Final Committee Draft ISO/IEC FCD	
Date: **2000-11-02**	Reference number: ISO/JTC 1/SC 32 **N 0583**
Supersedes document SC 32 N 0018	

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

ISO/IEC JTC 1/SC 32 Data Management and Interchange Secretariat: USA (ANSI)	Circulated to P- and O-members, and to technical committees and organizations in liaison for voting (P-members only) by: **2001-03-05** Please return all votes and comments in electronic form directly to the SC 32 Secretariat by the due date indicated.

ISO/IEC FCD 13238-1

Title: : Information Technology - Data Management Export/Import Facilities - Part 1: Standardization Framework

Project: 1.32.58.01.01.00

Introductory note: The attached document is hereby submitted for a four-month letter ballot to the National Bodies of ISO/IEC JTC 1/SC 32. The ballot starts 2000-11-02; ITTF must have the document two weeks before the balloting can start.

Medium: E

No. of pages:

Address Reply to: Douglas Mann, Secretariat, ISO/IEC JTC 1/SC 32, Pacific Northwest National Laboratory, 901 D Street, SW., Suite 900, Washington, DC, 20024-2115, United States of America

Telephone: +1 703 575 2114; Facsimile; +1 703 681 9180; E-mail: MannD@battelle.org

ISO/IEC JTC 1/SC 32

Date: 2000-10-13

ISO/IEC FCD 13238-1

ISO/IEC JTC 1/SC 32/RG 2

Secretariat: SCC

Information Technology — Data Management Export/Import — Part 1: Standardization Framework

Technologies de l'information — Échange de données — Partie 1 : Cadre pour la normalisation

Document type: International Standard
Document subtype:
Document stage: (40) Enquiry
Document language: E

D:\BerJ_Normes\5i-SC32 DM\P68-ei\12158-01_EI_Fram\odaa-doc\12158-01_32N0583_FCD_EI_Framework_V10.doc STD Version 1.0

Contents

Table of Illustrations

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 13238 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 13238-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 32, *Data Management and Interchange*.

ISO/IEC 13238 consists of the following parts, under the general title *Information Technology — Data Management Export/Import*:

— *Part 1: Standardization Framework*

— *Part 2: SQL Export/Import*

— *Part 3: IRDS Export/Import*

The Standardization Framework deals with the generic aspects of data and metadata interchange. It represents the view of the interchange of data that is required for standards management purposes. The other parts define transfer formats for specific domains.

In this standard *export/import* is understood in the broad sense as concerning

— the transfer data from one Data Management System (DMS) to another ***without*** establishing an association between the two data management systems for the duration of the transfer, and

— the sharing of a clear understanding between the exporting system and the importing system of the structure and the meaning of the data to be transferred.

The content of this standard has been developed over a number of years, initially JTC1 SC21, and now in JTC1 SC32, the international committee responsible for standardization in the area of Data Management and Interchange. The Framework described in this international standard is one of several ways of viewing the subject matter; other ways are possible. The publication of this part of ISO/IEC 13238 is intended to facilitate trial use and feedback, to enable the Framework to be refined as Data Management standardization itself evolves.

— Clauses 0 to 5 are prescribed ISO/IEC clauses.

— Clause 6: Concepts and facilities

  This describes the basic concepts of communication, and the process and data architecture associated with export/import.

— Clause 7 Standardized components of an export/import facility

This identifies the components of an export/import facility that are candidate for standardization.

Reference material has been confined to the annexes of this international standard. Annex A contains the initial rationale and requirements for this standard.

Information Technology — Data Management Export/Import — Part 1: Standardization Framework

1 Scope

1.1 Field of application

Export and import of bulk data are essential activities wherever bulk data is shared between information systems, within or between real systems. Whether and to what extent a uniform and standardized approach can be used in export/import has become an important issue in the broader context of open systems and information technology standardization.

The basic concept of export/import does not depend on the application, the type of bulk data to be exported/imported, or the medium used for transfer. Furthermore, export and import do not imply any predetermined relationship between the exporting and importing systems but assume that semantic integrity of the data is fully preserved in the process.

Aspects to be examined include:

— Export/Import as a generic concept applicable to a variety of data management environments and data transfer contexts.

— Aspects of export/import (e.g., Services, file formats, application specific aspects, etc.). Identification of aspects amenable to standardization.

— Existing standards that might be relevant to export/import and why.

— Whether new standards of some kind or other are needed, and if so how to go about defining them and getting them accepted.

The objective of export/import standards is to encourage the provision of general-purpose export/import facilities.

An export/import facility that is developed without reference to any standard will likely be usable only in conjunction with the specific tools for which it was developed. As the number of systems to be interfaced increases, the number of interfaces will increase factorially if each pair of tools has to be interfaced directly.

The number of interfaces between n products is given by: n * (n-l)/2.

With the provision of a standard interface format, each new tool need provide only one interface, to that standard format, to be able to interface to all other tools that have interfaces to the same standard format. Thus, the number of interfaces required in this case will be equal to n.

However, it is still necessary that the export and import systems have compatible data models for data to be transferable. That is the constructs and rules used to structure data in one system can be mapped to those used in the second system

This standard defines the format of a generic Export/Import file, and the functions that must be supported in conjunction with the export or import of such a file. Specialization of this standard will define specific formats of the Export/Import data file for particular data modelling facilities. e.g., IRDS and SQL.

1.2 Audience and purpose

This framework is written for standards makers in the data management domain and in domains that interface with or overlap with data management. The audience will be primarily ISO and IEC committees, but national bodies, professional bodies, trade associations and others involved in standards making may find value in the framework.

This international standard introduces an overall framework of Export/Import as a generic concept applicable to a variety of data management environments and data transfer contexts. It also contains a clear and concise statement of requirements and issues.

This international standard deals with the identification of aspects amenable to standardization (e.g., services, file formats, application specific aspects, etc.), and with the question whether new standards of some kind or other are needed.

The need for some kind of framework for standardization (often called a reference model) to provide an overview of the available and required standards in a particular domain has long been recognised within ISO and IEC. Such frameworks have proved to be useful tools when partitioning the work to be done into projects and also in developing a suitable administrative structure in terms of working groups and rapporteur groups.

2 Conformance

Conformance requirements are not stated in this part of ISO/IEC 12158, but in subsequent parts.

Other standards and implementations are said to be consistent with this standard when they exhibit an architecture compatible with the one proposed in this standard.

3 Normative reference(s)

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revisions, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

ISO 8601:1997, *Data elements and interchange formats - Representation of dates and times.*

ISO/IEC 8824-1:1995, *Information Technology - Abstract Syntax Notation One (ASN.1) - Part 1: Specification of basic notation.*

ISO/IEC 9075-1:1999*, Information technology - Database languages - SQL - Part 1: Framework.*

ISO/IEC 9075-2:1999, *Information technology - Database languages - SQL - Part 2: Foundation*.

ISO/IEC 10027:1990*, Information Technology - IRDS Framework.*

ISO/IEC 10032:1995, *Information Technology - Reference Model of Data Management.*

ISO/IEC 15475-1:2000, *Information Technology - CDIF General Rules for Transfer and Encoding*

4 Terms and definitions

For the purposes of this international standard, the following definitions apply. Unless otherwise noted, the definitions are specific to this international standard.

4.1 From other standards

4.1.1 ISO/IEC 10032

This part of ISO/IEC 13238 makes use of the following terms defined in ISO/IEC 10032, Reference Model for Data Management.

Data management system;
Data management environment.

4.1.2 ISO/IEC 8824-1

This part of ISO/IEC 13238 makes use of the following terms defined in ISO/IEC 8824-1 ASN.1:

Encoding;
Type;
Value.

4.1.3 ISO/IEC 9075

The following terms defined in ISO/IEC 9075 Database Language SQL are used either in this part or other parts of ISO/IEC 12158.

Attribute;
Column;
Data type;
Identifier;
Instance (of a value);
Row;
Subtype;
Supertype;
Table.

4.1.4 ISO/IEC 10027

This part of ISO/IEC 13238 makes use of the following terms from ISO/IEC 10027 IRDS Framework:

Application level;
IRD level.

4.1.5 ISO/IEC 15475-1

This part of ISO/IEC 13238 makes use of the following terms (without the CDIF prefix) from ISO/IEC 15475-1 CDIF General Rules for Transfer and Encoding:

Character Set;
Clear Text;
Encoding;
Syntax.

4.2 For this standard

For the purpose of this part of ISO/IEC 13238 the following definitions apply:

Agent (of a process)
The entity, outside of the scope of the export/import facility, that triggers and expects results from the export/import processes.

Transfer file
A file containing data to be interchanged. It is made up of a header, and a number of components. A component contains either data, or data definition data.

DMEI transfer file
A transfer file consistent with this standard.

CDIF transfer file
A transfer file conforming to ISO/IEC 15475.

DMEI/SQL transfer file
A transfer file conforming to ISO/IEC 12158-2.

This file containing data that conveys the definition and content of an SQL database, or a subset of such database. It is made of a header, and at least one of two components: transfer file SQL-schema definition and transfer file SQL-data .

DMEI/IRDS transfer file
A transfer file conforming to ISO/IEC 12158-3.

This file containing data that conveys the definition and content of an IRD or a subset of an IRD. It is made up of a header, and at least one of multiple components. There are two types of components: transfer file IRD definition component, and transfer file IRD content component.

DMEI Transfer file header
The first part of a transfer file. The header contains data that uniquely defines the transfer file. It also contains details about the source of the transfer file (the exporter) and source-defined parameters.

Export process
The process of generating a transfer file from a source environment.

Exporter
The agent of the export process.

Import process
The process of incorporating the content of a transfer file into a target environment.

Importer
The agent of the import process.

Clear text file encoding
A class of techniques for representing data based on a human readable representation (some specific character repertoire and a corresponding encoding).

5 Symbols (and abbreviated terms)

5.1 Diagramming notations

A system model is used to identify and position the main E/I functions or processes, with the associated input and output products or interfaces. The modelling is done at an implementation-independent level.

Figure 1 - System Modelling Notation

The following components are used:

— Processes (functions, procedures, activities, tasks) are shown as boxes with rounded corners.

— Square-cornered boxes are used for products (data).

— Associations are noted by arrows as illustrated in Figure 1. Processes "receive" and "produce" products.

Dotted lines are used for the symbols to show external components.

The following rules apply:

— Two products or two processes cannot be associated by the produces/receives associations.

— When introducing boundaries, thereby creating aggregate components, all components at the boundary must be of the same type. The aggregate component is of that type.

5.2 Abbreviations

The following abbreviations are used in this international standard:

API Application Programming Interface

DBMS Database Management System

DMS Data Management System

E/I Export/Import

RDA Remote Database Access

RPC Remote Procedure Call

6 Concept and facilities

The purpose of an Export/Import facility is to transfer data and/or data definitions from one Data Management System (DMS) to another ***without*** necessarily requiring:

a) direct communication between the two systems, and

b) an association between the two DMSs for the duration of the transfer.

Although most of the discussion is done using the notion of transfer of data through space, the same concepts apply to transfer of data through time. That is exporting data from a system, and importing the same data to the same system later. Backup and recovery are a manifestation of that type of export/import.

Data transfer in which such an association ***is*** established is considered a function of Remote Database Access, and/or other communication standards, rather than Export/Import, and is therefore not addressed by this framework.

The Export/Import Framework defines generic facilities based on the Reference Model of Data Management and may be used by any system.

This framework describes a common approach to E/I, and makes provisions for parts that are generic, and parts that make provisions for specialization to support the specific requirements of other standards, such as IRDS and SQL.

6.1 Data Communication/Interchange in a client/server environment

6.1.1 Communication mechanisms

There are different levels of automated connection between importer and exporter. The minimum level is where human beings, who know what to do with the Export/Import file, have established the liaison. At the highest level, the E/I file contains requests and responses following a protocol known to both entities, and identified on the file.

There is a dependency between the level of association and the media used, for instance from shared memory to removable storage media.

6.1.2 Communication model

Export/import is defined as a special case of the more generic communication model illustrated in Figure 2. This approach builds on facilities that have already been defined in other standards, for instance in ISO/IEC 9579 RDA.

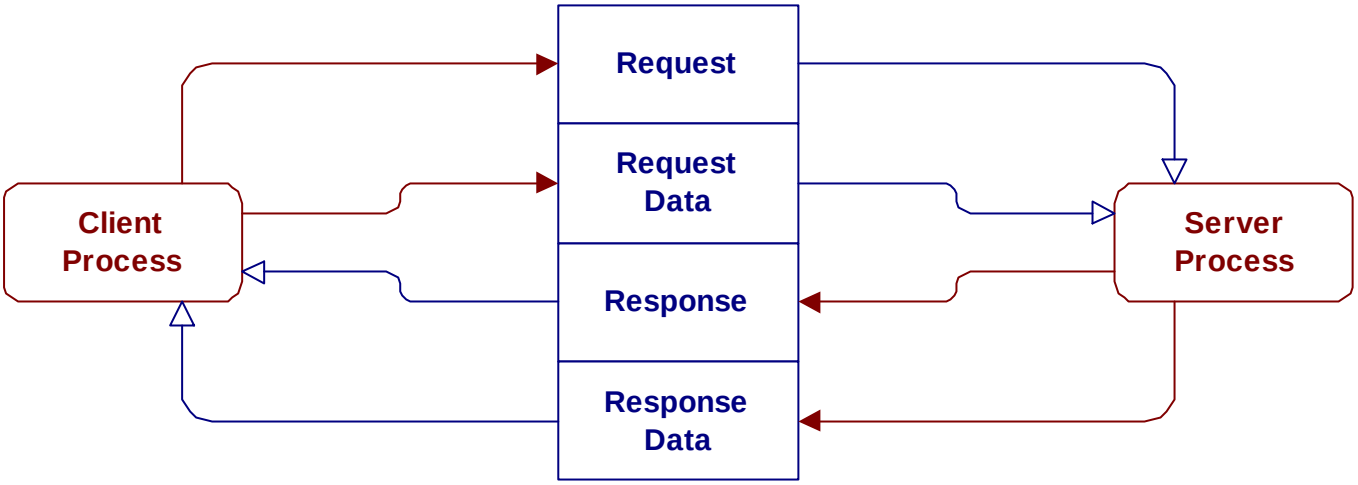

Figure 2 - General Client Server Communication Model

Service requests and responses are exchanged together with data, in blocks of data, and whether there is one or many blocks, or whether the set of data blocks, or data stream, is on a diskette, in memory, or a network message, is just a variant of the basic model. Not all the four messages (request, request data, response, response data), involved in the complete exchange need be automated. For instance, for the simplest case of producing a file on diskette (response data), the three other messages (request, request data, response) are human-readable, and not machine-readable. They have to be interchanged by a human being to complete the exchange.

This communication may involve different types of association between client and server, and different media used for communication. Some applications of this model are shown in figure 3.

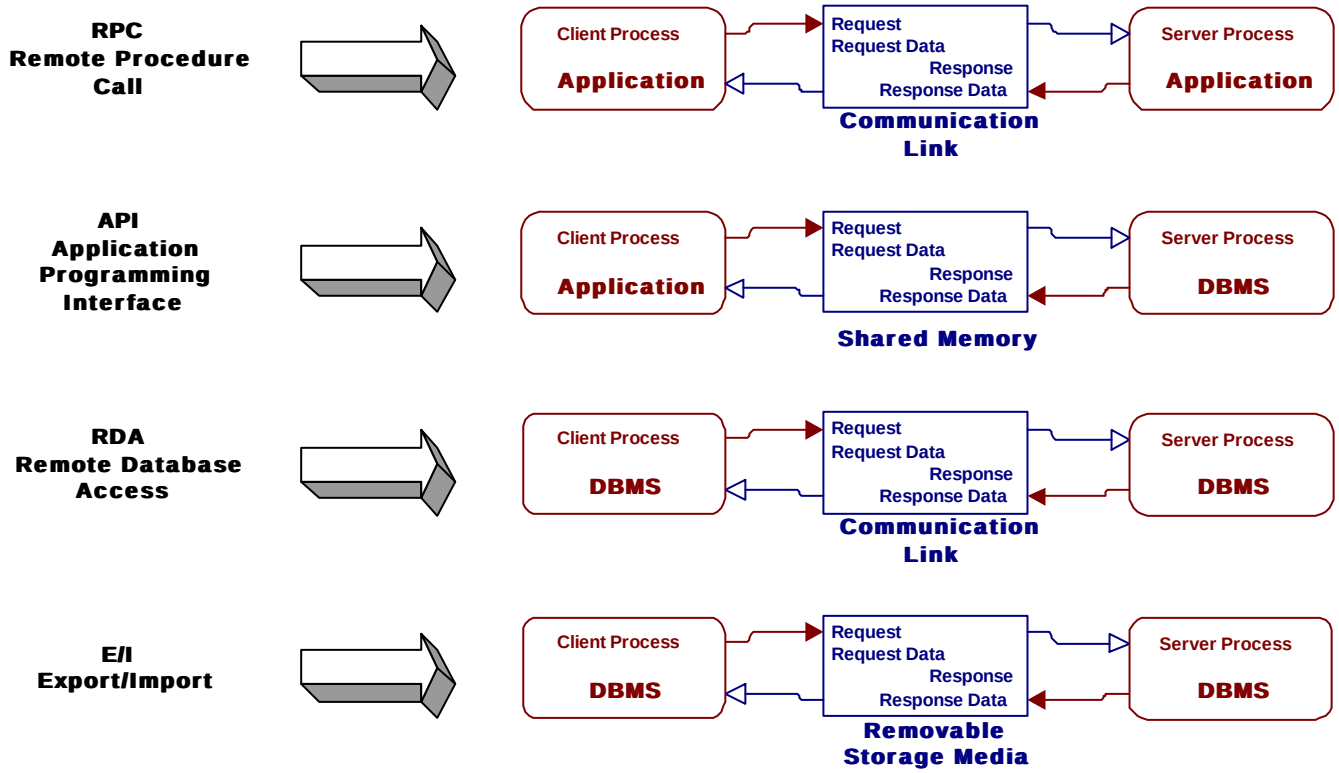

Figure 3 - Specific Client Server Communication Models

6.1.3 Interchange model

For data to be transferred from one data management system to another, it is not necessary for there to exist a common representation, or format, for the two data management systems. It only requires that the separate representations be described in a way that is understandable by the two systems.

This framework proposes a means of establishing a common understanding using an intermediate Export/Import transfer format. The export function will transform the data from the exporting system's format to the Export/Import transfer format, and the import function will transform the Export/Import transfer format to the importing system's format.

In order to transfer data, there must be a clear understanding between the exporting system and the importing system of the meaning of the data to be transferred.

In the context of the Reference Model for Data Management (ISO/IEC 10032), data may be transferred at level n if, and only if, there is agreement on the definition of that data at level n+l. At some level, this agreement must be established independently of the Export/Import facility (i.e., by using the native facilities of the data management systems). However, once a common data model, a mapping, or similar agreement has been established at one level, the Export/Import facility may be used iteratively to establish such agreement at successively lower levels.

The existence of commonalty at one data level does ***not*** imply that either higher or lower data levels also share commonalty.

6.1.4 Components of complete data interchange

The components are:

— Data messages (record instances, table rows, values...);

— Definition of the data messages (Record types, Table column types, encoding, domain, labels, tags...);

— Semantic of the Data Messages (Global schema (Tables), references, rules, constraints, etc.). A conceptual schema could be exchanged, if it captures more of the real world semantics than the database (DB) schema.

6.2 Export/ Import Architecture

6.2.1 Overall Architecture

The possible interactions (in the export/import domain) between two systems (the source and the target) are illustrated in figure 4.

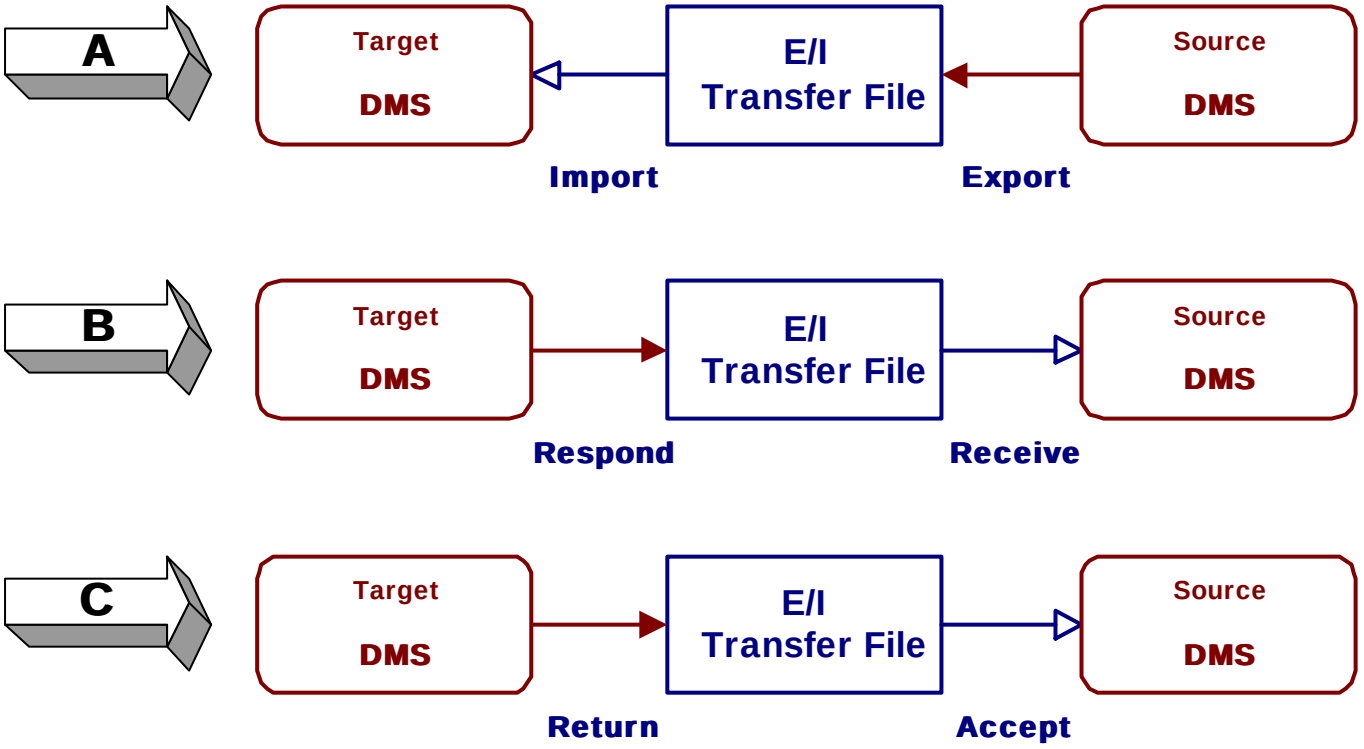

Figure 4 - Transfer-Related Operations

The basic interaction (A in figure 4), is what is identified as Export/Import. However, when the two DMSs are bound by some convention (for instance replication, or booking), there may be further interactions, illustrated by (B) and (C), and described below.

6.2.2 Response Mechanisms

As a result of accepting (importing) some data, the target system can produce a response, and send it back to the source system. That response can be carried in a specialized transfer file. This also requires specialization of the export process to prepare the response, and of the import process to receive the response.

The imported data can be returned to the source system at a later time. For this to be done in an orderly fashion, control information is required. This can be carried back in a transfer file with the proper fields populated. This also requires specialization of the export process to prepare the data to return, and of the import process to accept and process the returned data.

6.2.3 Human-readable E/I transfer file

Any information required to process the E/I transfer file that is not included within the automated file has to be communicated by the exporter to the importer as human readable information.

6.2.4 Transfer of Data Schema

When the schema (definition) of the data needs to be exported with the data, it is done using standardized tables, such as the SQL Information Schema (ISO/IEC 9075), or the IRDS IRD Definition tables (ISO/IEC 10728). It can therefore be manipulated using the same mechanisms as the application data. A different component is added to the transfer file.

In the absence of standardized ways of transferring the full semantics of the transferred data, over and above what is possible to convey in a database schema (for instance as SQL Data Definition), such semantics has to be transferred as human readable material.

6.3 Processes Associated with Data Management Export/Import

The transfer of data from one Data Management System to one or more other Data Management Systems can be divided into major functions, each of which is described briefly below.

6.3.1 Identification of the nature of the Export service requested

Requesting an Export service involves selection of different options, associated with different types of service requests.

6.3.2 Selection of data to be exported

The process of data selection precedes the export itself. Complete or partial selection criteria must be supported, as the manner in which the selected data is to be identified to the export function.

Many possibilities can be considered, such as:

— Selected data without schema;

— Selected data with associated schema;

— Selected schema without data;

— Selected schema with associated data.

6.3.3 Copy of the selected data to the transfer file

Whichever of the four export options is chosen, the exporting Data Management System must copy the appropriate data from its data in its database environment into a transfer file and also format the data in the transfer file according to rules defined in subsequent parts of this international standard.

Note: This part of this international standard recognizes the possibility of including other data in the transfer file, or updating the data in the exporting system. At time of publication, such capabilities are not yet supported in the subsequent parts of this international standard. Examples of such capabilities are:

— Addition of action codes and other protocol information;

— Addition of status, dates and other administration information;

— Marking of the selected data as 'booked out' (AKA 'checked out'), if there is an intent to return the data later, with updates;

— Marking of the selected data as 'replicated to'', if the intent is to refresh a copy of the data.

6.3.4 Export of the schema along with the data.

Some export options cause the schema information to be exported. This is a process analogous to the previous one.

6.3.5 Making the transfer file available to the importing system

This international standard does not define how the data in the transfer file has to be acquired by an importing data management system.

The manner in which the file is made available is diverse, and usually falls within general methods such as:

— transferring the physical file on removable media,

— transmitting the data to a different physical file at the importing system, or

— sharing the file in one system.

In all cases, the file is considered to have been logically transferred.

6.3.6 Identification of the nature of the import request

Requesting an Import service involves selection of different options, associated with different types of service requests. Furthermore an import service must recognize the alternative export options used (as described in clause 6.3.2).

6.3.7 Import of schema information

If present, schema needs to be processed first. This processing might be only a consistency check, or might actually involve, if such a facility is used, the definition of a database in which to load the data, or the definition of the transfer format as a view or a table to the receiving database.

6.3.8 Import of data to the importing Data Management System

The data will be converted from the standard transfer file format, into the format used by the Importing System. If more conversion is required, then it is conceivable that a conversion service might be required.

This function optionally supports the following features:

— Trial import;

  A trial import is provided to allow the execution of the import function for validation purposes only, without committing any updates.

— Book-in facility ('Check-in');

  If a Data Management system supports a book-out facility as part of the export function, it also supports a book-in facility as part of the import function. Data that has been previously 'booked out', is imported only when 'book in' is specified.

— Data Merging;

  There may be an option to import data into an already populated database. Where particular data already exists in the database, and is also on the import file, the following options are available depending on parameters specified when the import was initiated, and/or commands embedded in the import file:

  — Reject;

— Overwrite;

— Create a new version.

7 Standardized components of a DM Export/Import facility.

7.1 Standardization Approach

Clause 7 identifies components of a DM Export/Import facility that are candidate for standardization, and this clause summarizes the architecture rules applicable to such components.

Standardization of these components is done within other parts of this international standard, or within other standards.

7.1.1 Applicability to all level pairs

Data base schemas are just one kind of data. They should be exchanged using the same formats and mechanisms as data (as schema table definitions, for instance).

7.1.2 Applicability to all modelling facilities

The transfer format is considered an external presentation format. It is analogous to a screen layout, where data elements are arranged as fields, but may be structured in a different way from that for storage.

As such, an Export/Import facility has a set of data structuring rules, and a set of processes.

The data structuring rules used in the transfer format should enable isolating the exporter from the importer.

The best candidate for those data structuring rules is a subset of the data structuring rules of SQL. Organizing data in tables, with columns, and making associations between tables explicit, as table references should be robust enough to transfer any data.

7.1.3 Applicability to all applications

There is something common to the export/import of data, independent of what the data describes. This is why the core of export import is a structured transfer file with its own structuring rules, and the export/import format and basic services should be independent of the applications (including IRDS).

A definition of the E/I file should be independent of the services that created the file. That is the same file definition method should be useful, whether the file contains SQL data, IRDS data, or application specific data.

7.1.4 Modularity and Partitioning

The first partitioning required is to separate the specifications for the Export services, the Import services, and the transfer format. Whether these become different clauses in a standard, different parts of the same standard, or different standards depend on the level of reuse possible for each specification.

For instance a transfer format could be generic enough to be referred to in many application standards, and deserve its own standard, whereas the set of operations to create an export file, or receive one, could conceivably be specialized by applications, and simply be part of another standard, for instance the statements required to add export/import facilities to SQL.

7.1.5 Levelling of complexity

Inside the three sets of specifications (import, export, transfer format), there should be different standardized levels of sophistication, so that compliance to a simple standard, to answer simple requirements, is possible.

Furthermore there should be a form of downward compatibility, such that even if a file was created using a sophisticated export and file format, embedding commands, and administration information, the same file can be used as input for a simple import, who could ignore the information that is not needed.

7.1.6 Profiling and Conformance

Not all components of an export/import need be standardized, nor should an Export/Import facility be forced to comply with all possible standards. Specific implementation could conform to various profiles. It should be possible to read/write a standard export/import file by a non-standard application with that application not using any data management or communication facility. That is, the minimum level of compliance is the export/import file format alone.

7.1.7 Integration with the environment

There is something common to all transfer of data between applications, from a bound local API using shared memory, to SQL CLI or RDA protocols, and to a connectionless/protocol-less export import. These are just specializations of a set of basic structuring rules and operations that are applicable to any data (byte) stream.

7.2 Architecture of the E/I standards

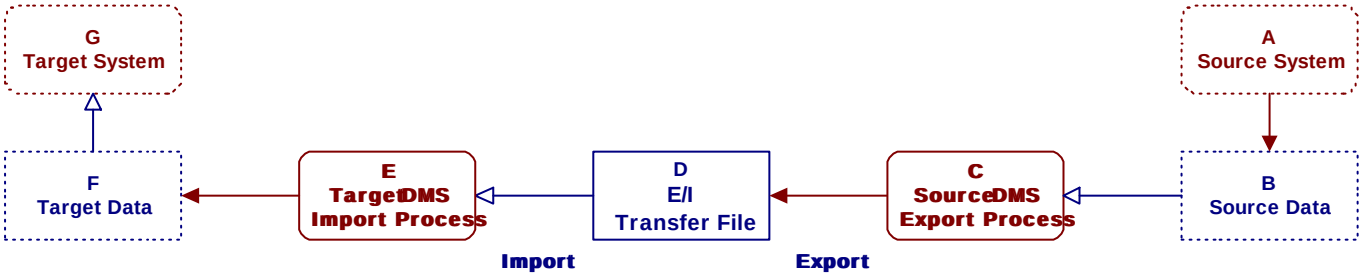

Figure 5 - Export/Import Standardization Areas

The first architectural subdivision is based on the nature of the potential standards. The categories used are:

a) The transfer file format used for transfer (D in figure 6);

b) The services offered by the process creating (exporting) the transfer file (C);

c) The services offered by the process accepting (importing) the transfer file (E).

The second dimension is the level of complexity of the services, and consequently of the required transfer format. The following levels have been identified:

a) The transfer file format used does not contain action request, or check-in/check-out information;

b) The exporter includes action requests in the transfer file format;

c) The exporter expects the importer to process action requests and possibly provide a response;

d) The exporter and the importer want to keep their data synchronized through multiple transfer cycles, and therefore include and process check-in/check-out data.

The third dimension is based on the relative independence of the transfer tables vis-à-vis the table definition as present in the importer (F in figure 6) or the exporter system (B). The following levels are identified:

a) The transfer tables correspond to base tables or predefined views in the importing and exporting system, and the data definition of the base tables have to be shared between the two systems, in addition to the data itself. In summary, one transfer component is not self-defining.

b) The transfer data tables do not correspond to predefined views or base tables from the exporting system. These transfer tables may or may not correspond to predefined views or base tables in the importing system. To enable processing by the importer, the definition of the transfer table must be present within the transfer component, and the transfer component is self-defining. For each table, the definition of the columns precedes the collection of rows.

The fourth dimension is based on the level of standardization of the exporting (A and C in figure 6) and importing (E and G) applications. The impact on the E/I standards is that if these applications refer to already standardized data components, then only references need to be transferred, in lieu of the detailed definitions. The following cases have been identified:

a) The exporter does not assume for the importer a standardized data management system or a standardized application. In such a case, the transfer file has to be portable across applications and across data management systems;

b) The exporter assumes for the importer a standardized application, such as banking, library, IRDS, CDIF;

c) The exporter assumes for the importer a standardized data management system;

d) The exporter assumes for the importer both a standardized data management system and a standardized application.

7.3 Interface and Communication Standards

7.3.1 The Interchange Modelling Facility Definition

Data is generally exchanged in files, made of records. Record types are made of fields, and each field associated with a value in the record. For the purpose of this standard, the SQL (relational) terminology is used, both for practical and consistency reasons.

The consistency reason is that if this is done, then the interchange format can be expressed as a view over a database schema. This facilitates the positioning of the export/import standard in the family of standards required for a standardized data management environment.

The following terminology equivalence is used:

— “Table” is used instead of “record type”;

— “Row” is used instead of “record”;

— “Column” is used instead of “field”.

The practical reason is that such a format is nearly identical to what is in use, and can accommodate the data part of most modelling facilities. For instance data describing entities and relationships (as in the ER model) can be exchanged using two types of table in the transfer format.

7.3.2 The E/I Transfer File

Because an actual exchange can include many different "files", a structured format has to be used. The complete exchange is called a file, a file has components, and components contain the transferred data. Structural information and headers are possible at each level. Note that this is a convention for the purpose of definition and specification of the transfer format. The term file should not be equated to "operating system" file.

The overall structure of the Transfer File is illustrated in figure 6.

One characteristic of this structure is property inheritance. When the same property, for instance the "data manipulation action" (insert, delete, select, update, compare), is specified at one level of the structure, it becomes the default for all contained (inside its scope) structural units (which inherit it). It can be overridden by the specification of another action code inside its scope.

For instance an action code specified on one row of a Transfer Data Table could override any other action code applicable to all rows in the file.

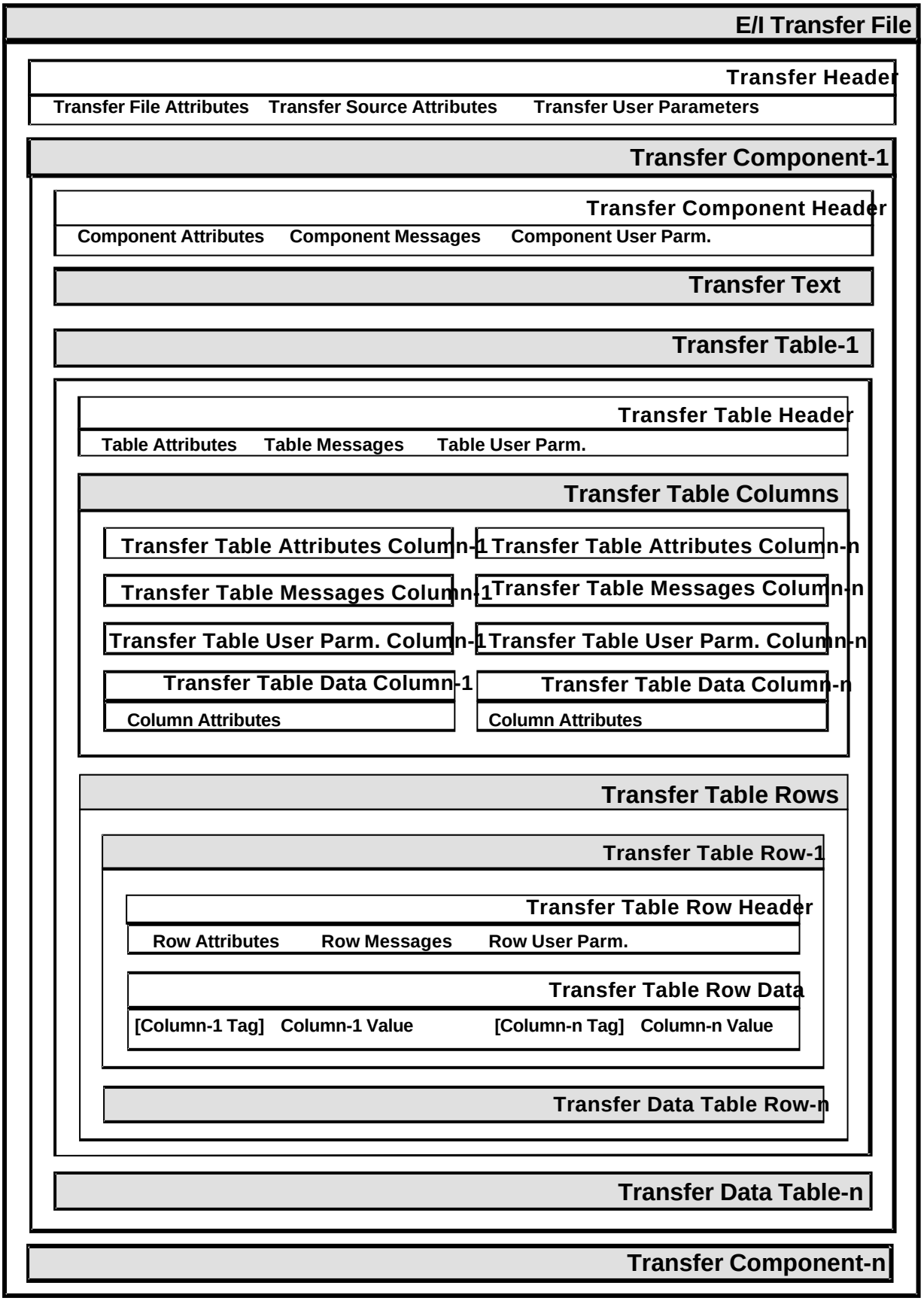

Figure 6 - Overall structure of the E/I Transfer File

The structure of the transfer file can also be described as follows:

- one transfer header
  - transfer file attributes
  - transfer source attributes
  - transfer user parameters
- one or many transfer components.
  - one transfer component header
    - component attributes
    - component messages
    - component user parameters
  - either one transfer text,
    - one text stream
  - or one or more transfer tables
    - one transfer table header
      - table attributes
      - table messages
      - table user parameters
    - none, one or more column definitions
      - transfer table attribute columns
      - transfer table message columns
      - transfer table user parameter columns
    - one or more table rows
      - one row header
        - row attributes
        - row messages
        - row user parameters
      - none, one or many row columns
        - none or one row column tag
        - one row column value

7.3.3 Transfer header

The Transfer Header contains the information required to identify and process the transfer. Its content is classified in three groups of attributes:

a) transfer file attributes,

b) transfer source attributes, and

c) transfer user parameters.

The Transfer Header contains only characters from Table 1 of ISO/IEC 10646-1, because the transfer header must be understandable for any parser. Each syntax shall support the use of the characters in the character set (CODESET) specified in the transfer header for values of string and text meta-attributes within a transfer. If no character set is specified in the transfer header, then the Basic Multilingual Plane (BMP) of ISO/IEC 10646-1 character set shall be assumed, with the character coding being the UCS-2 coding of ISO/IEC 10646-1.

7.3.3.1 Transfer file attributes

DMEI signature:	a constant to identify the file as a standardized Data Management Export/Import transfer.
DMEI title:	a constant to identify to which part of the standard the file is conforming to.
DMEI syntax identifier:	this attribute identifies which set of structuring is used for the file (e.g. BNF, ASN.1, XML, …).
DMEI encoding identifier:	this attribute identifies how primary tokens of the syntax are encoded in the file (e.g. CLEAR for readable text, BINARY, …).
DMEI codeset identifier	this attribute identifies the codeset and the codeset encoding used in the transfer file components, for instance "ISO/IEC 10646 part-1 level-1 UCS-2", or ISO 646.
file timestamp	this attribute provides the date and time that the file was created in a standardized format (as defined in ISO 8601). This is always expressed in UTC.

7.3.3.2 Transfer source attributes

source name	This is used to identify the person or group responsible for the preparation of the transfer file, generally designated as the "publisher".
source system name	This is the unique identifier assigned to the source's system, such as an OSI Directory name, a DNS name, or an IP Address. Besides its documentation utility, such an attribute enables the generation of unique keys at import time, to facilitate processing and integration in the target system.
source file version	This is a version number assigned to the file by its publisher.
source environment	This identifies the hardware/system software platform of the program used by the publisher (e.g. Intel Win95, MAC, SUN Solaris).
source program name	This is an identifier for the exporting tool or product.
source program version	This enables the exporter program name to be qualified with a version identifier.

source program vendor	This identifies the provider of the program used by the publisher.

7.3.3.3 Transfer user parameters

The user parameters are optional. The value of these, if present, is freely determined by the exporter. The selection criteria, scope of export and other values may be included as a user parameter.

7.3.4 Transfer component

The Transfer Component is the unit of transfer. It is made of a header and a content section. The header includes a set of descriptive and control attributes, and a component transfer message/response section (used also for Response). The content is either text (for instance a set of SQL statements) or one or more transfer data table sections.

Components can contain data, or what is identified as definition data, that is metadata defining some other data, present or not in the same transfer file.

7.3.5 Transfer component header

The transfer component header contains the supplementary information required to identify and process the component. Its content is classified in three groups of attributes:

a) component attributes,

b) component messages, and

c) component user parameters.

7.3.5.1 Component attributes

Typical transfer component attributes are:

component type	This attribute identifies if the component was considered application data or definition data at the time of the creation of the transfer file. Note that this typing is relative at import time. For instance definition data (the database schema) for the export of application data (a client database) could be considered application data by an importer who wants to import it into a dictionary.
component title	This identifies the standards to which the content of this component conforms. For instance the content could conform to the SQL standard (ISO/IEC 9075)
component identifier	This uniquely identifies the component within the transfer file
component name	This is a name that uniquely (within the context of the transfer file) identifies this component.
component definition name	This is a name that uniquely (within the context of the transfer file) identifies the component where the definition of the data in this component can be found.
component structure	This is where one would indicate that the component is made of unstructured (within the context of the EI transfer file) text, conforms to some language syntax, or is structured in tables).
component syntax identifier	This attribute gives the opportunity to override the equivalent parameter in the transfer header.

component encoding identifier	This attribute gives the opportunity to override the equivalent parameter in the transfer header.
component codeset identifier	This attribute gives the opportunity to override the equivalent parameter in the transfer header.
component row column tags	This indicates a default option for the presence of tags in the table section of the component. It has values such as FULL, when the full name of a column precedes the corresponding value in each row, to UNLABELLED, when such tags are absent. Other options are user-defined tags, abbreviations, numbers….

7.3.5.2 Component Message

This section is used to describe the control elements applicable to the transfer component. These control elements are different whether the component is a transfer component, a request component, or a response component. Typical message, response and control elements include:

component action	This could be, for instance, an SQL <<Query Expression>>, if the purpose of the component is to cause the export of the result of such query from the target system. It could also be a reference to a method or operation.
component response	This could be a return code in a response component, giving the exporter the result of an import.
component transfer protocol	This could identify the presence in the component of control records, such as checkpoints, checksums, etc.

7.3.5.3 Transfer User parameters

The user parameters are optional. The value of these, if present, is freely determined by the exporter.

7.3.6 Transfer Data Text

This type of component enables the transfer of bulk text between the exporter and the importer. In such a situation, the syntax of this text may be completely implementor or user defined, or may follow other industry standards.

7.3.7 Transfer Data Tables

This section contains the data accompanying a request, or a response, when required. The term "Transfer Table" is used, to benefit from the already defined relational (SQL) terminology. However the E/I transfer format is independent of the source and target system modelling facility, and comprehensive enough to transfer any of the ones currently in use. A component can contain many tables.

The Transfer Table has a header section containing descriptive or control attributes, an optional section containing the definition of the transfer data table columns, and a section containing the rows of the table.

Different syntax, and corresponding encoding, schemes can be used for the transfer data tables. These schemes have different level of encoding/decoding complexity, and different level of efficiency, when a large number of rows are transmitted. The range varies from the case where the full column definition (name and domain) is carried with each column of every row, to the case where each row contains only delimited values, and matching with column definition is based on position.

7.3.8 Transfer Table Header

The transfer component header contains the supplementary information required to identify and process the component. Its content is classified in three groups of attributes:

a) table attributes,

b) table messages, and

c) table user parameters.

7.3.8.1 Transfer Table Attributes

Typical Transfer Data Table Attributes are:

table name	This is a name that uniquely (within the context of the transfer file) identifies this table.
table type	This relates the semantics of the data group represented by the transfer table in the source system. For instance, if the source DMF is E/R (Entity-Relationship) then at least two table types would be used, one for entities, and one for relationships. Similar considerations would apply to records and sets in a network model.
table key type	Identifies whether the table has a unique identifier or not, and whether it is the concatenation of other columns, or a special column. The term key is used here to designate the unique identifier of a row, and has no access connotation.
table rows column tags	This enables the override at the table level the default option set at the component level for the presence of tags in the table section of the component.
table unique identifier	If agreed between parties, this unique identifier enables transfer without either a definition component, or transfer column definitions. This name could also be a registered, or standardized name. For instance the SQL Information Schema Table, or application level registered messages (as in EDI).

7.3.8.2 Transfer table message

This section is used to describe the control elements applicable to the transfer table. These control elements are different whether the component is a transfer component, a request component, or a response component. They override similar attributes set at the component level. Typical message, response and control elements include:

table action:	This could be a request to delete the table, to add to or to update it.
table response	This could be a return code in a response, giving the exporter the result of an import for that table.

7.3.8.3 Transfer table user parameters

The user parameters, are optional. The value of these, if present, is freely determined by the exporter.

7.3.9 Transfer table columns

This section contains the definition of each of the possible columns in the transfer data table. Four types of columns can be defined:

a) transfer table attributes columns,

b) transfer table message columns,

c) transfer table user parameters columns, and

d) transfer table data columns.

7.3.9.1 Attributes Columns

If required the following attribute can be used to uniquely identify each row:

row key	Added to each row, if required, to provide a unique identifier

The following columns are required if data is loaned (checked-out, booked-out) to the target system, with the intent of processing an eventual return.

source system last action code

source system last action time stamp

target system last action code

target system last action time stamp

Similar attributes would be used to support replication.

7.3.9.2 Message columns

Standardized column definitions (action section) are added to the application data columns. Their reserved column names are:

row action code	This identifies what the importer is to do with the values in the rows. It refers to standardized operations (ADD, DELETE, UPDATE, SELECT), or other methods or procedures.
row response code	This column enables the importer to confirm the result of the import by creating a response component with this additional message.

7.3.9.3 User parameters columns

Columns added by the exporter (other than data columns) are defined in this section.

7.3.9.4 Data columns

This section contains the definition of the data transferred, as opposed to the previous columns, which were targeted at the import process.

7.3.9.5 Attributes of columns

For each of the four types of columns, the following descriptive attributes could be used:

column name	This is a name that uniquely (within the context of the transfer file) identifies this column
column registered identifier	This could be used for matching columns to standardized data elements, across columns, across systems and applications;

column data type	Integer, character, etc.
column domain	used if domains are registered between parties. If applicable to a specific file, then a Transfer Data Table Domain section need be added;
key column sequence	If the column is part of the key, then this attribute identifies the concatenation sequence.
column tag	If column tags are used in the rows, then this attribute can be used to define a short form tag. It will be used to map columns in rows to the corresponding column definition;
column sequence	If no tags are used, this attribute enables the processing of the sequence of unlabelled column values in the row.
column position	If no tags are used, and the values appear in the row at specified positions, then this attribute combined with the following one (column length) identifies the beginning and the end of the value in the row.
column length	see above.

7.3.10 Transfer Data Table Rows

This section of the component contains rows of values corresponding to the transfer table columns.

If that correspondence is established by position, then each row contains only a set of values. In this case, the values could appear at fixed positions, or be separated by delimiters.

If the correspondence is established by the presence of tags, such as the name of the column, a user defined abbreviation or a sequence number, then each rows contains a set of pairs of values (column tag, column value). Since this is done to avoid the restriction of a fixed format, and the waste of space when a column has no value for a specific row, the values and the tags are separated by delimiters.

Each row is made of a header part, and a data part.

The transfer format for rows has three complexity variants:

a) The basic format contains only data, without any reference to a process to be performed by the importer;

b) The second level contains reference to actions to be performed by the importer, and possibly responses to be sent back. Examples of possible actions are data management operators, such as add, delete, update and query.

c) The third level contains control information to enable book-in, book-out (which is a kind of delayed commit), replication, version control, etc..

7.3.11 Transfer table row header

The row header section contains, for each row, the value corresponding to the header columns as defined in the columns section.

7.3.12 Transfer table row data

The row data section contains, for each row, the value corresponding to the data columns as defined in the columns section.

7.4 Service and Process Standards

Although the discussion below tends to describe export and import in the most simple case, where export corresponds to an extract, and import to the population of an empty database, one must consider this as a special case.

Sending a query expression, or a set of these, and returning query results, is also a special case of export import. Similarly, having an importing system return to the exporting a set of responses to indicate success or failure of importing operations is also a special case.

Although the file format has to be independent of the nature of both the exporting and importing systems, the definition of the services is not. The following description of services is intended to be generic, although examples are drawn from SQL environments.

7.4.1 Export Services

7.4.1.1 Basic Services

A simple set of services provides for the definition of the export file and a population by selective extraction. Defining the export file could be done by a mechanism similar to the current SQL mechanism to define temporary tables (views). Selection and population by a mechanism similar to the SQL SELECT. The following general capabilities should be supported:

a) Selection of some or all the data types defined by the specified schema, subschema and user view;

b) Selection of some or all of the data values that exist for the selected data types.

The manner in which selection criteria are to be specified has yet to be determined. The export standard should use an abstract specification technique. Specialization of this standard may use concrete specification techniques.

An export file must be created, into which the data to be exported will be placed. The manner in which the file is created and identified is implementation dependent.

It is **usual, but not** necessary, for the export file be created on the same real system on which resides the Exporting System. To do otherwise implies some sort of remote data access capability, which is not described by this framework. Such a capability would effectively combine the file transfer function with the export function.

Every export file will contain components, and file header information. These will be described in file format standards.

The data selected in the previous steps must be converted to the standard Export/Import File format. This conversion process is data model dependent, and the rules for any particular data model are contained in the specialization of this standard.

Encryption of the export file is optional, but is recommended to maintain data security while the file is being transferred from one system to another.

7.4.1.2 Schema Services

Schema information may be part of a transfer, either as a side product, accompanying the data, or as the main product. Schema information has to be understood as:

a) a reference to a schema known to both parties;

b) extensions to a schema known to both parties;

c) view definitions, making one schema a view on another schema;

d) base table definitions.

The term schema information is used here in an abstract manner, and not in the SQL sense, where it is a partitioning of the schema (meta) information.

A data transfer might require extraction of schema information from many SQL "schemas".

Schema information uses the same interchange format as data. Definition of the file to exchange schema information is standardized, not user controlled (based on schema information tables), and selection and population should be automatic in some scenarios. Thus a very simple service definition, or an option on the data transfer service, is sufficient.

The importing system must be able to identify the schema to which the export file contents conform. Either the identification must have been previously agreed between the two systems, or the schema information must be transferred with the Export file. This is a recursive problem. At some level, the two systems must already have a common understanding, as discussed before.

This framework does not restrict the number of data levels for which data may be exported at one time. For example, data from level 'n' can be exported as the schema for level 'n-l', which in turn is exported as the schema for 'n-2'. The Export/Import file format reflects this recursive structure. However, specialization of this standard may restrict the number of data levels that may be exported at one time, as may be appropriate for the Data Management Systems they are intended to support.

Before an export function can be initiated, the schema, subschema and user view to be used must be identified.

7.4.1.3 Administration services

A third level of services would include control information with the data exported, and also update the exporting database with control and status information.

A 'book-out' facility is provided for situations where the exported data is to be re-imported at a later date, and it is desired to prevent changes to the data within the exporting system in the meantime. Each data item exported will be marked as 'not-updateable' until such time as the data is re-imported, or the mark is explicitly removed.

7.4.1.4 Manipulation services

The Export/Import file may contain embedded commands to control the action to be taken by the importing system. These commands could apply to the whole file, be applicable to each record in the file, or to sets of related records.

The basic data management operations (get, modify, add and delete) have to be supported. A facility is needed to enable the transfer of information about data objects that have been 'logically deleted' at the Exporting System, in order that they may be similarly deleted at the Importing system.

The import function may specify periodic commit points based on criteria such as number of records updated. The exact definition of a 'record' will be specified in the appropriate standards.

This facility must take account of logical dependencies between records. (e.g., A referencing object cannot be committed unless the referenced object has already been committed, or is to be committed simultaneously.)

7.4.2 Import Services

In terms of data management services, we can think of three levels of complexity of import. Level one requires only one service, and it populates an empty database (creating it from the schema information, or schema reference provided). Level two can insert the data in an existing data base, coping with operations such as add, delete, update, either implicit or explicit (encoded in the export file). Level three would cope with control, booking and version information.

The data in the import file is to be copied to the importing Data Management System. The importing system may or may not already contain data. Several possibilities exist, such as:

a) Import into an empty database;

   This is the simplest situation since it requires no comparison of old and new data, but it is of use only when creating a new copy of an existing database.

b) Import into an already populated database, using 'Book-out/Book-in';

   This facility allows the update of data that was previously exported for exclusive use.

c) Import into an already populated database, using 'Data Merging';

   This facility allows the integration of an unsolicited import file with the existing database contents, based on certain specified rules for the action to be taken when duplicate records are encountered.

Only one possibility may be used for any one invocation of the import function.

Errors may be detected during the import function. The following error handling options will be specifiable:

a) Halt the import immediately, and backout all changes to the last commit point;

b) Report the error, but continue to allow detection of other errors. Do not commit any further changes, and backout existing changes to the last checkpoint;

c) Report all errors encountered, but apply all other updates as if no error had been encountered;

Reporting errors, or even success, might involve the creation of a new export file, containing the service responses to the service requests contained in the import file.

It should be possible to evaluate the impact of an import, before actually committing to the changes. This can be done by a "trial import" facility.

The purpose of this function is to verify that the import will execute successfully. All validation performed during an actual import will be performed during the trial import, including but not limited to:

a) Ensuring that the data in the import file is in fact compatible with the importing system's schema;

b) Ensuring the validity of all embedded commands;

7.4.2.1 Basic Services

The import file header may contain a schema name identifying the schema at the originating system. Verification of the schema name can be performed if the name is meaningful to the importing system.

The manner in which the importing schema is identified will depend on the manner in which the Import is initiated. This will be specified in the appropriate standards.

Note: It is not necessary that the originating schema have the same name as the schema at importing system, only that the two schemas be compatible.

If the file was encrypted, then decryption must take place.

7.4.2.2 Schema Services

If present, schema information has to be processed first. Processing could range from simple identification, to update, or creation of a new schema and corresponding database.

It could also be a verification process, whose purpose of this function is to ensure that the schema used to create the export file is compatible with the schema to be used to import the data.

The schema-to-schema compatibility check will usually be performed only when initially establishing an understanding between the two systems, or when one or other of the schemas are changed, and the database administrator wishes to verify that they are still compatible. If Export/Import is to be performed on a regular basis between the two systems, this function may be bypassed.

7.4.2.3 Manipulation Services

If data is imported into an already populated database using the 'Data Merge facility', the action taken when a match is found on keys will depend on the options specified. Possible options include:

— Reject the incoming occurrence;

— Replace the existing occurrence with the incoming occurrence;

— Reject or replace depending on a comparison of date-time stamps. (keep whichever is more recent);

— Create new version based on incoming occurrence, while retaining existing occurrence.

In the case where specific commands are embedded with the records, then this facility is not used.

The import file may contain embedded commands, the format of which is prescribed by a file format standard. These commands include:

a) Logical delete;

   This command indicates that the associated data item has been logically deleted at the exporting system, and is to be logically deleted at the importing system also.

b) New version;

   This command indicates that a new version of the corresponding data item is to be created, rather than replacing an existing version. This is mutually exclusive with 'Book-in'.

c) Update;

   This command indicates that an existing data item is to be updated.

d) Add;

   This command indicates that a data item should be absent from the importer data, and added.

e) Select;

   This command cause the creation of a response export file, with the selected data items.

7.4.2.4 Administration Services

Data that has been exported using the Book-out facility may be imported using the Book-in facility. When data is imported into an already populated database using the book-in facility, any existing data whose keys match those of some incoming data must be marked as "booked-out" to the system from which the incoming data originates. If this is not the case, then that portion of the data will be rejected.

f) Book-in;

This command indicates that the associated data is being returned to a Data Management system from which it was previously booked-out.

The import function may specify periodic commit points based on criteria such as number of records updated. The exact definition of a 'record' will be specified in the appropriate specialization of this standard.

This facility must take account of logical dependencies between records. (e.g., A referencing object cannot be committed unless the referenced object has already been committed, or is to be committed simultaneously.).

g) Commit;

This command indicates that the data imported up to this point represents a logical unit of work and can be committed independently of the remainder of the Export/Import file.

7.4.3 Convert Services

The definition of the structure of the E/I file is not the same as the definition of the schema of the exporting or importing database. In fact the general case is where all three are different. (but compatible, that is the E/I structure can be expressed as a view on both exporting and importing database).

These differences may be syntactic, such as different naming conventions, different value sets, etc.

There may be a requirement for a standard way of specifying conversion tables, and a standard service definition. Since the E/I file format is standardized, it is possible to produce such standards.

Such services could also do the mapping between data modelling facilities, although that might be out of scope of this work

7.5 Semantic and Content Standards

A separate set of standards is needed for the exchange of the definition of the data. These will not only have container specifications (which would probably be the one specified above), but also content definition.

This means that it is expected that something analogous like the definition schema table could be used to exchange schema definition, even between non-compliant SQL systems.

Annex A Annex A
(informative)

Rationale and Requirements

A.1 Audience and User Communities

Organisations concerned with standards fall into two categories - *users of standards* and *providers of standards.* Both may form the audience for standards products.

Standards users can be categorised as standards implementors, standards beneficiaries, and certification bodies. *Standards implementors* are the providers of products and services which conform to the standards, while *standards beneficiaries* are the procurers and users of those products and services. *Certification bodies* undertake conformance testing and the certification of products and services.

Implementors may implement a standard internally within their organisation, in which case they constitute a *single user*. Cases where provider and client agree to use a standard for a particular contract constitute *dual users*. *Multiple users* arise where widespread consensus exists, where particular suppliers dominate a market, or where governments and other regulatory bodies impose the use of standards on procurers and/or providers (for public health and safety, trade regulation, etc.).

Standards providers are those organisations concerned with the production, registration and distribution of standards products. *Standards makers* are the committees and groups of a standards organisation, while the standards organisation itself acts as the *publisher* of standards.

A.2 Current Issues

Although many products/standards have similarities (overall architecture, services offered, encoding mechanisms), nobody offers, in this area, a solution that is so widespread or exceptional that we can anticipate the emergence of a unique approach and solution.

Further more there are no generic data management export/import mechanisms.

However this area appears conceptually simple, and one has to conclude that the absence of any widely used transfer standard is due to the fact that there are more important, or spectacular issues in data management.

Traditionally, users and manufacturers have coped with ad hoc export, or import, by ad hoc programming or facilities. It is only since the explosion of personal computers, and the introduction of distributed architecture, that a generic facility is becoming essential.

A.2.1 The Interested Parties

As in all IT standardization efforts, the interested community includes:

— Users

— Procurers

— Suppliers

— Standards makers

A.2.2 The Current Situation

There are four categories of standards germane to export/import:

a) Application Specific standards

- Application Data Standards (Banking, Library, etc.)
- CDIF -- EIA (Standardized CASE Interchange Metamodel)
- EDI
- XMI

b) Enabling and support standards

- SQL
- ASN.1
- XML

c) Related Standards

- RPC
- RDA
- MHS
- CLI

d) Architecture standards

- RM for DM
- RM for OSI
- RM for ODP
- RM for EDI

Among the different standardization groups, most everybody, except the Data Management group, have been active one way or another in the area of interchange. However, in the absence of any recognized standard or approach, everybody is tackling the issues in a different manner.

In the industry, there does not seem to be any form of common export/import format and/or facilities among the suppliers of major hosts or LAN-based DBMS. However, in the PC world, the dBase file format, which is public domain, is becoming the de facto standard for exchanging tabular data, not only between DBMSs, but also between other packages, such as word processors and spreadsheets.

A.2.3 The Perceived Trends

There is no evidence resulting from the current trends and directions in the world of data management, and the world of standardization, pointing to the availability of a standard, and associated products on the horizon.

This means some initiatives have to be taken, by users, suppliers, and standardization organizations, to materialize some of the potential benefits, while avoiding painting oneself in a corner.

A.3 Expressed Requirements

Different communities have different needs and perspectives in this area. Users and suppliers sometimes have conflicting interests, and those acting as procurers, that is those acting on behalf of users in dealing with suppliers, are caught in between. Again basic data transfer requirements have to be separated from standardization requirements.

Requirements can be categorized as follows

a) User Requirements for export/import and more generic transfer mechanisms

   — Productivity, work organization, sharing and control, integration...

b) Standardization Requirements

   — Productivity, Effectiveness, Communication, Openness, Quality

c) User Requirements for Standards

   — Communication vs. Portability

   — Metadata transfer

   — Homogeneous vs. open environments

   — Timing requirements for data integration

A.4 Benefits of Standardization

There are two primary reasons for standardization in any domain. These are *consumer protection* and *communication*, both of which have social and economic aspects.

A.4.1 Consumer protection

Consumer protection embraces quality, productivity and effectiveness.

A.4.2 Communication

Communication embraces both human and machine communication. *Openness* is the term most often used for automated processors which are able to communicate successfully (cf. OSI). Openness includes interoperability, portability (of people and tools) and scalability.

Standardized formalisms for models, specifications and documentation facilitate the communication of knowledge and the verification and reuse of models. Because of the substantial costs associated with software engineering, the economic gains to be made from the portability and reuse of software engineering products are large.

With the evolution of the concept of open systems, consumers no longer expect any one supplier of CASE tools to dominate the marketplace. Communication standards covering both the storage of machine-readable information and its transfer between tools ensure consumer choice.