NISTIR 4904

FILE COPY
DO NOT TAKE

A COMPATIBILITY ANALYSIS OF THE ANSI AND ISO IRDS SERVICES INTERFACES

Jean Bérubé

IDEgenic, Inc.

Alan Goldfine
Editor

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Computer Systems Laboratory
Gaithersburg, MD 20899

NIST

NISTIR 4904

A COMPATIBILITY ANALYSIS OF THE ANSI AND ISO IRDS SERVICES INTERFACES

Jean Bérubé

IDEgenic, Inc.

Alan Goldfine
Editor

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Computer Systems Laboratory
Gaithersburg, MD 20899

September 1992

U.S. DEPARTMENT OF COMMERCE
Barbara Hackman Franklin, Secretary

TECHNOLOGY ADMINISTRATION
Robert M. White, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

FOREWORD

Alan Goldfine
National Institute of Standards and Technology
Computer Systems Laboratory

A key event in the evolution of data management software occurred in the late 1980s with the development of specifications for the Information Resource Dictionary System (IRDS), a software tool for controlling, describing, protecting, documenting, and facilitating the use of an organization's information resources. These dictionary/repository specifications, developed by Technical Committee X3H4 of Accredited Standards Committee X3, are popularly known as the "ANSI IRDS." The specifications were published as American National Standard X3.138-1988 in October 1988 and subsequently adopted as Federal Information Processing Standard (FIPS) Publication 156 for the Federal Government.

The ANSI IRDS specifies two user interfaces: a command language and a panel interface. X3H4 recognized at the time of development that specifications for a software interface to the IRDS were also required by users and vendors, and began work on what became known as the IRDS Services Interface. X3H4 hoped that the IRDS and its Services Interface would become an international standard, and to this end helped establish an IRDS Rapporteur Group within Working Group 3, Sub Committee 21 of Joint Technical Committee 1 of the International Organization for Standardization/International Electrotechnical Committee (ISO). However, the IRDS specifications that emerged from the international group were inconsistent with those of the ANSI IRDS, so X3H4 decided to proceed with the development of its own services interface that would maintain strict consistency with the X3.138-1988 standard.

The X3H4 IRDS Services Interface was adopted as American National Standard X3.185-1992 in February 1992. ISO adopted its services interface as IS 10728 in May 1992. All this, of course, raises questions of compatibility and makes for a great deal of confusion.

This report attempts to clarify the situation by providing a formal comparison of the functionality and underlying data models specified by the two services interfaces. The focus is on the level of harmony and degree of interoperability that would be found between an ANSI IRDS environment and an ISO IRDS environment.

The report also provides an overview of three other published specifications that include dictionary/repository components: IBM's Repository Manager Interface, DEC's A Tool Integration System (ATIS) proposal, and the Portable Common Tool Environment (PCTE) proposal developed by the European Computer Manufacturers Association (ECMA).

Note: Identification of specific commercial products in this report does not imply recommendation or endorsement by the National Institute of Standards and Technology.

TABLE OF CONTENTS

ABSTRACT

Standardization bodies have produced specifications for two Information Resource Dictionary System (IRDS) services interfaces: the "ANSI IRDS Services Interface" developed by Technical Committee X3H4 of Accredited Standards Committee X3, and the "ISO IRDS Services Interface" developed by ISO/IEC JTC1/SC21/WG3. This report provides a formal comparison of the functionality and underlying data models specified by the two services interfaces. The report's focus is on the level of compatibility and degree of interoperability that would be found between an ANSI environment and an ISO environment.

The report also provides an overview of three other published specifications that include dictionary/repository components: IBM's Repository Manager Interface, DEC's A Tool Integration System (ATIS) proposal, and the Portable Common Tool Environment (PCTE) proposal developed by the European Computer Manufacturers Association (ECMA).

1. INTRODUCTION

1.1 Background and Rationale

Standardization bodies have produced four repository/dictionary standards: the ANSI/FIPS Information Resource Dictionary System, the ISO IRDS Framework, and the ANSI and ISO IRDS Services Interfaces. In addition, manufacturers (e.g., IBM and DEC) have proposed proprietary products with published interfaces, and the European project PCTE has produced a standard containing a similar interface.

This has created some level of confusion, and it is currently difficult to isolate the political, marketing, and technical issues resulting from that diversity. This represents a problem for suppliers whose products are not limited to a national boundary, and who must coexist with other products to satisfy their users' requirements. Furthermore, government, industry, and other users of a repository/dictionary are faced with an apparent diversity of directions and products. This is a double edged sword, since an investment made too early in one direction can perhaps be partly lost, but a delayed investment postpones the benefits of such products in an open environment.

1.2 Scope of this Document

The primary purpose of this report is to compare:

- the ANSI IRDS standards [ANSI 1992], [IRDS 1988]
- the ISO IRDS standards [FRAMEWORK 1990], [ISO 1992].

The following are also part of the context of this report and will be surveyed:

- the IBM Repository Manager Interface [RM 1990]
- the ATIS IRDS proposal [ATIS 1990]
- the PCTE standard [PCTE 1988].

1.3 Objectives

The principal objective of this report is to establish the level of compatibility and interoperability possible between an ANSI IRDS environment and an ISO IRDS environment. This is done by attempting to establish if these environments can be reconciled:

- in definition. That is, can they share data?
- in operation. That is, could a single product offer the two types of services?

Chapter 3 refines these compatibility elements.

This is a preliminary study, not a detailed analysis, and the intent is to establish where the inconsistencies between the two sets of standards make a difference. The study provides a statement of the problems facing suppliers, users, and standardization bodies, and facilitates the development of solutions based on an objective statement of compatibility between the standards.

1.4 Assumptions

1.4.1 Assumptions about the IRDS Environment

An initial hypothesis is that the different specifications all share the same general requirements and the same universe of discourse (UofD). This hypothesis has not been previously recognized because requirements and universes of discourse were never formally defined by any of the standards. When definitions were attempted, the use of different conceptual modeling techniques in the different specifications overshadowed the similarities.

Another initial hypothesis is that differences in presentation, detail requirements, approach, and data modeling facilities make the standards look much more different than they really are. The proposals also appear to be different because they are positioned differently in the continuum of interfaces between the database interface and the user interface. Identifying these surface differences eliminates useless confusion and thus enables discussion of the real differences.

1.4.2 Assumptions about the IRDS Standard

The following definition is assumed for an IRDS standard:

> An IRDS standard standardizes those parts of a specialized application used to manage a database about information, information systems, and information resources.

Therefore, the following apply:

- Since the IRDS is about an application, approaches and techniques used for other business applications can be used.
- Since the IRDS is a data management application, the basic principles applicable to data management, as outlined in the current draft of the Reference Model of Data Management [RMDM 1990], also apply.
- Since the application is specialized, the items that make it specialized should be well defined.
- Since only parts of the application are standardized, the rationale for selecting those parts should be spelled out.

2. SUMMARY

2.1 Approach

To compare two apparently dissimilar things, some reference base must be established. To achieve that purpose, this report, by applying techniques used in application development, produces a form of conceptual schema. No formal modeling is done, although normalization (data) and cohesion (process) are sought.

The report generally uses tables to compare the standards to this reference base, and then to compare one standard against another. This identifies and highlights areas of compatibility and incompatibility.

2.2 Compatibility Assessment

Both the X3H4 and ISO committees began with the same initial set of concepts, or conceptual schema, concerning the area of the world an IRDS keeps data about. Most concepts remained compatible during the evolution of the two IRDSs. However, two operational facilities, security and version control, are now incompatible. Security in the ISO IRDS is applied at the attribute, or column, level, while the ANSI IRDS applies security at the entity, or table, level. As for version control, the difference in requirements, concepts, and implementation mechanisms makes the two approaches incompatible.

In terms of overall IRDS architecture, the ANSI IRDS was initially intended to facilitate the portability of human skills across environments by ensuring that different products exhibited common semantics and behavior. The ISO IRDS is targeted at ensuring the portability of tools across environments. This difference in perspective became a source of incompatibility because it influenced detailed design decisions.

Defining compatible IRDSs, and sharing data between them, is possible. In terms of the three-schema architecture [CONCEPTUAL SCHEMA 1987], it would be possible to design a single conceptual schema for both an ANSI and an ISO IRDS. The logical, or external schemas would be different in the two environments, and some parts of the conceptual schema could not be implemented. Other facilities, such as the ANSI IRDS ability to maintain non-normalized data and the ISO IRDS enforcement of constraints, would be lost, because each of these facilities is not supported in the other environment. Naturally, designing such an IRDS would be a non-trivial exercise.

As a consequence, while it would be possible in principle to exchange data between the environments. it could only be for a one-time transfer. Transferring data from one type of IRDS to the other, maintaining the data, and then returning it to the original IRDS, would not be feasible. This is because loss of integrity could happen in the less constrained environment (the ANSI IRDS), or some ANSI IRDS constructs could not be maintained in an ISO IRDS, even if simulated.

As for the possibility of having products offer both interfaces, either in parallel or layered, there are enough factors of incompatibility to prevent the behavior of each interface to be completely conformant with its respective standard.

For version control, security, record retrieval, and constraint definition, taking the lowest common denominator approach might be feasible, but again it would make each interface non-compliant to its respective standard. These differences are summarized below.

Version control in the strict sense is addressing the traceability of change by recording the successive states, or versions, of a component.

If the component is primitive, change of state is achieved by addition, deletion, and changes of attributes.

If the component is composite, or aggregate, it is made up of a root component, member components, and the associations that tie these in the composite. Any change to these components changes the state of the composite.

Finally there are clusters of components that need to be controlled and manipulated for specific purposes. These are called configurations, and although their nature is similar to composite components, their composition is somewhat arbitrary, and driven by the manipulation or control that is desired.

The ANSI IRDS equates the three aspects, simplifies the problem to a hierarchy, and adds a version identifier to be part of the key of everything. The ISO IRDS deals with the issue of versions of primitive components (object-versions) and the issue of configurations (working sets), but ties both together. In practice, the two approaches are incompatible, and neither is likely to be the final solution.

In the area of security, access rights are given to the user at the entity (or table) level in the ANSI IRDS, and to the column (or attribute) level in the ISO IRDS. Incompatibility comes from the fact that such detailed access rights in the ISO IRDS could not be defined or even simulated in the ANSI IRDS.

The record naming and selection facilities are also different in the two IRDSs. The ANSI IRDS Services Interface templates allow retrieval of many records at a time and navigation within the given retrieval tree (using the extended services). The same is not available for the ISO IRDS. However, for selecting individual records, the SELECT operator for retrieval in the ISO IRDS is much more flexible than wildcards in the ANSI IRDS. This is one of the important differences that would make interoperability impossible, unless restrictions are put on the retrieval mechanisms.

As for the definition and enforcement of constraints, an ISO IRDS might have update and delete actions (referential integrity) defined in its tables. Also, using the CHECK and the ASSERTION mechanisms, an ISO IRDS could be designed that implements, in the data modeling facility, a larger number of the conceptual schema constraints. There is no equivalent definition and enforcement mechanism in ANSI IRDS.

2.3 **Short Term Solutions**

There do not seem to be any short term solutions to the problems raised at the beginning of this section.

In terms of products, the market is currently in expansion, with CASE tool manufacturers having, or claiming to have, their own repository, and many promising evolution toward the three products/ standards (PCTE, IBM RM, and eventually CDD-ATIS from DEC) that they perceive will probably share the market.

In terms of the possibility of rallying the standardization effort around current projects, this may also not be possible. PCTE is a project of the European Computer Manufacturers Association (ECMA) that will probably go through ISO as a Fastrack standard; the ANSI IRDS is a U.S. standard; and the ISO IRDS is an International Standard. The three sponsoring groups are largely disjoint and have been, to some extent, competitive.

It is not likely that the community will rally around one standard. However, interoperability could be facilitated by the development and use of standard content modules and the addition, to any of the standards, of the major features of the others.

The major problem is that this is in the interest of the users, and not necessarily the priority of suppliers and standards groups.

2.4 **General Conclusion**

The best approach would be to consider what has happened in the last ten years, and will likely happen in the next five, as the normal progression towards maturity in this area. The concept and usage of a repository has evolved in the last ten years. Until we have implemented and used enough of these products, discussion is academic, since concepts and implementations have not yet reached maturity.

This is not to say that nothing can be done, and that we have to wait until 1995 to discuss integration of the standards. To the contrary, if we want to be ready in time, there are things that need to be done today.

2.4.1 **The IRDS Framework**

The ISO IRDS Framework document [FRAMEWORK 1990] is generally accepted in the world of IRDS and repositories. Since it is the only document to have such a consensus, it should be a base for future work. However, the reasons why there is consensus are also the reasons why, in one sense, it has failed. It is not precise enough, does not specify critical concepts such as version control and access control, does not offer a common data modeling facility, and does not unambiguously layer and characterize IRDS interfaces. It is at this framework level that the difference in requirements for a standard to ensure portability of human skills across tools, and for a standard to ensure portability of tools across environments, should be made explicit.

As the current projects reach stability, and implementations appear, one of the first so-called IRDS2 task should be to produce a revised framework, applicable to all current standards and products.

This revised framework could also benefit from work done in other groups, such as the ECMA Reference Model for Tools, the ISO Reference Model for Software Engineering, Electronic Data Interchange (EDI), Common Data Interchange Facility (CDIF), and Conceptual Schema standardization.

2.4.2 **Standards for Communication and Portability of Human Skills**

The initial objectives of the ANSI IRDS project, that is to prescribe a generic content and behavior of an IRDS so that humans would be comfortable using different products and understanding one another, is still valid. In fact, not only is it valid, but it is now more important than before.

It is now obvious that we could have as many as five IRDS Services Interfaces (IBM RM, PCTE, CDD-ATIS, ANSI IRDS and ISO IRDS), with some suppliers also implementing directly over a relational database. A common, more conceptual interface would enable communication and interoperability in such a situation.

To facilitate interoperability, each interface should offer a view facility:

- to enable isolation between its conceptual schema and the external view that may be required

- to enable separation between its internal schema and the external view that may be required.

2.4.3 **Conceptual Schema**

In the IRDS, as in any application of a DBMS, the basic requirement for communication is that communicators share the same conceptual schema, or have a standardized way of exchanging and mapping conceptual schemas. In the terminology of the Data Management Reference Model, this would be a conceptual data modeling facility (DMF).

As such, a conceptual DMF would also serve as a base to define the interface of a user-oriented IRDS, and standardization projects in this area should be started as soon as the above prerequisites are met.

2.4.4 **Functional/Multipart Standards**

To ensure communication, both container and content need to be standardized. This creates the need for sets of standards.

Instead of the current vertical approach we should have an approach by topics. For example, to ensure communication among data modelers, the data modeling tool, and the repository, the following standards are needed:

- A conceptual schema of data modeling
- Diagramming conventions (container and content)
- Diagramming/representation export/import
- Language conventions (container and content)
- The IRDS "front" interface
- The corresponding export/import
- The IRDS tool interface
- The corresponding export/import
- ...

These could be developed as sets of related standards, multipart standards, or standardized profiles.

3. DEFINITION OF LEVELS OF COMPATIBILITY

3.1 Introduction

This section defines the comparison levels and conventions for identifying compatibility of IRDSs. Levels of compatibility will be classified by the amount of work, if any, required for interoperability, and where actions can be taken.

3.2 Compatibility in Definition

Compatibility in definition is defined as follows:

> IRDSs are compatible in definition if they convey the same basic concepts, perhaps expressed differently. In other words, they share the same universe of discourse and conceptual schema. Compatible IRDSs may be mapped against one another once their conceptual schemas are expressed in the same way.

For discussion within this document, the assumption is often made that the problem being discussed is the implementation of one conceptual schema in two IRDSs.

3.3 Compatibility in Operation

Compatibility in operation, or interoperability, is defined as follows:

> IRDSs are compatible in operation if they have some compatibility in definition, and the differences in implemented interfaces and available services can be attributed mainly to syntactic differences and data modeling facilities (when equivalent).

3.3.1 Levels of Interoperability

1) No interoperability

   This is the situation when the conceptual schemas of two IRDSs have no common element.

2) User interoperability

   From the same conceptual schema two IRD definitions are prepared.

   Two tools are used, one based on an ISO IRDS, the other based on an ANSI IRDS. Each IRD is defined using the definition services of its IRDS, and the user can understand and conciliate input/output of the dictionary operations in a useful manner.

3) Data exchange

From the same conceptual schema two IRD definitions are prepared and loaded. Data can be exchanged through export/import facilities between the ANSI IRDS and the ISO IRDS.

4) Transparency of operations

From the same conceptual schema two IRD definitions are prepared and loaded.

An ANSI IRDS compatible tool can access the ISO IRD.

An ISO IRDS compatible tool can access the ANSI IRD.

A single tool can access both the ISO and the ANSI IRD.

5) Transparency of definitions

From a conceptual schema one IRD is prepared and loaded.

An ANSI IRD definition can be created from an ISO IRD definition.

An ISO IRD definition can be created from an ANSI IRD definition.

An ANSI IRDS compatible tool can access the ISO IRD definition.

An ISO IRDS compatible tool can access the ANSI IRD definition.

A single tool can produce both the ISO and the ANSI IRD definition.

4. DEFINITION OF REFERENCE BASE

4.1 Introduction

This section defines the underlying grid on which each standard will be mapped. A reference base is needed in the following areas:

- the context of the IRDS, i.e., the part of the real world the IRDS applies to
- the architecture of the IRDS, i.e., the major components and interfaces of the IRDS
- the various data modeling facilities applicable to an IRDS
- the definition of the IRDS, i.e., the categories of data managed by the IRDS
- the operation of an IRDS, i.e., the various services and facilities provided by the IRDS.

This section introduces these reference points at a coarse level of granularity. When discussion of a specific topic requires a refinement of the reference base, the details are introduced in the relevant section.

4.2 IRDS Context

4.2.1 Definition of an IRDS

An IRDS standard standardizes those parts of a specialized application used to manage a database about information, information systems, and information resources.

Since an IRDS standard is about an application, approaches and techniques used for other business applications can be utilized.

4.2.2 Context (Universe of Discourse) of an IRDS

As introduced earlier, an IRDS manages a database about information systems and information resources. The Universe of Discourse (UofD) of an IRDS is then the world of developing, implementing, and operating information systems (as defined in [FRAMEWORK 1990]). This subset of the real world is identified in this document as Information Systems Engineering (ISE), as defined in [BÉRUBÉ, J. 1990a].

No formal conceptual definition of the UofD is proposed or required. To avoid discussions on conceptual schema, the UofD is introduced here in the form of a simple diagram (Figure 1) and natural language sentences. The reader will recognize the overall diagram to be of the E-R family, as introduced in [CONCEPTUAL SCHEMA 1987].

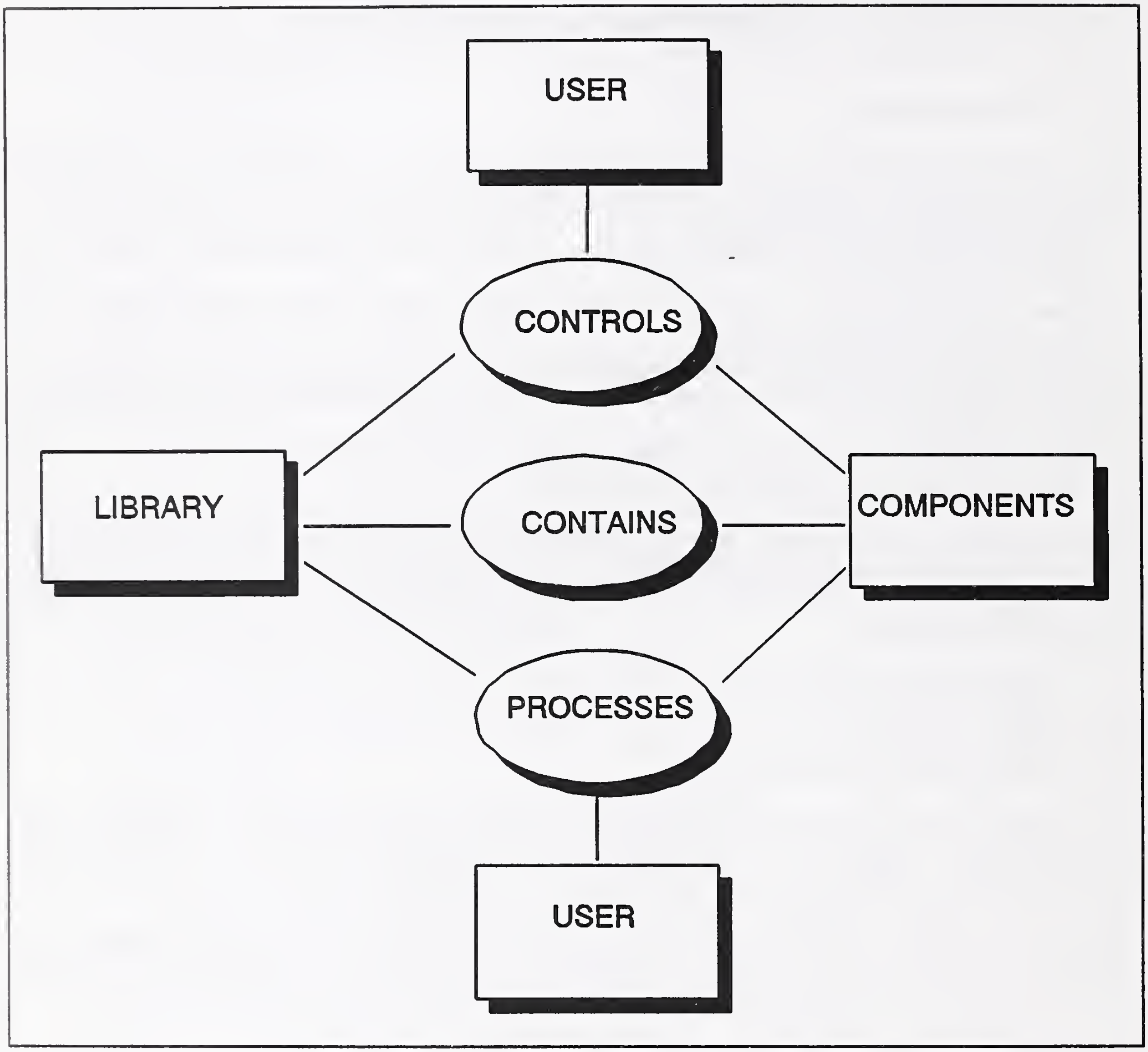

Figure 1. The IRDS Universe of Discourse

4.2.3 **Description of the IRDS UofD**

The various objects in Figure 1 are introduced at a high level, That is, they are grouped into sets called object groups. The word "object" is used with the same meaning as "entity" is in [CONCEPTUAL SCHEMA 1987]: objects (entities) are real or abstract facts of the UofD. They are not constructs of a modeling technique.

The various objects of interest to an IRDS can be grouped and described as follows:

- COMPONENTS Object Group

  This is the characteristic part of the IRDS. Components are related to information systems engineering. Components are grouped in models (sets of components, associations, and data elements). Models can be represented using different formalism (diagrams). These objects are specific to ISE.

- LIBRARY Object Group

  This set of concepts encompasses the aspects related to container and media. A library is made of sublibraries, themselves made of other sublibraries, or of members. These concepts can be classified using different schemes, such as:

  — Nature (Archival, Reference, Production,...)
  — Media/Technology (Host, Local Server, Workstation,...)
  — Scope (Enterprise-Wide, Unit, Individual,...)
  — States (Work-in-Process, Staging, Locked,...)

  Partitions, stages, versions, etc., can be modeled as virtual or real libraries.

- COMPONENTS STORED IN LIBRARY Object Group

  Assuming real or virtual sublibraries to model states and versions, this set of concepts associates COMPONENTS with these states and versions.

  It also associates COMPONENTS with their location and technology, thus describing the distribution aspects of an IRDS.

- USER Object Group

  This set of objects includes the various types of user that either execute production activities, or perform some control tasks. It also includes the various organizations, technologies, and facilities where these activities occur. Types of IRDS users are introduced later in this section.

- USER PROCESSES COMPONENTS IN LIBRARY Object Group

  This set of concepts describes the user activities on components, such as creates, updates, and deletes on components, and copy and merge on libraries.

  These are production operations, the user here performing information systems engineering activities that require or have impact on an IRDS.

• <u>USER CONTROLS COMPONENTS IN LIBRARY Object Group</u>

This set of concepts describes administration activities, such as the administrator setting up access rights and other controls. Some of these entities are associations between COMPONENT and USER, such as the associations to define access rights themselves.

4.3 IRDS Architecture

4.3.1 Identification of DMFs and Interfaces

Figure 2 identifies the data modeling facilities (DMFs) and interfaces relevant to this report. This diagram is an adaptation of Figure 3.4 in [CONCEPTUAL SCHEMA 1987] and Figure A.2 in [RMDM 1990].

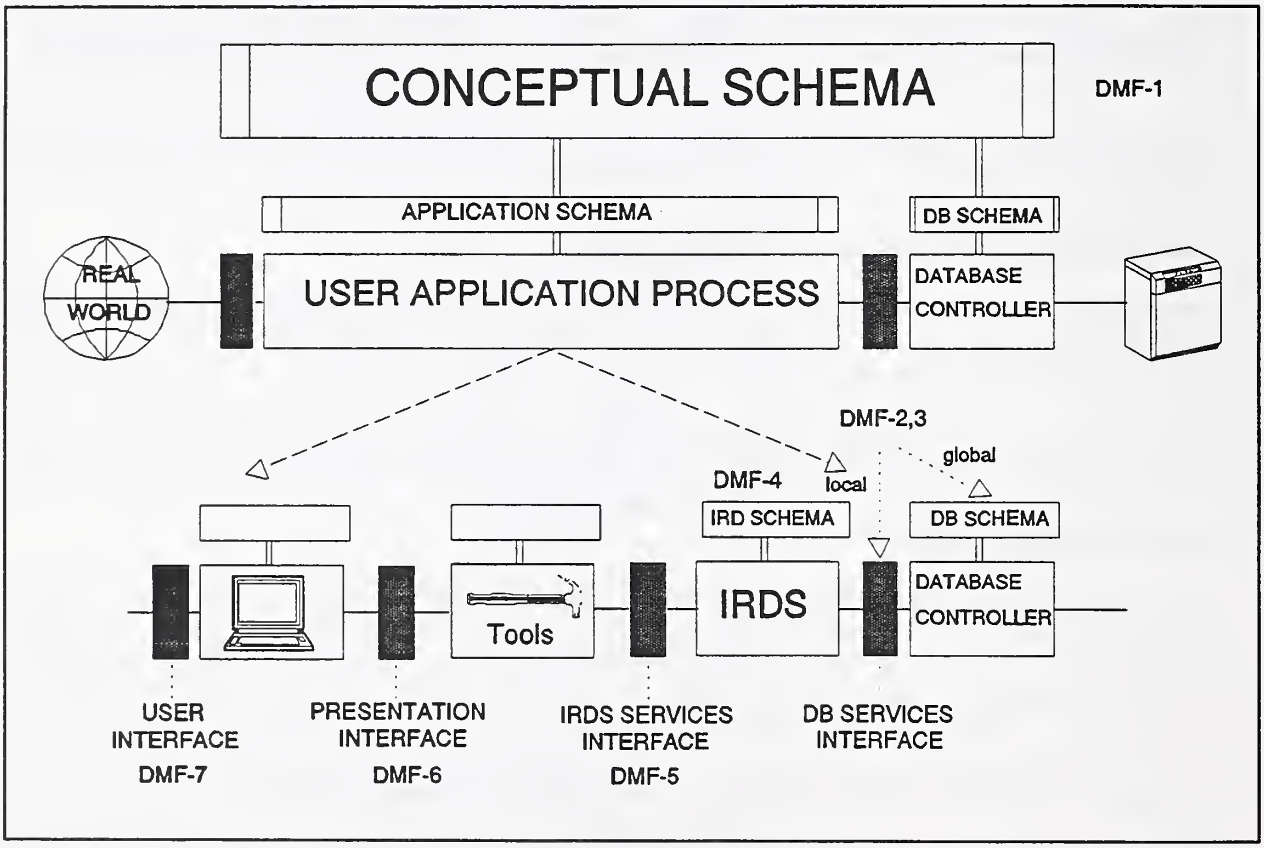

Figure 2. Schemas, Interfaces, and DMFs

Of the data modeling facilities portrayed in Figure 2, only the following are of interest to this report:

- the conceptual schema data modeling facility (DMF-1)
- the local and global logical (external) database management data modeling facilities (DMF-2, DMF-3)
- the IRDS global data modeling facility (DMF-4)
- the IRDS local data modeling facility (DMF-5).

4.3.2 Components of an IRDS

One of the major design decisions taken for both IRDSs was to make some of the definition of the dictionary data user controlled, or user extensible, and not predefined in the IRDS by the supplier. This means that while some definition data will be accessible only by the supplier, using the proper installation procedures, other definition data will be accessible via a set of IRDS services. This creates two levels of similar IRDS services, operating on the dictionary definition and the dictionary content.

4.3.2.1 Level Independent Data

Although both IRDSs introduce distinct data levels, defined the same but using different terminology, further analysis has revealed that not all processes and data can be, or need to be, classified in such a way. It is only when two object groups can be associated with the "is the definition of" association that the level concept applies. All other associations are level independent. This is a shift of perspective, as the initial definition of both IRDSs classified all IRDS data in levels, and resulting discussions tried to prove that some of that data did not fit in this level model. The approach taken here is that the level model applies to a subset of IRDS related information. Non level related data is called state/context data.

4.3.2.2 Level Independent Services

As established above, the issue is not what is level independent, but what is level dependent. Only services applicable to the object groups that can be associated with the "is the definition of" association can be classified using the level model.

4.3.2.3 IRDS User Roles

The various types of users of an IRDS are identified in Figure 3.

- <u>IRDS Suppliers</u>

  Suppliers of the IRDS implement the relevant standards in their products. They also predefine the definition level for "context/state data" data definition, and "dictionary definition data" data definition. They will also put in these definition

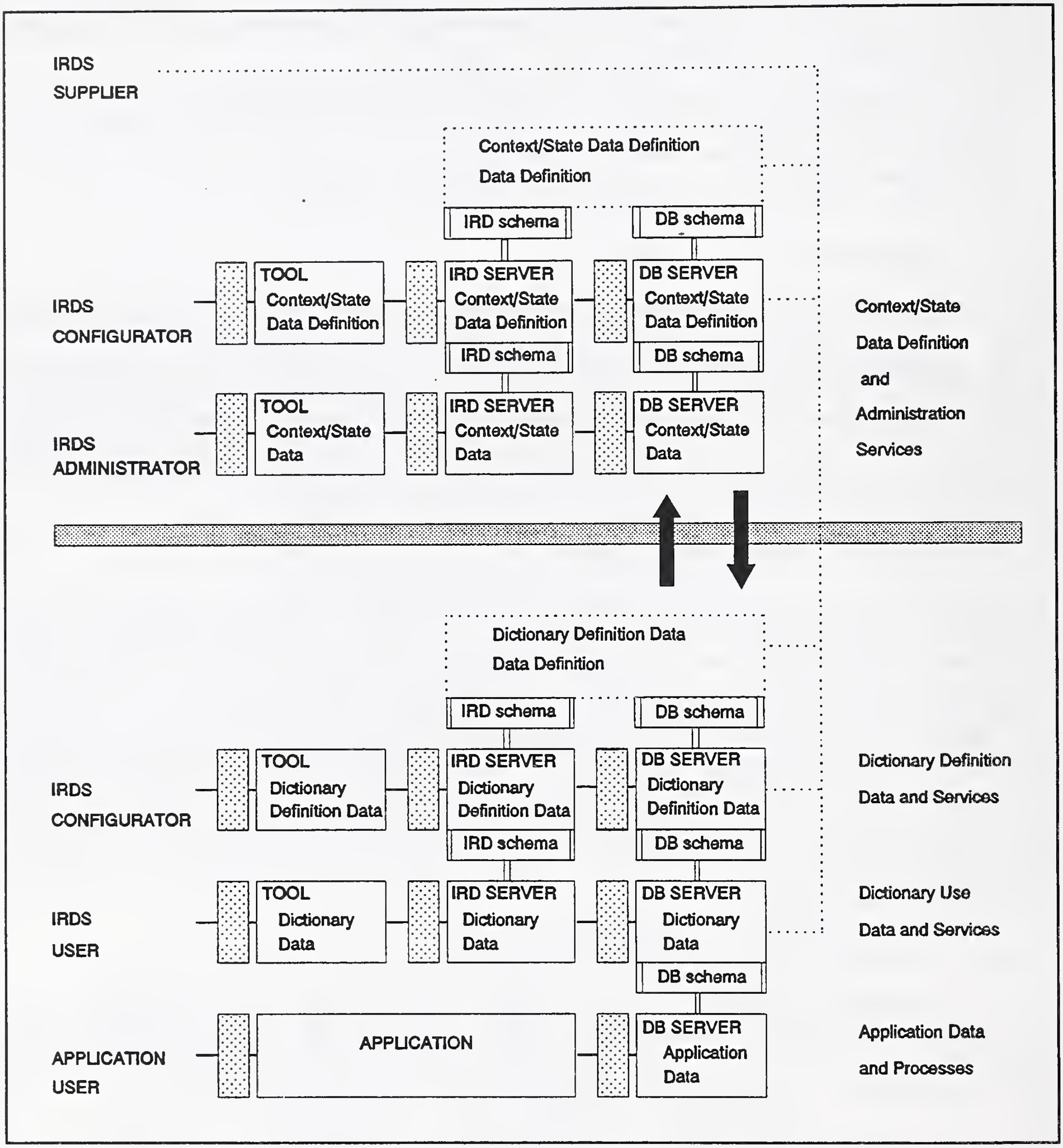

Figure 3. Components of an IRDS Environment

level implementation dependent values. They may (ANSI IRDS) or may not (ISO IRDS) provide initial values for dictionary data.

* IRDS Configurators

  IRDS configurators install the IRDS product in their environments. They use the tailoring and definition facilities to define the schema for context/state data (for the part that is configurable) and to define the schema for the dictionary. They will also set implementation dependent values that are configurable.

* IRDS Administrators

  IRDS administrators control the IRDS as a container, giving access rights and controlling relevant context/state data, such as version control data.

* IRDS Users

  IRDS users populate and use the dictionary, either directly via dictionary tools, or indirectly via other tools, such as CASE tools, library management tools, compilers, editors, etc.

4.3.2.4 **IRDS Data**

Since the data in an IRDS should be structured so as to reflect the structure of objects in the real world, the IRDS data is organized, for the purpose of this document, in two categories. They were identified in the previous section.

* ISE Data

  — COMPONENTS Object Group
  — LIBRARY Object Group
  — COMPONENTS STORED IN LIBRARY Object Group

* Context/State Data

  — USER Object Group
  — USER PROCESSES COMPONENTS IN LIBRARY Object Group
  — USER CONTROLS COMPONENTS IN LIBRARY Object Group

4.3.2.5 **IRDS Services**

If services were grouped according to the structure of the data they operate on, we would have four categories of services:

* Context/State Data Definition Services

* Context/State Data Operation Services (Administration)

* Dictionary Data Definition Services

* Dictionary Data Use Services

As the two standards generally group the first three in one level called definition, only two groups of services will be used in this report.

Figure 3 positions the IRDS components in relation to each other and to IRDS users.

4.4 IRDS Data Modeling Facilities

4.4.1 General

4.4.1.1 Definition

A collection of persistent data is defined by a schema which contains a particular set of data structuring rules. Such a set of data structuring rules, along with the associated set of data manipulation rules, is referred to as a data modeling facility [RMDM 1990].

In this document, the term Data Modeling Facility (DMF) is generally used to mean the set of data structuring rules for a collection of persistent data or a collection of transient data crossing an interface. The set of data manipulation rules is discussed, when necessary, as part of the discussion on services and interfaces.

4.4.1.2 Elements of a Data Modeling Facility

A data modeling facility is made of components, associations between these components, and rules governing the existence of components and associations. This section describes the selected elements, and establishes the conventions used in this document. All the components of a data modeling facility can be derived by successive aggregation and/or classification of the basic component values. Relevant associations between these components are then established. Finally, rules governing these associations are identified.

4.4.1.3 Derivation of Components of a DMF

Table 1 defines the major components of the reference DMF, by successive derivations (classification, aggregation) from the base concept of value.

4.4.2 Conceptual Schema Data Modeling Facility

Neither of the IRDSs has produced a formal conceptual schema using a modeling facility, so there is no basis upon which to compare conceptual data modeling facilities. However, the absence of such a model may have been the reason for some of the incompatibilities detected between the two IRDSs. The absence of a shared conceptual

BASE COMPONENT	CLASSIFICATION	AGGREGATION	DERIVED COMPONENT	DEFINITION
value				primitive specific representation of a property of an individual object example: 70
value	classification of values		**value type**	class of representable values example: numeric
value		aggregation of values	**domain**	set of permissible values example: (0...130) for age
value		aggregation of values	**instance**	set of values for properties of an individual object example: (brown, blue, 2m, 80k, 20.11.23, 70)
value	classification of values		**data element**	class of values for a property of an object, or property type example: weight
data element		aggregation of data elements	**data group**	set of data elements referred to collectively example: date (year, month,day)

Table 1 (1 of 2)

schema has created and is creating difficulty in the process of creating IRDS standards. The current IRDS Framework is not robust enough to be considered a conceptual schema. [CONCEPTUAL SCHEMA 1987] has clearly stated that in the absence of a shared conceptual schema, humans will have difficulty in communicating. This assertion has been somewhat illustrated in many IRDS group meetings.

This section introduces only the base reference concepts used later in the document.

4.4.3 **Data Management Modeling Facility**

The data used at the interface of each IRDS service and its required database services can be considered the data management local logical data modeling facility. The consolidation of these is a global schema. It is important to note that it is the global schema of the data at the database services interface, as seen by the IRDS services, *not the global schema of the data available at the IRDS services interface*.

BASE COMPONENT	CLASSIFICATION	AGGREGATION	DERIVED COMPONENT	DEFINITION
data element instance	classification of instances	aggregation of data elements	**record**	set of attributes common to a class of instances example: patient record (hair color, eye color, height, weight, birth date, age)
view record		aggregation of records	**record view**	set of records with subsets of their attributes, showing part of a database example: the patient admission view contains selected data from the patient and history records
sub record	classification of record		**record subtype**	set of records used to classify attributes of a real world object example: the male and female records contain specific attributes of the patient
schema		aggregation of records	**schema**	Set of all the previous components, defining a database example:schema of the medical history database

Table 1 (2 of 2)

4.4.4 **IRDS Data Modeling Facility**

The only prescriptive part of a service standard is the identification of the services to be performed, and the specification of the messages (service request and responses) exchanged with the client. This is the IRDS local logical data modeling facility.

Although this DMF is dependent on the global DMF, it may have special constructs, such as views and templates. It may also use constructs of another DMF; for example, the ISO DMF uses constructs of the PASCAL language.

The IRDS global logical data modeling facility is the consolidation of the local data definitions at the IRDS services interface, as seen by the IRDS Services Interface client.

4.5 IRDS Operations

IRDS operations are characterized by a class of service applied to a class of data. This is the user definition of a transaction — the external view (business transaction) of the operation — versus the view in terms of internal design units. This removes the artificial differences introduced by different internal design decisions, such as overloading of parameters and operators, using the name of the service or a parameter to convey the level, etc.

The following operations classes are used:

- <u>Definition (and administration)</u>
  - Implementation characteristics and definition session control
  - Definition library control
  - Definition transaction control
  - Definition transaction record naming and selection
  - Access control definition
  - Dictionary content definition
  - Naming definition and control
  - Dictionary library definition

- <u>Dictionary</u>
  - Dictionary session control
  - Dictionary library control
  - Dictionary transaction control
  - Dictionary record naming and selection
  - Dictionary content manipulation

5. ANALYSIS OF IRDS CONCEPTS

5.1 Introduction

This chapter compares the "real world of IRDS," the universe of discourse, as perceived by both the ANSI and ISO standards.

5.2 IRDS Context

Since both IRDSs have similar concepts, these concepts can be mapped against the reference concepts in Table 2, without a preliminary analysis of each IRDS.

The major conceptual issue raised here is about version control. There are three basic needs in this area, and they are combined in many different ways by different standards or products.

Version control, in the strict sense, deals with the traceability of change by recording the successive states, or versions, of a component. If the component is primitive, change of state is achieved by addition, deletion, and changes of attributes.

If the component is composite, or aggregate, it is made up of a root component, member components, and the associations that tie these in the composite. A composite component can change state not only by addition, deletion, and changes of attributes of the root component, but also by addition, deletion, and changes of attributes of the member components. Finally, just a rearrangement of the associations between root and members will cause a change of state. These composition graphs are not necessarily trees, and the same component can be a member of many composites.

There are also clusters of components that need to be controlled and manipulated for specific purposes. These are called configurations, and although their nature is similar to composite components, their composition is somewhat arbitrary, and driven by the manipulation or control that is desired.

These three aspects are made further complex by the fact that in a typical environment, changes may occur in parallel by different teams, and some consolidation of state changes needs to be possible.

The ANSI IRDS equates the three aspects, simplifies the problem to a hierarchy, and adds a version identifier to be part of the key of everything.

The ISO IRDS deals with the issue of versions of primitive components (object-versions) and the issue of configurations (working sets), but ties both together.

Reference ELEMENT	DETAILS	ANSI IRDS	ISO IRDS	NOTES
components		entities	dictionary object	will be discussed in the data modeling facility section
library	states	lifecycle phases partitions	ird content status	same concept
components stored in library	states	by a data element	by a data element	same mechanism
library	versions	version/variation	working set	different concepts
components stored in library	versions	by implicit membership (part of the key)	by explicit membership (working set reference)	different mechanisms
library	schema	schema	schema	similar concept
components stored in library	schema	by a relationship in-set-ird-schema	by definition	different mechanism
user		entity IRDS user	user base table	same concept
user processes components ...		sessions and transactions	sessions and transactions	same concept
"		audit attributes	audit attributes	same concept
user controls components	global and entity level security	permissions	working set privilege table privilege column privilege	same concept different level
"	quality indicators	used	not available	

Table 2

In practice, the two approaches are incompatible, and neither approach is likely to be the final, comprehensive solution.

The other area of incompatibility is in the level of granularity of access control, where ISO goes to a much finer resolution by operating at the column level.

5.3 IRDS Architecture

Although both services interfaces are presented as competing at the same level, they do not really have the same architecture. The ANSI IRDS interface is closer to the user, and the ISO IRDS interface is closer to the DBMS.

5.4 Non-IRDS Data Modeling Facility

The ANSI IRDS adopts the hypothesis that the user's work will be simplified by manipulating more abstract concepts, such as entities and relationships. That may be true for one community of users, the community of repository definers and administrators. But the vast majority of users are isolated from the IRDS by possibly two other layers of interfaces, a tool layer, where the units might be components, associations, deliverables, and attributes, and a representation layer, where the units are shapes, lines, columns, intersections, text, etc.

Thus, in practice, most IRDS users will never see the DMF used at a services interface.

5.5 Data Management Data Modeling Facility

The ANSI IRDS is an independent and isolated interface, and makes no assumptions about, nor introduces constraints from, the underlying database modeling facility. Even if some of its constructs (templates) can be traced back to current or proposed implementations, this is not a dependency on a modeling facility.

The ISO IRDS, on the other hand, is dependent on the database modeling facility. This dependency stems from the choices made to use SQL as the DMF and the data specification language.

The ISO IRDS document states quite correctly that even if the IRDS behaves like an SQL machine, this does not preclude a non-SQL implementation. However, implementing it on a non-SQL platform means building SQL interpreter-like functions to parse constraints and other SQL constructs. Many ISO IRDS service calls are, in fact, SQL statements with a different syntax.

Where the SQL DMF dependency is more obvious is in what is not specified in the ISO standard. In general, the IRDS DMF is limited to what is in the current SQL 92 standard. For example, some aggregation mechanisms, such as templates, cannot be offered at the services interface because they are not part of the SQL DMF, and/or could not be specified using SQL. The same applies to some forms of constraint. However, in one instance, subtables (for classification and subtyping) have been introduced into the ISO IRDS, but are not in SQL 92. This introduction is based on the

"SQL3" proposal. The converse will also create synchronization problems. For example, the BIT STRING data type has been added to SQL 92, and one might want to add it to the ISO IRDS for harmonization purposes.

6. ANALYSIS OF IRDS DATA MODELING FACILITIES

6.1 Introduction

As illustrated in section 4.3.1, the data modeling facility for the IRDS could be selected in various ways:

- It could be similar to a DMF used for database services. This was the choice made in the ISO IRDS.
- It could be similar to a DMF used for a conceptual schema. This was the choice made in the ANSI IRDS.
- It could be anything else. For example, many CASE tools have their own DMF, closer to the user way of structuring and viewing data (e.g. models, components, associations, diagrams, graphic objects).

Since a conceptual DMF is, by definition, better able to capture the semantics of the real world, and since the E-R DMF has been proved by usage to be easily understood by real world people, such a choice seen from the client perspective makes the IRDS easier to define and use. However the IRDS then needs to do more work to translate the service requests in a database services DMF.

The ANSI and ISO IRDSs each use the same respective DMF at both levels, so the major conclusions of this section apply to both the IRD definition level and the IRD.

Each DMF is mapped against the reference DMF introduced in Chapter 4. Conclusions are then reached by comparing these mappings.

6.2 Reference Data Modeling Facility

6.2.1 Reference DMF Elements—Basic Units

Tables 3, 4, 5, and 6 introduce the basic elements of the reference DMF used for the comparison of the DMFs of the ANSI and ISO IRDSs.

Each **reference element** (column 1) is defined in term of a basic **component** (column 2, defined in 4.4.1), or an **association** between basic components (column 2), or a **rule** on these association (column 3).

Reference ELEMENT	COMPONENT	ASSOCIATION	RULE	NOTES
value	value			
value type	value-type			
domain	domain			
domain **includes** values		association domain - value		
instance	instance			
data element	data element			
data element **has value**		association data element - value		
multiplicity of has value			rule on has value association	multiplicity 0,1,n
data element **has domain**		association data element - domain		
allowed value			rule on has value association	control on allowable values other than explicit domain
data element **dependent on** data element		association data element- data element		dependency derivation
allowed dependencies			rule on dependent on association	
data group	data group			
data group **includes** data element		association data group- data element		includes
data group **has value**		association data group - value		has value
multiplicity of has value			rule on has value association	multiplicity 0,1,n

Table 3

6.2.2 **Reference DMF Elements—Records**

Reference ELEMENT	COMPONENT	ASSOCIATION	RULE	NOTES
record	record			
record **includes** data element data group		association data element-record data group-record		
normalization **(first form)**			rule on include association	-no multivalued data elements -no grouped data elements
normalization **(second form) (third form)**			rule on include association	-no dependent/derived data elements -no null valued (non applicable) data elements -all included data elements dependent on key data elements
uniqueness			rule on include association	one value of data element for each instance of record
data element data group **identifies** record		association data element-record data group-record		key
multiplicity of identifies			rule on identifies association	
uniqueness of identifies			rule on identifies association	one value of data element for each instance of record

Table 4

6.2.3 **Reference DMF Elements—Record References and Constraints**

A reference is established when, from the content of one record, it is possible to know the identifier of another record. This is a generalization of the mechanism found in all data modeling facilities. Figure 4 gives an illustration of references in the relational and the E-R model. Cardinalities, integrity rules, referential constraints, and other constraints are expressed based on the existence of a reference and the existence, or non-existence, of instances of the referenced record.

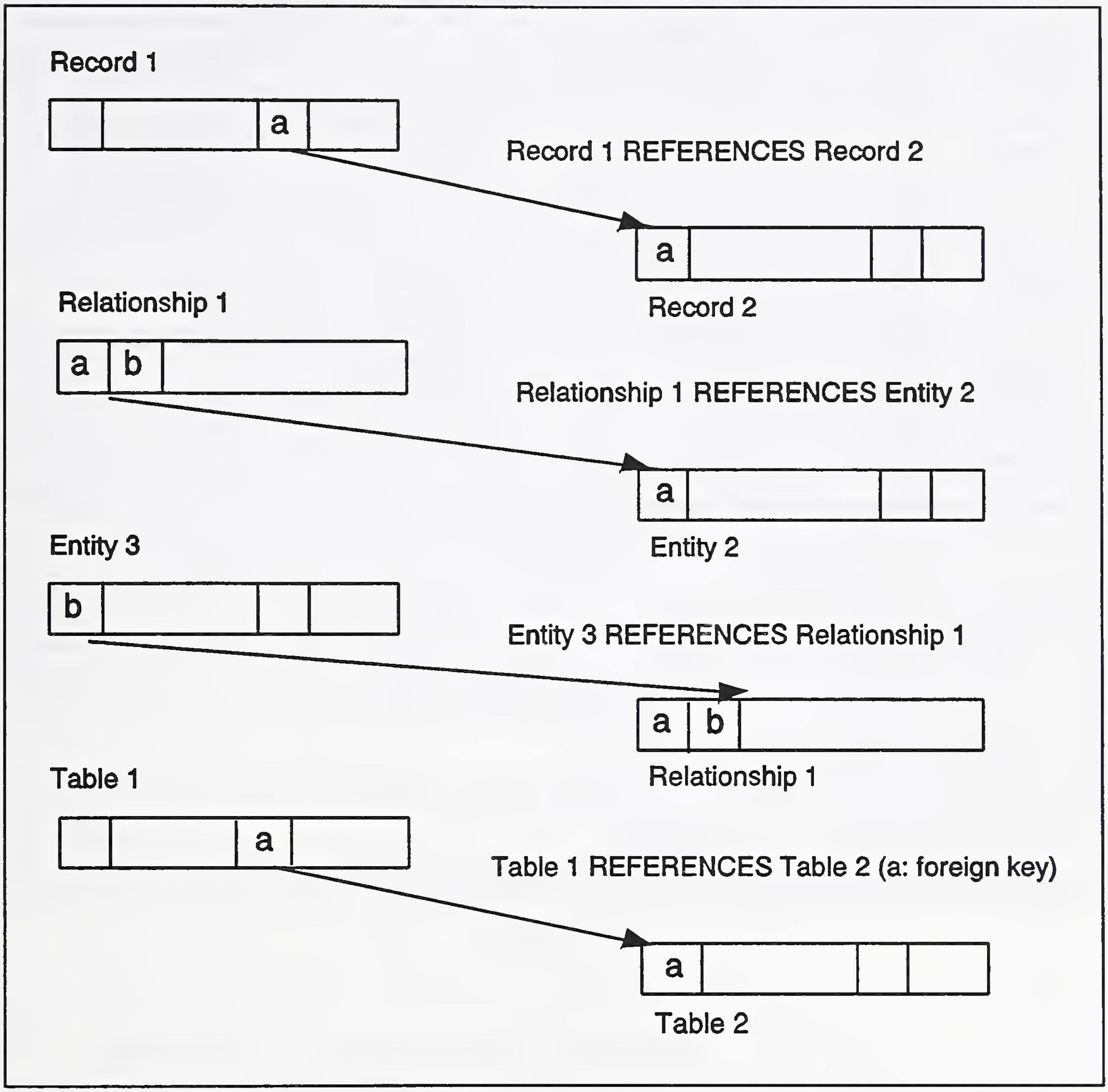

Figure 4. References Between Records

Reference ELEMENT	COMPONENT	ASSOCIATION	RULE	NOTES
record **references** record		association record - record		data elements in a referencing record identify referenced record
referential **cardinality**			rule on reference association	how many instances of the referenced record for one instance of the referencing record (0,1,n)
referential **integrity**			rule on instance and reference association	existence of unreferenced records
referential **constraints**			rule on allowable cardinality	
general **constraints**				existence of record based on multiple condition checking

Table 5

6.2.4 Reference DMF Elements—Sets/Subsets of Records

Reference ELEMENT	COMPONENT	ASSOCIATION	RULE	NOTES
composite	composite			
composite **made of** record composite		association composite- record		
multiplicity of made of			rule on composition association	record part of one or multiple composition scheme.
subtype	subtype record			
subtype record **subtype of** record		association subtype record-record		data elements in a record apply to a subclass of instances described by another record
multiplicity of subtype of			rule on subtyping association	record part of one or multiple classification scheme
view	view record			
view record **view of** record view record		association record-view record view record-view record		-data elements in a record are a subset of data elements in another record. -records that are not view of other records are called base records.
multiplicity of base records			rule on view of association	view record is made of data elements from one or many other records

Table 6

6.3 Comparison of the ANSI and ISO IRDSs

The ANSI IRDS data modeling facility is a variation of the E-R model as initially proposed by P. Chen [CHEN, P. 1977]. However, E-R models were proposed to be used as the data modeling facility to prepare conceptual schemas [CHEN, P. 1977, p. 9]. The ANSI IRDS, and the other IRDS/repository standards and proposals discussed in this report, are not at the conceptual, but at the external (or logical) schema level, closer to database design. Using a technique intended for conceptual modeling at the logical level has caused some confusion of objectives between semantic content and operational requirements.

The variation selected by the ANSI IRDS restricts relationships to binary relationships (presumably because they are simpler to implement), and has no mechanism to define constraints. This is somewhat counterproductive in term of the semantic content objective stated above, and the E-R variation described and rated in [CONCEPTUAL SCHEMA 1987] is richer in that respect.

The ISO IRDS data modeling facility is defined as the one implicit in the SQL 92 standard [ISO 1992, 5.1]. It is also defined in the IRD definition tables [ISO 1992, 5.2, 6.2]. For the purpose of this section, the SQL 92 concepts and terms are used, and the IRD definition tables are used for more precision, or when some restrictions to SQL seem to apply.

Table 7 compares basic units, Tables 8, 9, and 10 compare records, Tables 11 and 12 compare references and constraints, and Table 13 compares sets/subsets.

The tables are structured as follows:

Column 1	**Reference Element**, established in section 6.2
Column 2	**ANSI IRDS DMF term** corresponding to the reference element
Column 3	**ANSI IRDS DMF definition**
Column 4	**ANSI IRDS notes**
Column 5	**ISO IRDS DMF term** corresponding to the reference element
Column 6	**ISO IRDS DMF definition**
Column 7	**ISO IRDS notes**

Because there is rarely a one-to-one match, some parts of the tables are repeated to improve clarity.

Items appear on the same line when there is some equivalence. If there is no equivalence, the items appear on separate lines.

6.3.1 ANSI IRDS/ISO IRDS DMF Elements—Basic Units

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
value	attribute	specific value of an attribute-type for a particular entity or relationship		value		implicit definition
value type	attribute-type format	STRING INTEGER REAL TEXT DATE-TIME	REAL is not defined at the meta level	data type	set of representable values CHARACTER BIT NUMBERS DATETIME	BIT is not used by IRDS
domain	attribute-type validation-data/ -procedure	valid value set for one or more attribute-types	VALUE-VALIDATION, RANGE-VALIDATION	DOMAIN	set of permissible values	IRDS defined (example: IRD_KEY)
domain **includes** values						user defined (example: IRD_DOMAIN)
instance	entity, relationship			row, object	non empty sequence of values smallest unit for insert/delete	-definition object -dictionary object (ISO 5.0)

Table 7 (1 of 2)

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
data element	attribute type	type of property for an entity-type or a relationship-type		COLUMN	multi set of values smallest unit of data manipulation	
data element **has value**						
multiplicity of has value	singular plural	0 1 n	multivalued data elements allowed	NOT NULL NULL	0 1	n not allowed, no multivalued data elements
data element **has domain**	USES	attribute-type USES validation -data/-procedure		domain		domain name is associated at column definition
allowed value		attribute-type USES validation -data/-procedure		CHECK (in)		specific allowable values can be checked
data element **dependent on** data element		not available				
allowed dependencies		not available		CHECK ASSERTION	dependency can be enforced via CHECK and ASSERTION	derived columns can be defined as such
data group	attribute-group-type	logical collection of attribute-types				
data group **includes** data element	CONTAINS	attribute-group-type CONTAINS attribute-type				
data group **has value**					not available	
multiplicity of has value	singular plural	0 1 n	multivalued data elements allowed			

Table 7 (2 of 2)

6.3.2 ANSI IRDS /ISO IRDS DMF Elements—Records (Entities)

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
record	ENTITY-TYPE	type of information resource that is to be described		TABLE	multi set of rows	
record **includes** data element data group						
normalization **(first form)**			no—groups or multivalued data element types accepted			yes no groups or multivalued columns
normalization **(second form) (third form)**			not enforced same data elements (domain) can be included in many tables, other than keys			not enforced same column (domain) can be included in many tables, other than keys
uniqueness			uniqueness is not enforced except for ACCESS-NAME (primary key)			columns can be defined as unique

Table 8 (1 of 2)

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
data element data group **identifies** record	ACCESS-NAME	ACCESS-NAME is made of other data elements: ASSIGNED-ACCESS-NAME VERSION-IDENTIFIER (VARIATION-NAME, REVISION-NUMBER)		PRIMARY KEY		
multiplicity of identifies						-SQL allows multiple columns as keys. -IRDS allows 1 (IRD_OBJECT_KEY), with ASN_ACC_NAME as alternate
uniqueness of identifies	ACCESS-NAME	unique by definition		PRIMARY KEY (=UNIQUE)		

Table 8 (2 of 2)

6.3.3 ANSI IRDS/ISO IRDS DMF Elements—Records (Relationships, Unsequenced)

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
record	RELATION-SHIP-TYPE not sequenced	type of association between two types of information resource that is to be described	-directed, with both forward and backward interpretation. -each relationship-type has two implicit instances, one for each direction.	TABLE	multi set of rows	
record **includes** data element data group						
normalization **(first form)**			no. groups or multivalued data element types accepted			yes. no groups or multivalued columns
normalization **(second form) (third form)**			not enforced same data elements (domain) can be included in many tables, other than keys			not enforced same column (domain) can be included in many tables, other than keys
uniqueness			uniqueness is not enforced except for ACCESS-NAME (primary key)			columns can be defined as unique

Table 9 (1 of 2)

Reference ELEMENTS	ANSI IRDS			ISO IRDS		
	DMF TERM	**DMF DEFINITION**	**NOTES**	**DMF TERM**	**DMF DEFINITION**	**NOTES**
data element data group **identifies** record		concatenation of ACCESS-NAME of participating entities (2), within each relationship-type	participating entities are labeled superior and subordinate	PRIMARY KEY		
multiplicity of identifies						-SQL allows multiple columns as keys. -IRDS allows 1 (IRD_OBJECT_KEY), with ASN_ACC_NAME as alternate
uniqueness of identifies	ACCESS-NAME	unique by definition		PRIMARY KEY (=UNIQUE)		

Table 9 (2 of 2)

6.3.4 ANSI IRDS/ISO IRDS DMF Elements—Records (Relationships, Sequenced)

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
record	RELATION-SHIP-TYPE sequenced	type of association between two types of information resource that is to be described	directed, two implicit relationship-types; one for each direction, with different keys	TABLE	multi set of rows	
record **includes** data element data group						
normalization **(first form)**			no; groups or multivalued data element types accepted			yes no groups or multivalued columns
normalization **(second form)** **(third form)**			not enforced; same data elements (domain) can be included in many tables, other than keys			not enforced; same column (domain) can be included in many tables, other than keys
uniqueness			uniqueness is not enforced except for relationship key (primary key)			columns can be defined as unique

Table 10 (1 of 2)

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
data element data group **identifies** record		concatenation of ACCESS-NAME of both participating entities and a sequencing attribute, within one relationship-type	two different key structures for each direction: -superior entity + sequence -subordinate entity + sequence	PRIMARY KEY		
multiplicity of identifies						-SQL allows multiple columns as keys. -IRDS allows 1 (IRD_OBJECT_KEY), with ASN_ACC_NAME as alternate
uniqueness of identifies		unique by definition		PRIMARY KEY (=UNIQUE)		

Table 10 (2 of 2)

6.3.5 ANSI IRDS/ISO IRDS DMF Elements—Record References and Constraints (Relationship to Entity)

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
record **references** record		relationship references entity		FOREIGN KEY		
referential **cardinality**		1 instance of a relationship refers to exactly 1 instance of each participating entity			0 : foreign key in referencing table can be NULL 1 : foreign key in referencing table can be NOT NULL UNIQUE n : other cases	
referential **integrity**				REFERENCE ON DELETE ON UPDATE	SQL integrity	removal of a referencing row can remove the referenced row
referential **constraints**				uni-directional referential	existence of referenced table from a referencing table	uses SQL constructs introduced under the cardinality rule above
referential **constraints**	bidirect-ional referential	existence of referenced entity from referencing relationship 1:1 by definition	always enforced	bi-directional referential	existence of referencing table from referenced table 1:1	-not fully definable using SQL constructs -enforced by service
referential **constraints**				mutually exclusive referential	existence of referenced tables from a referencing table	uses combination of SQL cardinality constructs and CHECK clauses

Table 11 (1 of 2)

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
general constraints				CHECK constraint	requires that a specific search condition not be false for any row in a table	available to IRDS "definer" to define additional constraints
general constraints				ASSERTION definition	integrity constraint that may relate to the content of individual rows of a table, to the entire content, or to a state between tables	

Table 11 (2 of 2)

6.3.6 **ANSI IRDS/ISO IRDS DMF Elements—Record References and Constraints (Entity to Relationship)**

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
record **references** record		entity references relationship		FOREIGN KEY		
referential **cardinality**	cardinality	0 : 1 entity refers to 0 relationships 1 : 1 entity refers to 1 relationship n : 1 entity refers to n relationships			0: foreign key in referencing table can be NULL 1: foreign key in referencing table can be NOT NULL, UNIQUE n: other	
referential **integrity**				REFERENCE ON DELETE ON UPDATE	SQL integrity	removal of a referencing row can remove the referenced row
referential **constraints**	uni-directional referential	existence of referenced relationship from a referencing entity minimum cardinality 0, 1 maximum cardinality 1, n		uni-directional referential	existence of referenced table from a referencing table	uses SQL constructs introduced under the cardinality rule above
referential **constraints**				bidirectional referential	existence of referencing table from referenced table 1:1	not fully definable using SQL constructs. Enforced by service
referential **constraints**				mutually exclusive referential	existence of referenced tables from a referencing table	uses combination of SQL cardinality constructs and CHECK clauses

Table 12 (1 of 2)

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
general constraints				CHECK constraint	requires that a specific search condition not be false for any row in a table	available to IRDS "definer" to define additional constraints
general constraints				ASSERTION definition	integrity constraint that may relate the content of individual rows of a table to the entire content, or to a state between tables	

Table 12 (2 of 2)

6.3.7 ANSI IRDS/ISO IRDS DMF Elements—Sets/Subsets of Records

Reference **ELEMENT**	**ANSI IRDS**			**ISO IRDS**		
	DMF TERM	**DMF DEFINITION**	**NOTES**	**DMF TERM**	**DMF DEFINITION**	**NOTES**
composite						
Composite **made of** record composite				Working sets Reference paths		
multiplicity of made of						
subtype						
Subtype record **subtype of** record				Subtable column	Refers in subtable to supertable	
Multiplicity of subtype of				Single inheritance		
View			The ANSI concept of views and partition is partly a concept of view (as defined here) and partly a concept of access control	Tables		
view record **view of** record view record	SUB SCHEMA	application view of IRDS data	SI Extension		views can be defined over base tables. Views are defined as tables	
Multiplicity of base records						

Table 13 (1 of 2)

Reference ELEMENT	ANSI IRDS			ISO IRDS		
	DMF TERM	DMF DEFINITION	NOTES	DMF TERM	DMF DEFINITION	NOTES
View	TEMPLATE	structured record buffers, means of communication, unit of data transfer	this is part of the local DMF, but its definition is part of the global DMF	-		
view record view of record view record						
Multiplicity of base records						

Table 13 (2 of 2)

6.4 **Analysis and Findings**

The purpose of this partial analysis is to identify items that would compromise compatibility and interoperability. The Data Modeling Facility is only one of the comparison factors. Furthermore, this analysis is preliminary, and the conclusions reached should be interpreted as: "more detailed analysis in this area would probably prove that these items are compatible (or incompatible)."

Interoperability for this comparison factor can be discussed within the framework of export/import operations. No distinction is made between the IRD definition level and the IRD level. Both will be analyzed later.

6.4.1 **Basic Units**

6.4.1.1 **Compatibility**

Text Value Type

The ANSI IRDS TEXT value type is structured by lines. It is not equivalent to the ISO IRDS CHARACTER value type. However, text structured as the ANSI IRDS defines it could be represented using the ISO IRDS CHARACTER value type by storing an end-of-line delimiter.

Conversely, text structured in an ISO IRDS as CHARACTER would have to be split into lines of the proper length before insertion into an ANSI IRDS.

Domains and Allowable Values

The facilities to verify allowable values for attributes in the ANSI IRDS (VALUE and RANGE) are a subset of those provided, in the ISO IRDS, by the DOMAIN and CHECK facilities or by the UNIQUE constraint on non-key attributes.

All domains and allowable values specified within an ANSI IRDS could be specified in an ISO IRDS, but the converse is not true. That is, more strict value constraints could be defined in an ISO IRDS, and these could not be replicated in an ANSI IRDS.

Multivalued Data Elements and Data Groups

The ISO IRDS does not support any of these constructs, and does not have equivalent constructs. Such constructs could, however, be stored in and retrieved from an ISO IRDS using a CHARACTER ATTRIBUTE and using field and value delimiters known to the tool using the IRDS. They would, however, be unknown to the IRDS and could not be manipulated by its services.

Allowed Dependencies

CHECK, ASSERTION, and derivation equations can be put in an ISO IRDS to tie the value of an attribute to the value of other attributes, or to the state of some records or combination of records. Since there is no equivalent mechanism in an ANSI IRDS, such value constraints could not be defined, and could not be replicated in an ANSI IRDS.

6.4.1.2 **Interoperability**

Defining the Same IRDS

Assuming a given conceptual schema for an IRDS, and assuming that an IRDS is designed for ease of update and schema maintenance more than access performance (in such a case the IRDS schema will be as normalized as possible), then a workable ANSI and ISO IRDS could be defined from this conceptual schema.

Since the normalized schema would eliminate multivalued attributes, attribute groups, derived data elements, and non key dependencies, these constructs of the ANSI IRDS would not be used.

However, the ISO IRDS could implement more value constraints and residual dependency checks.

Export/Import (1 Way) Between IRDSs

In such an environment, data could be transferred once between the IRDSs. However there is the possibility that data accepted by the ANSI IRDS would be rejected by the ISO IRDS, if the ISO IRDS implements stricter constraints.

Data accepted by the ISO IRDS would be acceptable to the ANSI IRDS, since the ISO IRDS has equal or stricter constraints.

Export/Import (2 Ways) Between IRDSs

For the reason stated above, transfer of data from an ISO IRDS to an ANSI IRDS and then back to an ISO IRDS would be somewhat difficult, because of the possibly less stringent value constraints of the ANSI IRDS. In a situation where the constraints of the conceptual schema that were not implemented within the IRDS are implemented by the tool layer above it, and where the constructs not available are simulated (e.g., multivalued attributes), there could be export/import between the two IRDSs. More carefully stated, there would be nothing in either data modeling facility that would prevent such interoperability.

6.4.2 **Records**

6.4.2.1 **Compatibility**

Tables vs. Entities and Relationships

The ISO IRDS TABLE construct can be used to represent entities and the two types of relationships. For entities the match is complete except for the identifier, discussed below. For relationships, some conventions have to be made. For example, each relationship corresponds to a table (1:1-1:n relationships without attributes are not collapsed in a FOREIGN KEY reference) and the table is understood to be bidirectional. Identifiers of relationships are discussed below.

Since the table construct contains less semantic information about the behavior of the real world than the entity and relationship constructs, some columns would have to be added to the tables when an ISO IRD is defined, to carry that semantic information. For example, a table type would reveal if the table corresponds to an entity, a non sequenced relationship, or a sequenced relationship. Additional referential constraints would be based on the value of that column emulating the proper behavior (see next section).

Identifiers

ISO IRDS tables are identified by a simple data element. ANSI IRDS entities and relationships are identified by composite data elements, whose values are the concatenation of values of other data elements. This is analogous to a data group.

The ANSI IRDS ACCESS-NAME triplet ASSIGNED-ACCESS-NAME, VARIATION-NAME, REVISION-NUMBER, could correspond to a concatenated value for the ISO ASN_ACC_NAME.

For relationships, however, the corresponding "relationship table" of each relationship type would have to include additional columns for the ASN_ACC_NAME of the participating "entity table" and a sequencing element, if required. These additional columns would have to be made UNIQUE as a group. Separate columns are required, instead of simple concatenation of values, to enable the expression of constraints and referential integrity.

6.4.2.2 **Interoperability**

Defining the Same IRDS

Assuming a given conceptual schema for an IRDS, workable ANSI and ISO IRDSs could each be defined from this conceptual schema. However, the ISO IRDS would need additional columns and constraints to implement, in the table construct, the semantics of the relationship construct.

Export/Import (1 Way) Between IRDSs

In such an environment, data could be transferred once between an ANSI IRDS and an ISO IRDS. However, transfer from an ISO IRDS to an ANSI IRDS would be possible only from a specially designed ISO IRDS. If the semantics of the relationship construct are to be preserved, the same specially designed ISO IRDS is required to accept ANSI IRDS transfers. Both types of transfer involve more than a simple mapping, and the tool layer above each IRDS would have to contain the transformation logic.

Export/Import (2 Ways) Between IRDSs

Assuming a specially designed ISO IRDS, transfer of data to and from ANSI and ISO IRDSs would not be more difficult than one way transfer, and would result in no loss of semantics for the relevant basic constructs. However, we will see in the next sections other factors that would prevent this transfer.

6.4.3 **Record References and Constraints**

6.4.3.1 **Compatibility**

Because the two ANSI IRDS record constructs are not symmetrical, the rules for referencing are different depending on which is the referenced record and which is the referencing record. Furthermore, no referencing is allowed between constructs of the same type. All the referencing rules specified in the ANSI IRDS can be implemented in an ISO IRDS with the proper combination of REFERENCES, UNIQUE and NOT NULL. This assumes, as above, a specially designed ISO IRDS.

As the ISO IRDS allows for a table to reference more than one table, an ISO IRDS could contain constructs that have no equivalence in an ANSI IRDS, such as a table playing the role of a ternary relationship, a subtyping reference between two tables playing the role of entities, etc.

Because of the possibility of referential integrity actions, an ISO IRDS might have defined, in its tables, update and delete actions that would be part of a service definition in an ANSI IRDS.

Finally, using the CHECK and the ASSERTION mechanisms, an ISO IRDS could be designed that implements, in the data modeling facility, a larger number of the conceptual schema constraints, thus ensuring better integrity.

6.4.3.2 **Interoperability**

Defining the Same IRDS

Assuming a given conceptual schema for an IRDS, workable ANSI and ISO IRDSs could each be defined from this conceptual schema. However, the ISO IRDS could implement more reference paths and integrity constraints. It could not implement the semantics of the records without additional attributes and constraints. Making both implementations compatible requires a special design for the ISO IRDS.

Export/Import (1 Way) Between IRDSs

In such an environment, data could be transferred once between the IRDSs. However, there is the possibility that data accepted by the ANSI IRDS will be rejected by the ISO IRDS.

Export/Import (2 Ways) Between IRDSs

For the reason stated above, there would be some difficulty in transferring data from an ISO IRDS to an ANSI IRDS and back to an ISO IRDS, because of the possibly less stringent value constraints of the ANSI IRDS. In a situation where the constraints of the conceptual schema that were not implemented within the IRDS are implemented by the tool layer above it, there could be export/ import between the two IRDSs.

6.4.4 **Sets/Subsets of Records**

6.4.4.1 **Compatibility**

The ANSI IRDS uses the concept of a subschema, that is, an application view over the schema. In an ISO IRDS environment, the same level of isolation could be attained by always defining "view" tables over "base" tables, and never operating on the base tables.

The ANSI IRDS template and the ISO IRDS view facilities are similar as defined, but not in operations. Templates at the meta level are predefined, and as such not equivalent to views.

6.4.4.2 **Interoperability**

This will be discussed with the services.

7. ANALYSIS OF IRDS DEFINITION OPERATIONS

7.1 Introduction

This chapter compares the two IRDS standards in terms of the possible operations at the definition level. The various categories are described in section 7.3.

7.2 Local Data Modeling Facility

This is the data organization as exchanged at the IRDS Interface.

In the ISO IRDS, Pascal is used as the specification language, with the correspondence established between SQL and Pascal data types.

In the ANSI IRDS, the interface is specified in a programming language independent manner, the only parameter exchanged being the address of the buffered template.

7.3 Reference Definition Operations

As introduced in chapter 4, operations are organized into classes to facilitate comparison between the different approaches. The classes used for definition and administration operations are introduced below.

<u>Implementation Characteristics</u>

This class of operations gives access to the values given by the implementor to some of the parameters of the IRDS. Some of these values may be set at installation time by the IRDS administrator.

<u>Definition Session Control</u>

These services enable the user to initiate and terminate an IRDS definition session.

<u>Definition Library Control</u>

These services provide maintenance operations on an IRDS definition library, defined, in the ISO terminology, as a version of a working set of one definition (schema). The ANSI IRDS library concept implements only the notion of versions within one schema.

<u>Definition Transaction Control</u>

These operations enable the user to initiate and terminate IRDS definition transactions.

Definition Transaction Record Naming and Selection

These operations support the various naming schemes and retrieval mechanisms for records.

Access Control Definition

These operations enable the IRDS administrator to define and maintain access rights to the definition level and the dictionary level.

Dictionary Content Definition

These operations define and maintain the definition of the IRD itself, that is, the items that are dependent on the DMF selected.

Naming Definition and Control

These operations facilitate the implementation and control of the naming scheme of the IRDS.

Dictionary Library Definition

These operations maintain and activate the definition (schema) of an IRD.

7.4 Comparison of the ANSI and ISO IRDSs

The tables in this section compare the service classes introduced in section 7.3.

The tables are structured as follows:

Column 1	**Reference Element**, established in section 6.2
Column 2	**ANSI IRDS Service** identification corresponding to the reference element
Column 3	**ANSI IRDS Service Input/Output**
Column 4	**ISO IRDS Service** identification corresponding to the reference element
Column 6	**ISO IRDS Service Input/Output**

Because there is rarely a one-to-one match, some parts of the tables are repeated to improve clarity.

Items appear on the same line when there is some equivalence. If there is no equivalence, the items appear on separate lines.

If the service has many types of input/output, the name of the service is not repeated on each line.

7.4.1 Implementation Characteristics

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
Implementation characteristics				
	Retrieve Meta-Entity	IRDS Defaults ... Template Type ... Template Type Tree	Add Object Retrieve Object Modify Object Delete Object	IRDS Default
		IRDS Limits ... Template Type ... Template Type Tree	Retrieve Object	Implementation Limit
		IRDS Reserved Names ... Template Type		IRD Definition Level Reserved Names IRD Level Reserved Names
		Names ... Template Type ... Template Type Tree		
				IRD Module

Table 14

7.4.2 Definition Session Control

Reference **ELEMENT**	**ANSI IRDS**		**ISO IRDS**	
	SERVICE	**SERVICE INPUT/OUTPUT**	**SERVICE**	**SERVICE INPUT/OUTPUT**
definition session control				
	Open IRDS	Session Template Type	Open IRDS	-IrdDefName identifies definition -IrdSchemaWkgSet Name & VerId = "" to indicate a definition session
	Open IRD Schema	IRD Schema Template Map Template Type Tree		
			Create IRD Definition	IrdDefName identifies definition
	Retrieve Session Default			
	Modify Session Default			
	Close IRD Schema			
	Close IRDS		Close IRDS	

Table 15

7.4.3 Definition Library Control

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
definition library control				
	-Add Meta-Entity -Retrieve Meta-Entity -Modify Meta-Entity Meta-Attributes -Delete Meta-Entity	IRD Partition ... Template Type ... Template Type Tree	Modify Content Status	IRD Definition Working Set
	Modify Meta-Entity Version Set Assigned-Access-Name	Meta-Entity Modify Version Set Name Template Type		
	Modify Meta-Entity Version Set Assigned-Descriptive-Name			
	Modify Meta-Entity Life-Cycle-Phase			

Table 16 (1 of 2)

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
definition library control				
			Create Working Set	IRD Definition Object
				IRD Definition Working Set
				IRD Definition Working Set-Xref
			Create Reference Path Modify Reference Path Drop Reference Path	IRD Definition Working Set (To)
			Add Object Retrieve Object Modify Object Delete Object	IRD Definition Object
				IRD Definition Working Set
				IRD Definition Working Set-Xref
			Set Context	
			Drop Working Set	
			Drop IRD Definition	

Table 16 (2 of 2)

7.4.4 Definition Transaction Control

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
definition transaction control				
	Commit		Commit	
	Rollback		Rollback	
	Retrieve Message Line	IRDS Message Template Type	Get Diagnostics	

Table 17

7.4.5 **Definition Transaction Record Naming and Selection**

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
definition record naming				
	-Add Meta-Entity -Retrieve Meta-Entity -Modify Meta-Entity Meta-Attributes -Delete Meta-Entity		-Add Object -Retrieve Object -Modify Object -Delete Object	
				IRD Substitute Name
		Meta-Entity-Type Map Template-Type		
		Variation-Names-Data ... Template Type ... Template Type Tree		IRD Variation Name
	-Add Meta-Relationship -Retrieve Meta-Relationship -Modify Meta-Relationship Meta-Attributes -Delete Meta-Relationship			
		Meta-Relationship-Type Template Type		
		Entity-Type uses Variation-Names-Data ... Template Types ... Template Type Trees		IRD Variation Name Set Usage
	Modify Meta-Entity Version Set Assigned-Access-Name			
	Modify Meta-Entity Version Set Assigned-Descriptive-Name			

Table 18 (1 of 2)

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
definition transaction position control			-	
	SIfirst, SI..., SI..., ..., SIdelr	Proposed Extensions	Open Cursor	
			Close Cursor	
definition transaction record retrieval				
	-Retrieve Meta-Entity -Retrieve Meta-Relationship	Use of wildcards allows for retrieval of multiple records following selection criteria	Retrieve Object	Use of the SELECT operation allows for retrieval of multiple records following selection criteria

Table 18 (2 of 2)

7.4.6 Access Control Definition

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
definition access control definition				
	-Add Entity -Retrieve Entity -Modify Entity Attributes -Delete Entity		-Add Object -Retrieve Object -Modify Object -Delete Object	
			IRD Definition Object	
		IRDS User	User	
		IRD Schema View Retrieval Template Type		
		IRD Schema View		Definition Working Set Privilege
	-Add Relationship -Retrieve Relationship -Modify Relationship Attributes -Delete Relationship			
		IRDS User has IRD Schema View		

Table 19 (1 of 2)

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
dictionary access control definition				
	-Add Entity -Retrieve Entity -Modify Entity Attributes -Delete Entity		-Add Object -Retrieve Object -Modify Object -Delete Object	
				IRD Definition Object
		IRDS User		User
				Working Set Privilege
		IRD View Retrieval Template Type		
		IRD View		IRD Table Privilege
				IRD Column Privilege
	-Add Relationship -Retrieve Relationship -Modify Relationship Attributes -Delete Relationship			
		IRDS User has IRD View		

Table 19 (2 of 2)

7.4.7 Dictionary Content Definition

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
basic DMF dictionary definition				
	Retrieve Meta-Entity Type Template Map			
		Meta-Entity Type Map Template Type		
	-Add Meta-Entity -Retrieve Meta-Entity -Modify Meta-Entity Meta-Attributes -Delete Meta-Entity		Add Object Retrieve Object Modify Object Delete Object	
				IRD Definition Object
		Entity-Type Template Type ... Template Type Tree		IRD Table
		Relationship-Class-Type ... Template Type ... Template Type Tree		
		Relationship-Type ... Template Type ... Template Type Tree		IRD Table
		Attribute-Group-Type ...Template Type ...Template Type Tree		
		Attribute-Type ...Template Type ...Template Type Tree		IRD Column

Table 20 (1 of 5)

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
basic DMF dictionary definition				
		Attribute Validation Data ...Template Type ...Template Type Tree		
		Attribute Validation Procedure ...Template Type ...Template Type Tree		
				IRD Table Constraint
				IRD Domain
				IRD Assertion
				IRD Check Constraint
	Copy Meta-Entity			
		see above for meta-entities		
		Meta-Entity Copy Template Type		

Table 20 (2 of 5)

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
basic DMF dictionary definition				
	Retrieve Meta-Relationship-Type Template Map			
		Meta-Relationship-Type Template Type		
	-Add Meta-Relationship -Retrieve Meta-Relationship -Modify Meta-Relationship Meta-Attributes -Delete Meta-Relationship		-Add Object -Retrieve Object -Modify Object -Delete Object	
		Entity-Type **contains** Attribute-Type and Attribute-Group-Type ...Template Types ...Template Type Trees		IRD Column
		Relationship-Type **contains** Attribute-Type and Attribute-Group Type ...Template Types ...Template Type Trees		IRD Column
				IRD Key Column Usage

Table 20 (3 of 5)

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	**SERVICE INPUT/OUTPUT**	**SERVICE**	**SERVICE INPUT/OUTPUT**
basic DMF dictionary definition				
		Relationship-Type **connects** Entity-Type ...Template Type ...Template Type Tree		
		Relationship-Type **member of** Relationship-Class-Type ...Template Type ...Template Type Tree		
		Attribute-Group-Type **contains** Attribute-Type ...Template Type ...Template Type Tree		
		Attribute-Type **uses** Attribute-Type Validation-Data ...Template Type ...Template Type Tree		
		Attribute-Type **uses** Attribute-Type Validation-Procedure ...Template Type ...Template Type Tree		
				IRD Check Table Usage
				IRD Check Column Usage

Table 20 (4 of 5)

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
basic DMF dictionary definition				
	Add Service for Arrayed Template			
	Retrieve Service for Arrayed Template			
	Modify Service for Arrayed Template			
	Delete Service for Arrayed Template			
	Attribute Type Decoded Maximum Length			
	Decode Attribute Value			
view dictionary definition			-Add Object -Retrieve Object -Modify Object -Delete Object	
				IRD Definition Object
				IRD Table
				IRD Column
				IRD View Table Usage
				IRD View Column Usage
supertable/ subtable creation			Declassify Object Reclassify Object	IRD Table

Table 20 (5 of 5)

7.4.8 **Naming Definition and Control**

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
naming				
	-Add Meta-Relationship -Retrieve Meta-Relationship -Modify Meta-Relationship Meta-Attributes -Delete Meta-Relationship		-Add Object -Retrieve Object -Modify Object -Delete Object	
		Entity-Type uses Variation-Names-Data ...Template Types ...Template Type Trees		IRD Definition Object
				IRD Variation Name Set
				All IRD Names (View)
				All SQL Names (View)

Table 21

7.4.9 Dictionary Library Definition

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
dictionary library definition				
			-Add Object -Retrieve Object -Modify Object -Delete Object	
				IRD Definition Object
				IRD Schema
			Validate IRD Schema	IRD Schema
			Create IRD	IRD Schema
			Drop IRD	IRD Schema
	Activate IRD		Activate IRD	IRD Schema
	Deactivate IRD		Deactivate IRD	IRD Schema
	Restore IRD Schema			

Table 22

7.5 Analysis and Findings

7.5.1 General

The notion of schema and the number of defined IRD schemas available to activate is different in the ANSI IRDS (one) and the ISO IRDS (many).

Relationships have no versions in the ANSI IRDS.

7.5.2 By Operation Classes

Implementation Characteristics

In this area the differences are mainly due to different design choices made by the standards builders, and are not fundamental. For example, the reference to the IRDS standard prescribing some or all the definition level is made by a reference to a table (IRDS MODULE) in the ISO IRDS, and by the value of an attribute (ORIGIN) in the ANSI IRDS.

Definition Session Control

Because the ANSI IRDS assumes a minimal IRD schema, and the ISO IRDS can be initiated with an empty IRD, the first session control operations on the definition of an IRD would be done in a different manner. After the first session, the operations are fairly equivalent.

Definition Library Control

In this area, the differences of approach are substantial. In the ISO IRDS, there can be many versions of a definition working set, each definition can have many working sets, and there can be many definitions. In the ANSI IRDS, there can be only one definition (schema), and this definition can have many versions.

Naturally, one major difference is in the versioning scheme, as was established earlier. There are also other fundamental differences that would inhibit interoperability.

Although the ANSI IRDS approach could be emulated using ISO IRDS operations, the reverse is not true, since some ISO IRDS operations do not have equivalents in the ANSI IRDS.

Definition Transaction Control

These operations are similar in the two IRDSs.

Definition Transaction Record Naming and Selection

These operations are different in the two IRDSs. The ANSI IRDS templates offer retrieval of many records at a time and navigation within the given retrieval tree (using the extended services). The same is not available for the ISO IRDS. However, for selecting individual records, the SELECT operator for retrieval is much more flexible than wildcards.

This is one of the important differences that would make interoperability impossible, unless restrictions are put on the retrieval mechanisms.

Access Control Definition

The major differences in the two IRDSs at the definition level is in the level of granularity of access control. The ANSI IRDS gives access at the library level (partition), whereas the ISO IRDS gives access at two levels below, at the record and data element level (table and column privilege). Security extensions proposed in the ANSI IRDS refine the permissions given and increase the level of granularity to the record level by introducing subschema.

Although the difference between the access control mechanisms would not prevent interoperability of other definition operations, they would have to be made following the policies of the less secure environment, and access control operations could not be made equivalent.

Dictionary Content Definition

Given that the difference in DMF has already been discussed, the difference here is not on the basic services, but on additional facilities, namely views and templates.

Although both facilities want to isolate the client of the services interface from the basic IRDS data, the similarity ends there. Templates are mandatory for the ANSI IRDS services interface, and views are optional in the ISO IRDS. Operations to define views are provided, but operations to define templates are not.

Naming Definition and Control

Although the ANSI IRDS does not have facilities equivalent to the ISO IRDS ones, some template services provide identifications of names used.

Dictionary Library Definition

These operations are similar.

8. ANALYSIS OF IRDS DICTIONARY OPERATIONS

8.1 Introduction

This section compares the operations performed on a defined dictionary.

8.2 Reference Dictionary Operations

As introduced in chapter 4, operations are organized into classes to facilitate comparison between the different approaches. The classes used for definition and administration operations are introduced below:

Dictionary Session Control.

These services enable the user to initiate and terminate an IRDS dictionary session.

Dictionary Library Control.

These services provide maintenance and selection operations on an IRDS dictionary library, defined, in the ISO terminology, as versions having different statuses.

Dictionary Transaction Control.

These operations enable the user to initiate and terminate IRDS dictionary transactions.

Dictionary Content Manipulation.

These operations define and maintain the definition of the IRD itself, that is, the items that are dependent on the DMF selected.

Naming Definition and Control.

These operations facilitate the implementation and control of the naming scheme of the IRDS.

8.3 Comparison of the ANSI and ISO IRDSs

The tables in this section compare the service classes introduced in section 8.2.

The tables are structured as follows:

Column 1	**Reference Element**, established in section 6.2
Column 2	**ANSI IRDS Service** identification corresponding to the reference element

Column 3 **ANSI Service IRDS Input/Output**
Column 4 **ISO IRDS Service** identification corresponding to the reference element
Column 5 **ISO IRDS Service Input/Output**

Because there is rarely a one-to-one match, some parts of the tables are repeated to improve clarity.

Items appear on the same line when there is some equivalence. If there is no equivalence, the items appear on separate lines.

If the service has many types of input/output, the name of the service is not repeated on each line.

8.3.1 Dictionary Session Control

Reference	ANSI IRDS		ISO IRDS	
ELEMENT	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
dictionary session control				
	Open IRDS			
		Session Template Type		
	Open IRD		Open IRDS	-IrdDefName identifies applicable definition -IrdSchemaWkgSetName & VerId identifies applicable definition
		IRD Template Map Template Type Tree		
	Retrieve Session Default			
	Modify Session Default			
	Close IRD			
	Close IRDS		Close IRDS	

Table 23

8.3.2 Dictionary Library Control

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
dictionary library control				
	Modify Entity Version Set Assigned-Access-Name			
	Modify Entity Version Set Assigned-Descriptive-Name			
		Entity Modify Version Set Name Template Type		
	Modify Entity Life-Cycle-Phase		Modify Content Status	
		Entity Modify Life-Cycle-Phase Template Type		IRD Working Set
			Create Working Set	
				IRD Definition Object
				IRD Working Set
				IRD Working Set-Xref
				IRD Object Version
			Add Object Retrieve Object Modify Object Delete Object	
				IRD Definition Object
				IRD Working Set
				IRD Working Set-Xref
			Set Context	
			Drop Working Set	

Table 24

8.3.3 Dictionary Transaction Control

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
dictionary transaction control				
			Prepare	
	Commit		Commit	
	Rollback		Rollback	
	Retrieve Message Line		Get Diagnostics	
		IRDS Message Template Type		

Table 25

8.3.4 Dictionary Transaction Record Naming and Selection

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
dictionary record naming				
	-Add Entity -Retrieve Entity -Modify Entity Attributes -Delete Entity		-Add Object -Retrieve Object -Modify Object -Delete Object	
		Variation Names Data ... Template Type ... Template Type Tree		IRD Variation Name
				IRD Substitute Name
	-Add Relationship -Retrieve Relationship -Modify Relationship Attributes -Delete Relationship			
		Entity-Type uses Variation-Names Data ... Template Types ... Template Type Trees		IRD Variation Name Set Usage
	Modify Entity Version Set Assigned-Access-Name			
	Modify Entity Version Set Assigned-Descriptive-Name			

Table 26 (1 of 2)

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
dictionary transaction position control				
	SIfirst		Open Cursor	
	...		Close Cursor	
	SIdelr			
dictionary transaction record retrieval				
	-Retrieve Entity -Retrieve Relationship	Use of wildcards allows for retrieval of multiple records following selection criteria	Retrieve Object	Use of the SELECT operation allows for retrieval of multiple records following selection criteria

Table 26 (2 of 2)

8.3.5 Dictionary Content Manipulation

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
base dictionary manipulation				
	Retrieve Entity-Type Template Map			
		Entity-Type Map Template Type		
	-Add Entity -Retrieve Entity -Modify Entity Attributes -Delete Entity		-Add Object -Retrieve Object -Modify Object -Delete Object	
		Generic Entity-Type Template Type		IRD Object
				IRD Object Version
		Entity-Type Template Type Trees		
		Generic Text Attribute-Type Template Type		
		Generic Plural Attribute(-Group)-Type Template Type		
	Copy Entity			
		Entity Copy Template Type		

Table 27 (1 of 3)

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
base dictionary manipulation				
	Retrieve Relationship-Type Template Map			
		Relationship-Type Map Template Type		
	-Add Relationship -Retrieve Relationship -Modify Relationship Attributes -Delete Relationship		-Add Object -Retrieve Object -Modify Object -Delete Object	
		Generic Relationship-Type Template Type		IRD Object
				IRD Object Version
		Relationship-Type Template Type Trees		
		Generic Text Attribute-Type Template Type		
		Generic Plural Attribute(-Group)-Type Template Type		
	Attribute-Type Decoded Maximum Length			
	Decode Attribute Value			
		Attribute-Type Decoded Value Template Type		

Table 27 (2 of 3)

Reference ELEMENT	ANSI IRDS		ISO IRDS	
	SERVICE	SERVICE INPUT/OUTPUT	SERVICE	SERVICE INPUT/OUTPUT
view dictionary manipulation			-	
			-Add Object -Retrieve Object -Modify Object -Delete Object	
				IRD Object
				IRD Object Version

Table 27 (3 of 3)

8.4 **Analysis and Findings**

8.4.1 **By Operation Classes**

<u>Dictionary Session Control</u>

These operations are similar in the two IRDSs, except for the fact that the ANSI IRDS supports some session defaults.

<u>Dictionary Library Control</u>

Although the operations are similar, the difference in concepts and implementation between the ISO IRDS versioned working set and the ANSI IRDS versioned identifier is such that major incompatibilities exists.

<u>Dictionary Transaction Control</u>

These operations are similar in the two IRDSs.

<u>Definition Transaction Record Naming and Selection</u>

These operations are different in the two IRDSs. The ANSI templates offer retrieval of many records at a time and navigation within the given retrieval tree (using the extended services). The same is not available for the ISO IRDS. However, for selecting individual records, the SELECT operator for retrieval is much more flexible than wildcards.

This is one of the important differences that would make interoperability impossible, unless restrictions are put on the retrieval mechanisms.

Dictionary Content Manipulation

Given that the difference in DMF has already been discussed, the divergence here is not on the basic services, but on additional facilities, namely views and templates.

Although both facilities have the goal of isolating the client of the services interface from the basic IRDS data, the similarity ends there. Templates are mandatory for the ANSI IRDS services interface, and views are optional in the ISO IRDS. Operations to define views are provided, but operations to define templates are not.

9. SURVEY OF OTHER PROPOSALS/PRODUCTS

9.1 Introduction

This chapter is not an analysis of compatibility, but a high-level survey of three other proposals or products, focusing on areas where differences from the IRDSs may be meaningful. The formal process of building detailed tables mapping each proposal or product to a reference base, and then mapping them against one another was not followed. The items mentioned in this section should therefore be considered as areas for further investigation, more than definitive statements about equivalence or difference. As with the ANSI and ISO IRDS standards, it is assumed that the reader is familiar with the base documents of each proposal or product.

9.2 PCTE

9.2.1 Introduction

PCTE [PCTE 1988] is intended to be a platform providing common services to various tools in a Software Engineering Environment. One component of PCTE, the Object Management System (OMS), provides services analogous to what an IRDS would provide. It is the data modeling facility aspect of the Object Management System that is surveyed here. Unless otherwise noted, the abbreviation PCTE in the remainder of this section should be understood as PCTE OMS.

The PCTE OMS can be seen as an evolution of the traditional file management system. It is, in fact, a specialized database management system. Compared to a general purpose DBMS, PCTE does not offer all the services associated with the DBMS, but it hides distribution of the data objects from the user.

From the UNIX file system, PCTE has inherited the characteristic that any access to an object is the result of a navigation operation, or the expression of a fully qualified pathname whose syntax is a superset of the UNIX one.

The PCTE data modeling facility is a variation of the E-R model as initially proposed by P. Chen to prepare conceptual schema [CHEN, P. 1977, p. 9]. As in the case of the ANSI IRDS, using such a technique for an external (or logical) schema forces some compromises.

The DMF has strong affinity with the network data model, and relationships are introduced as pairs of links. Those links have a definition somewhat similar to sets in a network DMF. However, as is the case with the ANSI IRDS, the variation selected by PCTE restricts relationships to binary relationships. PCTE also has predefined constructs to define constraints. Constructs to describe aggregation and subtyping are also added to the basic DMF. Some object types may have an associated content, subject to specific operations.

9.2.2 **Basic Units**

PCTE has value types similar to the ANSI IRDS and ISO IRDS data types, that is, STRING, BOOLEAN, INTEGERS, REAL NUMBERS, and DATE.

The notion of DOMAINS is implemented in PCTE by the ENUMERATION data type.

Data elements are called attribute types, and are single valued.

Dependencies, or absence thereof, between attributes cannot be enforced, and data groups are not supported as separate constructs. As such, first normal form is enforced. However, both object types and link types contain sets of references, which are similar to data groups or multivalued data elements, containing keys to other records.

9.2.3 **Records**

<u>Record types</u>

PCTE has two basic types of record, the object type and the link type. The relationship type is introduced in PCTE as a pair of links such that the origin of each is the destination of the other. The purpose of this derived construct is to enforce integrity constraints between two link types, However, link types are not necessarily members of relationship types. Some link types control existence, composition, and reference.

Both the object type and the link type can have attributes. Object types are similar to ANSI entity types, and relationship types are similar to ANSI IRDS relationship types, although PCTE explicitly decomposes relationships into their two constituent link types, the link type being the basic definition unit for attributes and constraints. One of the two constituent links of a relationship can be declared with the category implicit, and is then maintained as a side effect of the other constituent link. The ANSI IRDS, on the other hand, defines everything at the relationship type level, with the constituent directed relationships implicit. ANSI IRDS sequenced and unsequenced relationships have some equivalence in PCTE, with multiple link key attributes. Entity types, relationship types and link types correspond to different TABLE types in the ISO IRDS.

The notion of an object type hierarchy is inherent in PCTE, and all objects must belong to such a hierarchy. Each hierarchy has the common ancestor type "object," and attributes and links are inherited. Multiple inheritance is possible. The possible set of predefined object types partition the inheritance graph. However, relationships and links cannot be subtyped. This is discussed further under the section Sets/Subsets of Records.

Identifiers

In the ANSI IRDS DMF, relationships are identified by the concatenation of the keys of participating entities, within relationship types. In PCTE, the opposite approach is taken. Objects are named by the concatenation of the keys of a specified set of links. If the cardinality of these links is one, then the link name is a sufficient identifier; if the cardinality is many, then the link has an ordered set of key attributes.

Relationships are not basic constructs, and therefore have no identifiers.

The sequence of link names (and keys) used to identify an object is called a pathname, and a given object can be identified unambiguously by two pathnames. Paths may have a reference object as origin. Reference objects can provide direct access to an object once it has been navigated to by a process.

Objects also have some surrogates, an attribute of name "system-object_exact_identifier," and a volume and object number, although these may change.

The PCTE identification scheme is very different from the one in both the ANSI and ISO IRDS. This alone would prevent easy movement of data between these environments.

9.2.4 **Record References and Constraints**

References

In the ISO IRDS, one-way references between tables are defined via foreign keys. In the ANSI IRDS, one-way references between relationships and entities are established by definition of relationships, and the converse reference between entities and relationships is not maintained. No reference can be established between entities or between relationships.

In PCTE, two way references are maintained between object types and link types. A link type will contain a set of origin object types, and a set of destination object types. Conversely, the object type will contain the set of links for which it is an origin, and the set of objects for which it is a destination, These sets of references are either specific to the objects, or inherited from parent object types.

As in the ANSI IRDS, the cardinality of a link is one or many, whether one or many links of this type may start from the origin object. The ANSI IRDS cardinality of none (0), which, in fact, means optional participation, is the default, and mandatory participation is not supported. ECMA PCTE will allow optional/ mandatory, and unique/non-unique restrictions.

Constraints

Referential integrity and referential constraints are supported by inherent characteristics of some categories of links, and some link properties.

The existence link implies deletion of an object if it is not the target of any existence link. The reference link prevents the deletion of the referenced object. If a reference is desired without that existence constraint, then a designation link can be used. With the stability property, update to the target of a link can be prevented. This also prevents the creation of further links to that target.

The ANSI IRDS does not support a constraint mechanism, but the ISO IRDS has the mechanism to support all the PCTE constraints, and more.

9.2.5 **Sets/Subsets of Records**

Aggregation

Some form of aggregation is mandatory in PCTE, and the creation of a new object requires the creation of a link—an existence link between an origin object and the new object. In earlier versions of PCTE, this was the composition link, under the assumption that the existence of the part is dependent on the existence of the whole.

The composition link is created to an existing object. The graph of a composition link is not restricted to a hierarchy, nor to a directed acyclic graph. It may contain cycles. However, the composition link can have the exclusiveness property, which can be used to enforce a tree structure on the graph of composition links.

To enable manipulation of a set of objects as a whole, PCTE defines the notion of a composite entity. A composite entity consists of a root object, a set of composition links, and a set of objects—components—via the composition link, of the root object.

Although an entity in the ANSI IRDS can be defined as being made of other entities, this is equivalent to the root of a PCTE composite object, and not to the composite object itself.

In the ISO IRDS there exists only one type of composite object, and it is called a working set.

Subtyping

Subtyping is also mandatory in PCTE, and the creation of a new object type requires the definition of what object it is a subtype of.

Subtyping associations cannot be explicitly manipulated, and are maintained by the system. This does not exist in the ANSI IRDS. In the ISO IRDS, subtyping is implemented as a foreign key in the subtable, and can be manipulated.

Views

Working schemas can be defined, even dynamically, and correspond to the set of objects, links, and attributes that can be manipulated within a process. These play a role analogous, but not identical, to subschemas in the ANSI IRDS and views in the ISO IRDS.

Schema

PCTE has no explicit global schema. An implicit global schema is implied by the total set of local schema, or SDSs. This approach is quite different from the traditional database approach, but may be the practical approach for coping with distributed databases.

9.3 IBM Repository Manager

9.3.1 Introduction

The IBM Repository is an organized collection of information that supports business and data processing activities. Repository Manager [RM 1990] is the platform providing definition, access, and manipulation of that repository. Part of the services provided by Repository Manager are analogous to those an IRDS would provide.

Although not explicitly stated, RM assumes an underlying database management system and uses it to provide some of its services. RM is, however, independent of the characteristics of that DBMS. RM is, in fact, implemented over DB2, a relational DBMS.

The RM data modeling facility is a variation [RM 1990, p. 5] of the E-R model initially proposed by P. Chen as the data modeling facility to prepare conceptual schema [CHEN, P. 1977, p. 9]. The comment previously made in connection with the ANSI IRDS and PCTE, about using a conceptual technique at the logical level, is again applicable.

As is the case for the ANSI IRDS and PCTE, the E-R variation defined by RM restricts relationships to binary relationships, with the further restriction that relationships have no attributes. However, extensions (over the restrictions) to support relationships between relationships facilitate the representation of n-ary relationships. As in PCTE, extensions are provided for subtyping and aggregation. RM also has both predefined constructs to define constraints, and general purpose constraint facilities, called repository policies.

Some object types may have associated content, called objects, subject to specific operations.

9.3.2 **Basic Units**

Possible value types (or formats, as they are called) are CHARACTER, FIXED BINARY, and DECIMAL. Boolean, dates, and enumerated types are not supported directly.

The notion of DOMAINS in RM is implemented by the use of integrity policies, where ranges and possible values can be verified at the time of insertion of instances of entities.

Data elements are called attribute types, and are single valued. Data groups are not supported. As such, first normal form is enforced. However, composite values are recommended for key attributes to make them unique, when required, which introduces normalization anomalies.

Dependencies between attributes can be defined/enforced by derivation and integrity policies.

9.3.3 **Records**

Record types

RM has two basic types of record, the entity type and the relationship type.

Entity types are similar to ANSI IRDS entity types, and relationship types are similar to ANSI IRDS relationship types, except for the fact that they have no attributes, and that a relationship can associate other relationships between themselves or to other entities.

RM introduces a construct called dependent entity type, and uses it to enable the representation of data groups, multivalued attributes, and entities for which no unique identifier is available or desired.

Relationships have properties, however, which are used to enforce constraints.

Identifiers

Entity types are identified by one key attribute that has a unique value for each entity. Values can be made unique by adding prefixes to create a context in which the value is unique.

Dependent entities have their own key attribute made unique by the addition of a qualifier, i.e., the key attribute(s) of the parent(s).

Entities can also have a system-assigned key attribute.

Relationships are identified by the concatenation of the keys of the two participating entities.

Relationships can have a property, the ordered set property, which makes the relationship similar to what the ANSI IRDS calls a sequenced relationship.

9.3.4 **Record References and Constraints**

References

As in the ANSI IRDS, one-way references are established between relationships and entities by the definition of the relationships. However, the relationships have more properties than the usual cardinality. These properties define various constraints.

Constraints

The cardinality, or the instance control property, is defined differently than in the ANSI IRDS. In the ANSI IRDS, the cardinality compares the number of instances of the relationship to the number of instances of the participating entity. In RM, the instance control property compares the instances of the two participating entities.

The controlling property causes source instances, target instances, or both, to be deleted when the relationship instances are deleted. This enforces one form of referential integrity, for the delete operation.

The mandatory property requires the existence of a relationship instance if a source instance, a target instance, or both exist.

Besides these preprogrammed constraints mechanisms, RM has what are called repository policies, which enable the implementation of other constraints, using a constraint language (REXX). There are three kinds of policies: derivation policies, integrity policies, and trigger policies. The trigger policies are activated when entities and relationships are manipulated.

9.3.5 Sets/Subsets of Records

Aggregation

Aggregation types are created in RM. Aggregation is made around a root entity, and is not based on a specific association or relationship, as in PCTE.

The composition of an aggregation type is specified by building a chain of aggregation elements, that is, a sequence of relationships and entities.

The resulting aggregation graph is a tree. That is, aggregation is hierarchical.

Subtyping

There is no specific construct in RM to deal with classification. As in the ANSI IRDS, maintaining a relationship between two entities can be used to relate the supertype to the subtype, and the use of the relationship properties can facilitate the conservation of integrity, but there are no specific features to deal with inheritance of properties and relationships. The same mechanisms could be used to classify relationships, as relationships between relationships are allowed.

Views

Combination of aggregation types and templates can give to a process a specialized view of the repository.

9.4 ATIS

9.4.1 Introduction

A Tool Integration System [ATIS 1990] is an object-oriented approach to the integration of tools, providing a set of interfaces that support schema driven dispatching of behavior. ATIS provides an interface to and defines a data model for an IRDS.

Since the document reviewed is, in fact, a proposal to use ATIS constructs to define an IRDS behaving as closely as possible to the ANSI IRDS, only the basic constructs and the differences with the ANSI IRDS are discussed here.

9.4.2 Basic Units

Data types in ATIS are more elaborate than in the other proposals or standards. Since data types are an element in ATIS, they could conceivably be extensible, although this is not the case in the current proposal.

Data elements are called properties, or attributes. Because methods can be defined for objects, or elements, dependencies between attributes can be enforced or prevented.

Relationships are implemented using multivalued data elements, called scan values.

9.4.3 **Records**

Records in ATIS are elements, and all elements are organized in a type hierarchy whose root is the type element. To support the ANSI IRDS, element has two subtypes, named element, representing entities, and relationship. ATIS is currently a single inheritance model.

Since the identification of participants (owner, member) in a relationship can be multivalued, ternary and n-ary relationships can be represented.

Elements (named elements and relationships) are uniquely identified by their element-id's. They also have names.

9.4.4 **Record References and Constraints**

As in PCTE, ATIS allows each element to contain scan valued properties pointing to one or more elements. This means that two-way references have to be maintained.

Referential integrity and referential constraints are enabled using methods.

9.4.5 **Sets/Subsets of Records**

Aggregation element types can be defined for compound elements. Subtyping is a required basic construct in ATIS.

ABBREVIATIONS AND ACRONYMS

ANSI	American National Standards Institute
ATIS	A Tool Integration System
CASE	Computer Assisted Software Engineering
CDD	Common Data Dictionary (DEC product)
CDIF	Common Data Interchange Facility
DB2	Database 2 (IBM product)
DBMS	Database Management System
DEC	Digital Equipment Corporation
DMF	Data Modeling Facility
E-R	Entity-Relationship
ECMA	European Computer Manufacturers Association
EDI	Electronic Data Interchange
FIPS	Federal Information Processing Standard
IBM	International Business Machines
IRD	Information Resource Dictionary
IRDS	Information Resource Dictionary System
ISE	Information Systems Engineering
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
JTC1	Joint Technical Committee 1 (of ISO/IEC)
NBS	National Bureau of Standards (now NIST)
NIST	National Institute of Standards and Technology
OMS	Object Management System (part of PCTE)
PCTE	Portable Common Tool Environment
REXX	IBM interpretive procedure language

RM Repository Manager (IBM product)

SC21 Subcommittee 21 (of ISO/IEC JTC1)

UofD Universe of Discourse

WG3 Working Group 3 (of SC21)

X3 ANSI accredited standards committee on Information Processing Systems

X3H4 X3 technical committee on Information Resource Dictionary System

REFERENCES AND OTHER INPUT

ANSI 1992. American National Standard X3.185-1992, Information Resource Dictionary System—IRDS Services Interface, American National Standards Institute, New York, 1992.

ATIS 1990. "ATIS—A Tools Integration Standard," ISO/IEC JTC1/SC21/WG3 N1020, February 1990.

BÉRUBÉ, J. 1990a. "Classification of IRDS Data and Services," ISO/JTC1/SC21/WG3/IRDS SEL23, May 1990.

BÉRUBÉ, J. 1990b. "Specification of IRDS Data and Services," ISO/JTC1/SC21/WG3/IRDS SEL24R1, May 1990.

BEYER, H. 1990. "A Comparative Analysis of Repository Approaches," ISO/JTC1/SC21/WG3/IRDS SEL20, May 1990.

CHEN, P. 1977. The Entity-Relationship Approach to Logical Database Design, QED Information Sciences, Wellesley, MA, 1977.

CONCEPTUAL SCHEMA 1987. "Concepts and Terminology for the Conceptual Schema and the Information Base," ISO TC97/SC5/WG3 TR9007.

EARL, A. 1990. "A Review of PCTE (Issue 3) according to the CASEE Reference Model," HPL-SEG-TN-90-04.

FRAMEWORK 1990. "Information Resource Dictionary System (IRDS) Framework," ISO/IEC JTC1/SC21 DIS 10027, 1990.

GATES, R. 1988. "Cross-reference to IRDS Facilities in ISO/ANSI Documents," ISO/IEC JTC1/SC21/WG3 N626, June 16, 1988.

GOLDFINE, A. and KONIG, P. 1988. A Technical Overview of the Information Resource Dictionary System (Second Edition), NBS IR 88-3700, National Bureau of Standards, Gaithersburg, MD, January 1988.

IRDS 1988. American National Standard X3.138-1988, Information Resources Dictionary System, American National Standards Institute, New York, 1988, (also adopted as FIPS PUB 156).

ISO 1992. "Information Resource Dictionary System (IRDS) Services Interface," ISO/IEC JTC1/SC21 IS 10728.

PCTE 1988. "PCTE Functional Specification," Issue 3, October 1988.

RM 1990. IBM Repository Manager Modeling Guide, GC26-4619-1

RMDM 1990. "Reference Model of Data Management," ISO/IEC JTC1/SC21/WG3 DIS 10032, May 1990.

SQL 1990. "DATABASE LANGUAGE SQL," ISO/JTC1/SC21/N5215--CD 9075.

WINSBERG, P. 1988. "Dictionary Standards: ANSI, ISO and IBM," InfoDB, 3, No 4 (Winter 1988/89).

NIST-114A
(REV. 3-90)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4904

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
SEPTEMBER 1992

4. TITLE AND SUBTITLE
A Compatibility Analysis of the ANSI and ISO IRDS Services Interfaces

5. AUTHOR(S)
Jean Berube, IDEgenic, Inc.
Alan Goldfine, Editor

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)
U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

Standardization bodies have produced specifications for two Information Resource Dictionary System (IRDS) services interfaces, one produced by ANSI X3H4, and the other by ISO/IEC JTC1/SC21/WG3. This report provides a formal comparison of the functionality and underlying data models specified by the two services interfaces. The focus is on the level of harmony and degree of interoperability that would be found between an ANSI environment and an ISO environment.

The report also provides an overview of three other published specifications that include dictionary/repository components: the IBM Repository Manager Interface, the ATIS IRDS proposal, and the ECMA PCTE standard.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

ATIS; Information Resource Dictionary System; IRDS; PCTE; Repository Manager; Services Interface.

13. AVAILABILITY

X	UNLIMITED
	FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
	ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
X	ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
103

15. PRICE
A06

ELECTRONIC FORM