC21 Working Group 3. Both groups are addressing the general area of data dictionaries, but, as described below, the emphases of the activities differ.

The ANSI group primarily addresses the user interface area; that is, how a human user can access the Data Dictionary Services described in 4.4.4.

The ISO groups concentrate more on the service interfaces (APIs) among lower level components and utilities of the database model.

Differences in scope and incompatibilities exist between the model being developed by ISO and the model approved by ANSI. They are independently developing the Services Interface, and an export/import facility.

4.4.5.3 Gaps in Available Standards

There are two significant areas described in 4.4.4 as requirements that are not addressed by standards:

— Methods to access data such as hashing and indexed sequential access to data is not currently standardized. There is no consensus in the standards community as to whether such standardization would be beneficial.

— Standardization of semantic and object oriented models have also not taken place, though groups like the ANSI Database system study group (DBSSG) are currently investigating the feasibility of standardization in these areas.

— I/O Services such as screen generation.

— Management and control of database services. Also include all gaps (all services without standards).

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

4.4.6 OSE Cross-Category Services

4.4.6.1 Security

The ability to specify logical database access control mechanisms is important to database security.

4.4.7 Related Standards

The standards and activities described in this subclause are related to the above and may also be relevant to your activities.

There are several areas closely related to the Database area that are worth considering with respect to standardization.

The first area to consider is the communications and networking area. Interoperability for distributed applications or the use of distributed databases may indicate the use of communications software adhering to networking standards. See 4.3 for further discussion of that area. (Specifically consider the following standards described in that subclause:

ISO/IEC 9804.3	(OSI CCR services)
ISO/IEC 9805.3	(OSI CCR protocol)
ISO 8824	*Information Processing Systems— OSI—Specification of Abstract Syntax Notation One (ASN. 1)*
ISO 8825	*Information Processing Systems— OSI—Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN. 1)*

The second area to consider is transaction processing. That area goes further in addressing the total requirements for distributed applications. See 4.6.

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

4.5 Data Interchange Services

Responsibility: Richard Scott

4.5.1 Overview and Rationale

The Data Interchange/Information Exchange components of the POSIX Open System Environment provide specialized support for the exchange of data between applications or components of applications. Without support for data interchange, problems can arise when attempts are made to move data between different operational environments or between two related applications. More specifically, data interchange problems arise in each of the five following situations:

— Movement of a single application program and its associated data between operational environments,

— Movement of data between cooperating application software within the same operational environment,

— Movement of data between cooperating application software operating in differing operational environments,

— Movement of data between related, but not cooperating, application software within a single operational environment, and across differing operational environments.

From the global view, the data interchange components can provide the means to satisfy the needs in each of these situations. These standards need to define physical formats, data formats, code sets, and data descriptions that are consistent across all implementations of the POSIX Open System Environment.

4.5.2 Scope

The data interchange component of the POSIX Open System Environment includes standard services, protocols, and data formats required to ensure that data can be interchanged between related application software. Physical media formats are beyond the scope of the POSIX Open System Environment.

4.5.3 Reference Model

The Data Interchange Services relate directly to the POSIX Open System Environment reference model that was presented in Figure 3-1. Figure 4-12 shows the components of the reference model that are significant for data interchange. The reference model defines the conceptual relationships required to provide these facilities. It should not be viewed as a description of an implementation. The key entities within the figure are the Application Software, the Application Platform, and the External Environment. To satisfy the data interchange service requirements, the POSIX Open System Environment must permit application software to transfer data to and from the external environment.

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

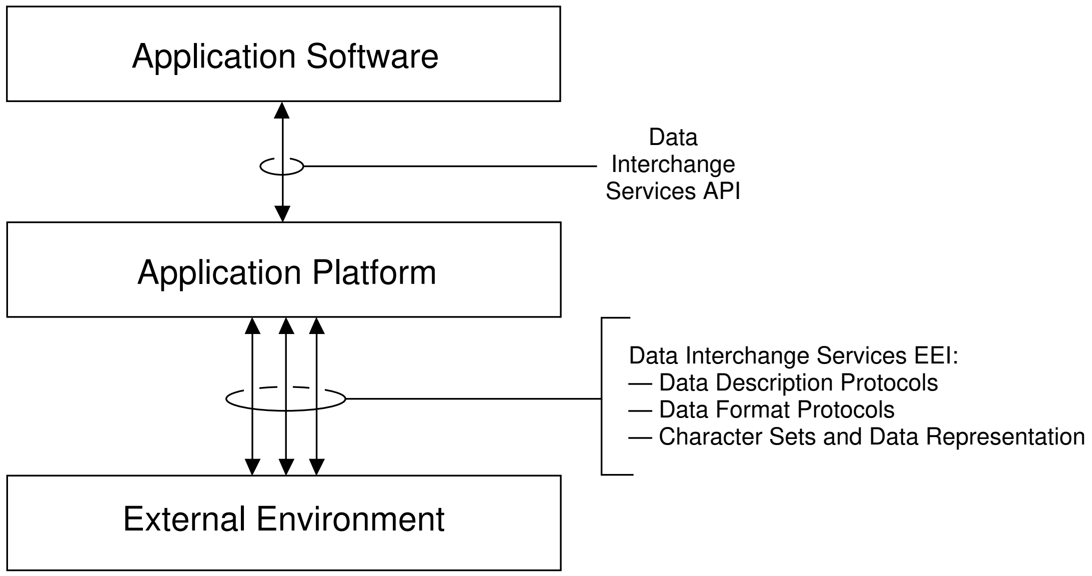

Figure 4-12 – Data Interchange Reference Model

The application software requests this transfer through the Application Program Interface. In response to those requests, the data interchange components of the Application Platform handle conversions to and from standard formats and the transfer of the information across the External Environment Interface (EEI). The EEI , which defines the format specifications required to support data interchange, can be divided into Data Description Protocols and Data Format Protocols. Data Description Protocols provide a means to identify the data that is present. Data Format Protocols provide the storage representation of the actual data.

Today, this model is only partially supported by standards. Physical formats are fairly well standardized. Some work is beginning on data format protocols standards, particularly in the networking area. At this time, no general standards exist to support Data description protocols.

4.5.4 Service Requirements

This subclause details the Data Interchange Services and protocols that are required to support application portability and interoperability. Subclause 4.5.4.1 describes the API service requirements. 4.5.4.2 describes the EEI service (i.e., protocol) requirements.

Data interchange is one of the components of the POSIX Open System Environment that is now just beginning to evolve. At this time, the general requirements for services are understood, but there is little general existing practice that can be pointed to as showing that current service requirements are both necessary and complete. Most existing practice is limited to a specific application domain. As a

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

developing area, data interchange represents gaps in both the definition of service requirements and standards. The data interchange component is, none the less, critical for supporting application portability and interoperability. The data interchange service requirements are currently described to the extent possible at this time in their evolution. More detail will be added in future revisions of this guide.

4.5.4.1 Application Program Interface Services

The API services to support data interchange need to provide the ability to store and retrieve data using the formats and protocols provided at the data interchange EEI.

At this time little work has been directed at defining API-level service requirements for data interchange. Data interchange API services need to provide a means to request that specific data be represented using the EEI services defined below. Progress in this area is similar to the development of the networking area. Initial standards defined protocols and only after those were in use has attention shifted to providing a standard mechanism for requesting those networking services.

4.5.4.2 External Environment Interface Services

This section identifies the EEI services required to support data interchange. These services are all in the form of protocol and format definitions. As shown in Figure 4-12, these protocols include:

— Character Sets and Data Representation

— Data Format Protocols

— Data Description Protocols

These protocols are required to support the exchange of information between application software entities, both within a single application platform and between application platforms.

4.5.4.2.1 Character Sets and Data Representation

The ability to support Character Sets and Data Representation is crucial to providing effective data interchange between application software operating under differing language and cultural conventions. These services add facilities to the POSIX Open System Environment to identify the character set and data representations associated with textual data. A detailed description of the requirements in this area can be found in 5.1.

4.5.4.2.2 Data Format Protocols

The data format protocols need to provide the ability to identify the representation of the data in a manner that is independent of the specific execution environment. The data format protocol layer adds attributes that describe the physical

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

characteristics of the data that must be known to properly retrieve the data value, given the storage formats that are native on the hardware/software environment where the data is used. The complete attribute information required to decipher that data value includes:

— Detailed storage format for the value

— The data value in an environment-neutral format

The data format protocols protect applications from hardware/software differences between environments. Specifically, the protocols ensure that data remains stable when moving between environments where the character set, word size, or byte ordering may differ.

4.5.4.2.3 Data Description Protocols

Data description protocols provide the ability to share data between related application software entities, even if they were not specifically written to cooperate. Building upon the facilities provided by the previous two Data Interchange EEI Services, data description protocols provide a means of associating a name or other identifier with the individual data elements in a standard manner. This permits an application program to correctly identify data that was created by an unrelated application. To date, most standards in this area have limited themselves to specific application areas and no general solution has been provided.

4.5.5 Standards, Specifications, and Gaps

See Table 4-6.

4.5.5.1 Current Standards

Open Document Architecture (ODA)/Open Document Interchange Format (ODIF)

The ODA/ODIF standard (ISO 8613 Parts 1-8) provides a standard for the structures used to represent documents. The ODA model defines a comprehensive description of a documents format. It supports both reproduction of the original document and also editing and formatting of the document.

Part 5 of the ISO ODA standard specifies the interchange format for ODA documents; specifically ODIF. ODIF is an ASN.1 (ISO 8825) based presentation of the ODA document.

Standard Generalized Markup Language (SGML)

SGML (ISO 8879) is a language that allows users to precisely define the structure of a document. The key difference between SGML and ODA/ODIF is that the former provides the flexibility to define custom document types.

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

Table 4-6 – Data Interchange Standards

Service		Type	Specification	Subclause	
Data Description Protocols	ODA/ODIF	S	ISO 8613 Parts 1-8	4.5.5.1	E
	SGML	S	ISO 8879	4.5.5.1	E
	EDIFACT	S	ISO 9735	4.5.5.1	E
	STEP	E	ISO DP 10303	4.5.5.2	E
	EDIFACT	S	ANSI X.12	4.5.5.1	E
	IGES	G	NBSIR 86	4.5.5.3	E
	VHDL VHSIC	G	IEEE P1076	4.5.5.3	E
Data Format Protocols	ODA/ODIF	S	ISO 8613 Parts 1-8	4.5.5.1	E
	SGML	S	ISO 8879	4.5.5.1	E
	CGM	S	ISO 8632	4.5.5.1	E
	CGM	S	ANSI X3.122-1986	4.5.5.1	E
	STEP	E	ISO DP 10303	4.5.5.2	E
	EDIFACT	S	ISO 9735	4.5.5.1	E
	EDIFACT	S	ANSI X.12	4.5.5.1	E
	IGES	G	NBSIR 86-3359	4.5.5.3	E
	VHDL VHSIC	G	IEEE P1076	4.5.5.3	E
	CDIF	G		4.5.5.3	E

Computer Graphics Metafile (CGM)

CGM (ISO 8632, ANSI X3.122-1986) provides a standard means for the storage and exchange of computer graphics. Graphic information is stored in a device- and resolution-independent fashion that can support both the display and the manipulation of the data.

Electronic Data Interchange (EDI)

The EDI standards [ISO 9735 (EDIFACT), ANSI X.12] provide support for the exchange of structured business data. EDI is typically used to transfer business documents such as purchase orders, invoices, promotional announcements, and electronic funds transfer information.

E

4.5.5.2 Emerging Standards

Standard for the Exchange of Product Model Data (STEP)

STEP (ISO DP 10303) is a neutral mechanism capable of completely representing product data throughout the life cycle of a product. The completeness of this representation makes it suitable not only for file exchange, but also as a basis for implementing and sharing databases of archiving.

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

E

4.5.5.3 Gaps in Available Standards

4.5.5.3.1 Public Specifications

Most standards activity in the data interchange area has been isolated to specialized application areas. These activities have attempted to support data interchange by limiting the scope of the effort to a specific type of data. These industry standards include:

E

Initial Graphics Exchange Specification (NBSIR 86-3359)

IGES is used heavily in the exchange of graphical information between applications.

E

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.

E

Hardware Description Language (VHDL VHSIC)

The VHDL standard (IEEE P1076) defines a representation for the exchange of CAD representations of electronic circuits.

4.5.5.3.2 Unsatisfied Service Requirements

None of these standards addresses a general means to handle application data in a manner to ensure portability between environments. The closest attempt is the effort just beginning in POSIX.8 to define a means within the network interface to provide data portability. However, even this effort is not attempting to solve the broader issue of interoperability of multiple applications. The existing standards have all evolved to support the interchange of specific types of data between separate applications. Support for general data portability is not addressed by existing standard, except for ISIS, which does not appear to have caught on.

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

4.5.6 OSE Cross-Category Services

Not applicable.

4.5.7 Related Standards

The following standards are related to the services covered in this clause as they address some of the services described in section 4.6.4 at some level. Each of these related standards are addressed fully as part of another service category.

ASN. 1	ISO 8824	Abstract Syntax Notation (Clause 4.3)
	ISO 8825	ASN. 1 Basic Encoding Rules (Clause 4.3)
MHS	ISO/CCITT X.400-1984	Message Handling System (Clause 4.3)
	ISO/CCITT X.400-1988	Message Handling System (Clause 4.3)

4.5.8 Open Issues

Data interchange support must address hardware/software differences between environments. The key concerns in transporting data that must be addressed will include the character set, word size, and byte ordering for the operating environment along with a accurate identification of the data value.

The data portability standards adopted within POSIX Open System Environment need to define data formats that will enable applications to create data that can be used read and properly interpreted on differing operating environments and by other application software.

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.