-- --

IEEE P1003.0 Draft 14 – November 1991

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.

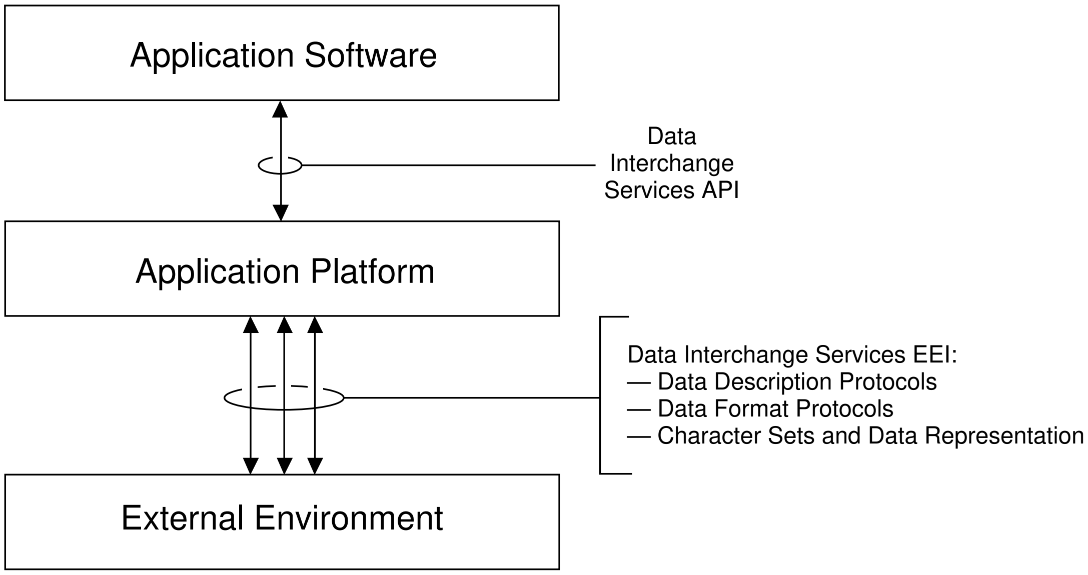

Figure 4-12 – Data Interchange Reference Model

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 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 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.

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

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.

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.

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.

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.

4.6 Transaction Processing Services

Responsibility: Bob Gambrel

4.6.1 Overview and Rationale

The database management clause (see 4.4) described some transaction processing (TP) service requirements (specific to databases). This clause describes the complete set of transaction processing services from the application software point of view. Note that transaction processing services have long been been regarded, variously, to be within the domain of databases or within the domain of operating systems and more recently within the domain of interconnect. These services are more broadly applicable than just both of these areas, and so are treated here as a separate clause.

Transactions (''units of work'') have boundaries (start points and end points) that are determined by the action of the transaction application program. The transaction application program can request to either commit or rollback the work done in the transaction when it identifies the end point. The system will complete a commit operation only if all operations performed during the transaction can complete successfully. Otherwise the system will abort the transaction (rollback the work done by it) and notify the transaction application program of this action.

The following is quoted with a few editorial changes from ISO/IEC DP 10026-1, the ISO Distributed Transaction Processing standard draft:

> A transaction is characterized by four properties: atomicity, consistency, isolation, and durability. These are the *ACID* properties.
>
> Atomicity implies that the operations of a unit of work are either all performed, or none of them are performed.
>
> Consistency implies that the operations of a unit of work, if performed at all, are performed accurately, correctly, and with validity, with respect to application semantics.
>
> Isolation implies that the partial results of a unit of work are not accessible, except by operations which are part of the unit of work. Isolation also implies that units of work which share bound data are serializable.
>
> Durability implies that all the effects of a completed unit of work are not altered by any sort of failure.

4.6.2 Scope

This clause deals with the transaction processing services needed for a large number of styles of transaction processing including the following:

— Transactional access to a single database manager on a single machine