NIST Special Publication 500-197

Computer Systems Technology

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology

Guide to Schema and Schema Extensibility

Bruce K. Rosen
Isabella des Fontaines

NIST Special Publication 500-197

Guide to Schema and Schema Extensibility

Bruce K. Rosen
Isabella des Fontaines *

Computer Systems Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

* U.S. Geological Survey
Water Resources Division

November 1991

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

Reports on Computer Systems Technology

The National Institute of Standards and Technology (NIST) has a unique responsibility for computer systems technology within the Federal government. NIST's Computer Systems Laboratory (CSL) develops standards and guidelines, provides technical assistance, and conducts research for computers and related telecommunications systems to achieve more effective utilization of Federal information technology resources. CSL's responsibilities include development of technical, management, physical, and administrative standards and guidelines for the cost-effective security and privacy of sensitive unclassified information processed in Federal computers. CSL assists agencies in developing security plans and in improving computer security awareness training. This Special Publication 500 series reports CSL research and guidelines to Federal agencies as well as to organizations in industry, government, and academia.

National Institute of Standards and Technology Special Publication 500-197
Natl. Inst. Stand. Technol. Spec. Publ. 500-197, 32 pages (Nov. 1991)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1991

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

PREFACE

The Computer Systems Laboratory (CSL) (formerly the Institute for Computer Sciences and Technology (ICST)) within the National Institute of Standards and Technology (NIST) has a mission under Public Law 89-306 (Brooks Act) to promote the "economic and efficient purchase, lease, maintenance, operation, and utilization of automatic data processing equipment by Federal departments and agencies." When a potentially valuable technology first appears, CSL may be involved in research and evaluation. Later on, standardization of the results of this research, in cooperation with voluntary industry standards bodies, may best serve Federal interests. Finally, CSL helps Federal agencies make practical use of existing standards and technology through consulting services and the development of supporting guidelines and software.

This report provides system managers, planners, data administrators, database administrators and potential dictionary or repository developers with a readable description of schemas and the concept of schema extensibility. Certain commercial software products and companies are identified in this report. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply the the products identified are necessarily the best available for the purpose as described.

ACKNOWLEDGMENTS

The technical work for this report was done through extensive review of published literature and hands-on use of the Information Resource Dictionary System prototype software developed at the National Institute of Standards and Technology (NIST). This document was done as a cooperative effort between personnel from NIST and the U.S. Geological Survey. The authors would like to thank Ms. Thomasin Kirkendall and Dr. Alan Goldfine for their technical and editorial contributions.

ABSTRACT

This guide was developed to assist both the casual user of Information Systems (IS) as well as ADP professionals in understanding the concepts behind database and data dictionary schemas and schema extensibility. It was developed in the context of its application and pertinence to the ANSI standard X3.138-1988, Information Resource Dictionary Systems (IRDS).

This guide begins with a set of definitions that provide an understanding of data dictionary and repository terminology. It then follows with a discussion of the IRDS. After establishing these basic definitions, the discussion of schema is initiated. The guide then takes the reader step by step through the concept of extensible schemas and how they are useful in performing the functions that should be accomplished in the areas of Information Resource Management (IRM) and Data Administration. The document also discusses the importance of schema extensibility to the exchange of information between Computer Aided Software Engineering (CASE) tools. Included is a discussion of the CASE tool to IRDS data exchange prototype that was developed at NIST.

The document concludes with a discussion of possible future impacts of schema extensibility on the development of standards in areas related to data dictionaries and repositories.

Keywords: CASE; Computer Aided Software Engineering; databases; data dictionary; DBMS; directory; encyclopedia; Information Resource Dictionary System; IRDS; life-cycle; repository; schema

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Purpose

This guide was developed to assist both the casual user of Information Systems (IS) as well as ADP professionals in understanding the concepts behind database and data dictionary schemas and schemas extensibility. It was developed in the context of its application and pertinence to the ANSI standard X3.138-1988, Information Resource Dictionary Systems (IRDS).

1.2 Scope

This guide is limited to the study of schema and schema extensibility as they apply to the Information Resources Management (IRM) and Data Administration of an organization or enterprise.

Chapter 2 provides an overview of schema concepts. It illustrates the use of schema building blocks in Information Systems modeling and briefly discusses schema use in an Information Resources Dictionary (IRD).

Chapter 3 describes database schema extensibility and the benefits of total extensibility. The IRDS standard and its schema extensibility are presented as the basis for future data dictionary implementations.

Chapter 4 examines the current applications of the IRDS standard for schema extensibility and where they are going in the 1990's.

2. SCHEMA CONCEPTS

2.1 Basic Terminology Definitions

Organizations have resources such as money, materials and people. In order to survive and thrive they must manage their resources efficiently. To manage its resources, an organization needs facts (data). For example; how many and what parts are needed to build a new component, when will they be needed, how many person hours must be allocated for assembly. The information required to manage the enterprise is derived from data. Thus, in order to manage resources effectively, the administration and processing of data is essential. However, only in recent years, with the introduction of ever increasing computer technology, has "data" come to be viewed as a resource in itself. Data is now viewed as sharing the common characteristics of all resources; those of cost and value. Based on these premises, we will now define a few basic terms.

Data - Facts about resources, also a resource itself.

Metadata - Since data is facts about resources, and is also a resource itself, we shall define metadata as a specific type of data that describes characteristics and facts about other data. Thus metadata is also a resource itself.

Database - An organized collection of data, the items of which are related in one form or another. A personnel database, for example, contains facts about the employees of an organization such as birth date, number of children, etc. The processing of a database can be either automated or manual.

Data Dictionary - A specialized database of information which describes the characteristics of the data used to design, monitor, document, protect, and control the information resources of the enterprise. A data dictionary can be described as a **metadatabase.**

Database Management System (DBMS) - A set of software components that provide for the automated creation, update, access, maintenance, and control of one or more databases. A DBMS also facilitates the access to the data by application software.

Repository - An expanded form of a data dictionary that may also include information relating to the planning, analysis, design, construction, and maintenance of systems. It may also include other facts such as the rules and policies that govern the enterprise.

2.2 What is a Schema?

Schema - Webster defined schema as a plan, an outline or a diagram. Schema, in the context of data dictionary and database is the plan(s) which organizes and structures the data in such a way that it models the desired flow of information. A schema is an essential component of any DBMS. In other words, it establishes a map or navigation path on top of the data by providing a common description of all components and how they interrelate. Accordingly, a documented schema description can also be thought of as metadata. A sub-schema is a smaller part of the total schema. It presents a partial perspective of the database or data dictionary. It is used to support a user group's view of the organization's resources.

Metaschema - Whereas the schema organizes and structures the data, the metaschema organizes and structures the metadata.

"Schema" is most commonly used in association with DBMS, yet a schema can be a multi-purpose component that is instrumental in defining, describing and analyzing an enterprise's resources in a formal manner in order to build the enterprise business model.

2.3 The ERA Model

All schema examples in this document are based on the entity relationship attribute (ERA) information model. This model was first introduced by Peter Chen [CHEN79] in 1979. Since that time it has undergone continued development [TEOR86]. The model supports the analysis of entities, relationships, and attributes.

Entities are used to model resources and correspond to nouns or objects in the English language. They can represent a diversity of items such as STRATEGIC-PLANS, SUPPLIER-NAME, ORDER-NUMBER, etc. An entity does not stand on its own. It belongs to a group of entities sharing common characteristics, called an entity-type. An individual entity can have one or more instances. Instances are data values of an entity and are at the lowest level of available information. Instances are data that exist in one or more databases, while entity-types and entities are metadata. Table 1 portrays some examples of entity structures.

Entities are related to one another through relationships. Relationships correspond to verbs. For example, to show how the DIVISION-GOALS are set by the enterprise, we would define the relationship SET-BY and express it in the following way; GOALS SET-BY ORGANIZATION.

Attributes corresponds to adjectives or adverbs. They are used to describe entities or relationships. For example, LENGTH can be an attribute of the entity ORDER-NUMBER describing the length of the data element ORDER-NUMBER. Attributes can take specific values. In this case LENGTH has a value of 6. Relationships can also have attributes. For example, an attribute of the relationship SET-BY might be the FREQUENCY of the goal setting performed by the organization. Figure 1 shows an example of a schema using the ERA model.

Table 1. Examples of Entity Structures

Entity-Type	Entity	Instance
GOALS	DIVISION-GOALS	Division A Division B Division C
ORGANIZATION-PLAN	STRATEGIC-PLAN	Plan FY90
ELEMENT	SUPPLIER-NAME ORDER-NUMBER	GM Corp. 123456 654321

2.4 The Three Schema Architecture

To succeed in fully defining the schema concept, let us take a closer look at the IRM architecture.

In 1976 the Standard Planning and Requirements Committee of the American National Standards Committee on Computers and Information Processing (ANSI/SPARC) introduced what has come to be known as the "Three Schema Architecture." This principle takes full advantage of the "schema concept" by showing that the IRM of any enterprise can be documented and modeled through the definition and representation of three levels of schemas; conceptual, external and physical. Figure 2 portrays a modified version of the Three Schema Architecture as presented in [ROSE89].

The conceptual schema, also referred to as the logical view, is the integrated view of all the information resource models developed to describe the enterprise's information architecture. It has an organization-wide scope and is stable with respect to technology and applications. It portrays the relationships and attributes between all resources as represented by entities. This representation should be independent of any specific software or hardware.

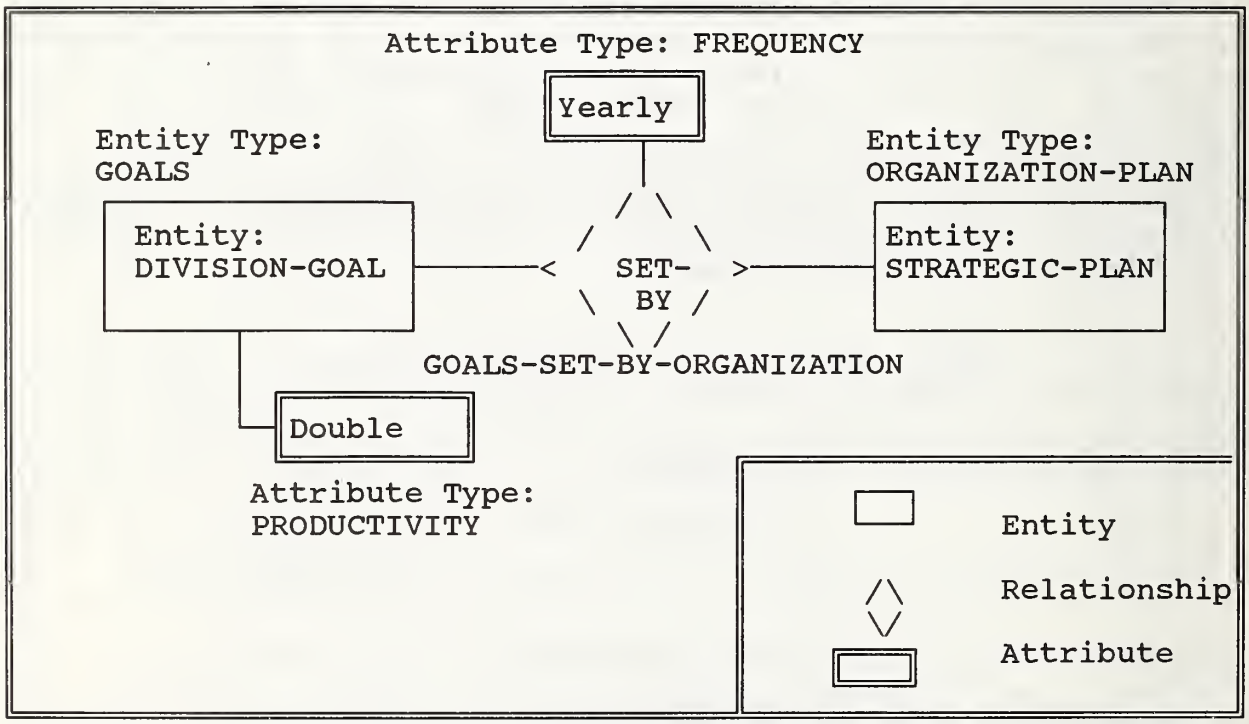

Figure 1. ERA Model of a Division Strategic Plan Sub-Schema.

The external schemas are sometimes called "user views." They represent portions of the information resources as they pertain to a specific group of users, or an application, like a phase of the System Development Life Cycle (SDLC). Figure 1 could, for example, be considered an external schema representing the strategic planning User View of a specific Division Head. User views are not exclusive; some may overlap as External Schemas 1 and 2 do in figure 2. An external schema has a narrower scope than the conceptual schema. It may change in time as organizational or user's needs change. Yet, it is relatively stable with respect to software or hardware. Individual external schemas are sometimes called sub-schemas, since each external schema represents a user specific view of some small part of the conceptual schema.

Finally, the internal schema, also referred to as the physical schema, provides a model of the physical storage of the data

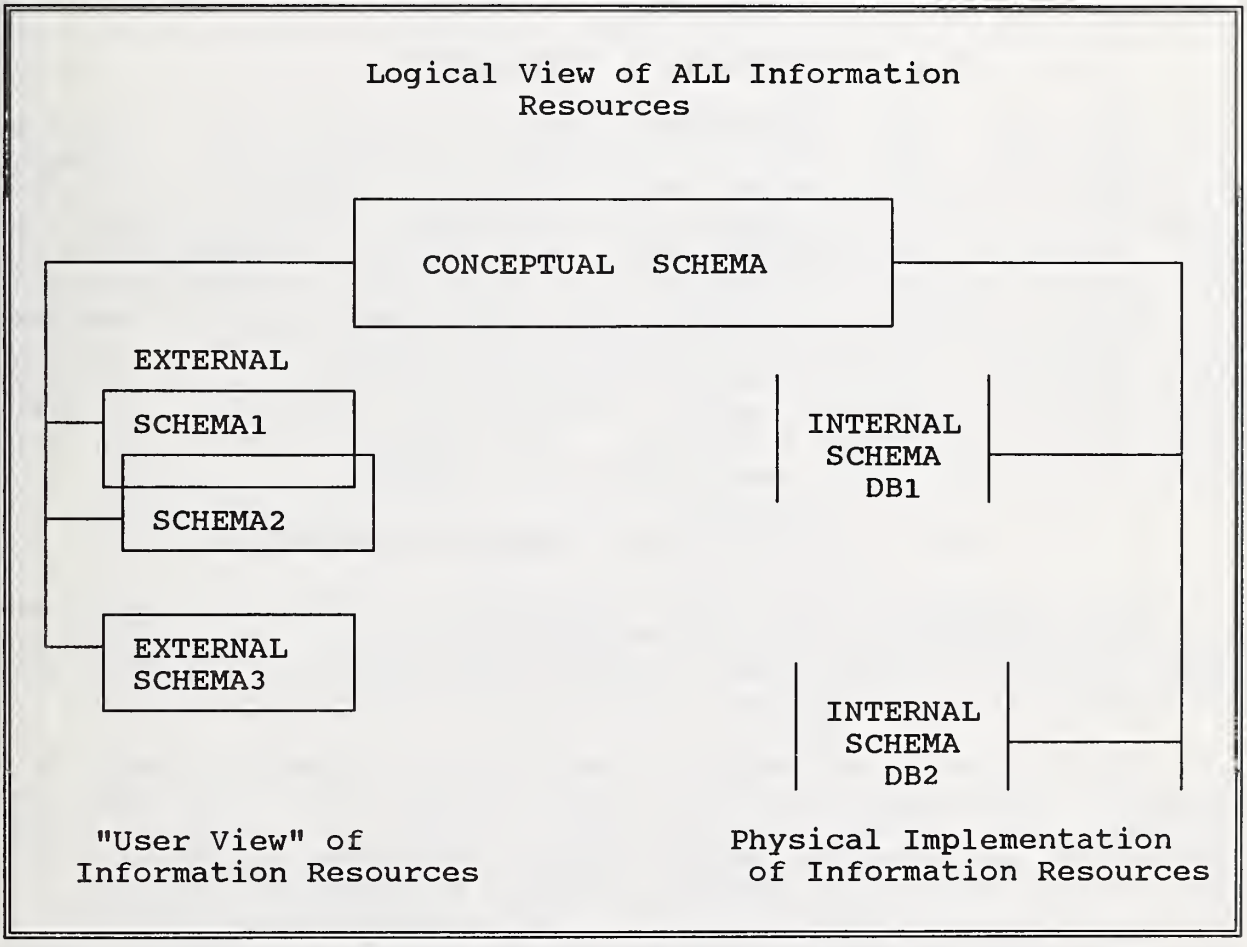

Figure 2. Modified Three Schema Architecture.

resources of the enterprise. The storage media may be automated (a disk file), or manual (a file cabinet). It may involve single or multiple databases of various types, including multiple vendor DBMSs. It may also involve geographically dispersed databases and DBMSs. Internal schemas are dynamic and change with workloads, technology and the financial status of the enterprise. The internal schema is completely dependent on hardware and software. For an extended discussion on the concept of the "Three Schema Architecture" refer to [JARD77].

2.4.1 Schema Architecture of the IRDS

In this section we will examine the Information Resource Dictionary System (IRDS) schema architecture. It is through the use of an IRDS that an organization can describe and analyze its information resources based on the concepts embodied in the Three Schema Architecture. For more information on the application of the IRDS to the Three Schema Architecture the reader should refer to [ROSE89]. To begin, a short history of the IRDS standard is provided.

2.4.1.1 Historical Background

The IRDS is a standard defining a set of software specifications for the definition, description, access, maintenance, and control of data dictionary systems. The IRDS standard was developed by the American National Standards Institute (ANSI) Technical Subcommittee X3H4. The initial document on which the current standard was based was developed at the National Institute of Standards and Technology of the Department of Commerce (then known as the National Bureau of Standards). It was approved and published in 1988 as ANSI Standard X3.138-1988. It was also approved and issued as Federal Information Processing Standard Publication 156 (FIPS PUB 156). For further details on the background of the IRDS, see [GOLD88] or [ROSE89].

2.4.1.2 IRDS Schema Layered Architecture

From this point on in this document, we will only refer to automated systems when we discuss data dictionary. The IRDS is an expanded data dictionary that includes the support of metadata for all System Life Cycle phases of any system in an organization.

The IRDS is a software tool which can be used to describe, document, protect, control, and enhance the use of an organization's information resources as modeled in one or more Information Resource Dictionary (IRD). The architecture of the IRDS and its schema layers are portrayed in figure 3.

The concepts behind the IRDS architecture can be described in terms of four layers. Each of these layers is more fully discussed in the following paragraphs of this section. The IRDS standard

dictates the metaschema that makes up Layer 1 of any IRDS. Layers 2 and 3 are controlled by the IRDS administrator(s). One IRDS software package may manage more than one IRD schema (Layer 2) as well as store the metadata for more than one IRD. An IRD is a specific collection of entities, relationships and attributes used by an organization to model a specific information environment.

An IRDS software package could be developed as either a passive dictionary or, through use of the services interface module, as an active dictionary. If an IRDS is implemented in the

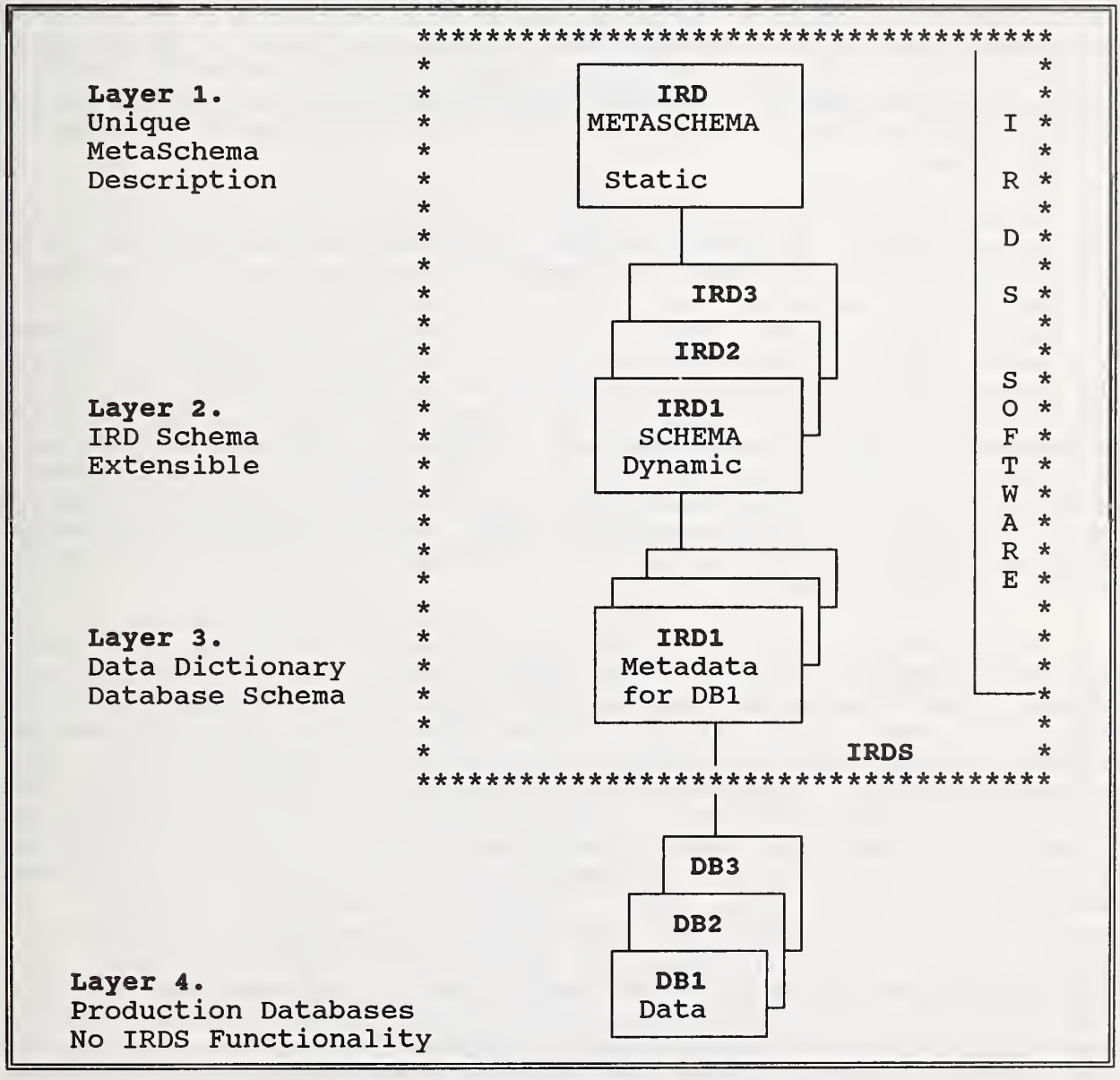

Figure 3. IRDS Schema Layers.

active mode, the organization could ensure that data at the production layer could only be created, managed, accessed and controlled through the appropriate IRD. In that case the IRD is directly utilized by the DBMS in performing such functions as editing data elements, checking access rights, and insuring data integrity. For a further discussion of active versus passive dictionaries refer to [WERT89] or [MEAD90].

Layer 1 is fully defined by the IRDS standard and is based on the ERA model. This level describes the type of objects that can be defined at Layer 2. Thus Layer 1 can be viewed as a metaschema since it represents a set of concepts and terminology for expressing the schemas that Layer 2 will design. It is the foundation upon which all the other layers are built. This layer is static and contains a set of unique descriptors such as ENTITY-TYPE, RELATIONSHIP-TYPE, RELATIONSHIP-CLASS-TYPE, ATTRIBUTE-TYPE, ATTRIBUTE-GROUP-TYPE. These descriptors are sometimes called Meta-entity-types.

Layer 2 defines the IRD schema of an IRD. To be compliant to the standard, a Minimal Schema must be available with any implementation of the IRDS. The IRDS standard supports full schema extensibility, meaning that schema structures can be added and modified to satisfy the user need for information modeling. Schema extensibility is further discussed in section 3. The IRDS standard also specifies an optional starter set schema called the Basic Functional Schema in order to support intra- and inter-organization resources management. This Basic Functional Schema is defined in IRDS Optional Module 1. The Basic Functional Schema is required for any software implementation of the IRDS standard that is to be in conformance with FIPS 156. For further details on the Minimal Schema and Basic Functional Schema of the IRDS see [ANSI88] and [GOLD88].

An organization can choose to create more than one IRD under the IRDS umbrella. Each of these IRD's could have their own unique schema. For example, an organization might wish to create multiple IRD's that are organized along either application system lines or organizational structure lines. It is also possible that it may not be practical to represent an entire system lifecycle for one extremely large application in a single IRD. In this situation an organization could use the following scenario. It could subdivide the system lifecycle in 4 super phases: (1) the early system development phase which would include strategic planning, and requirements definition, (2) the intermediate system development phase which would include the functional specifications, the logical database design and data and function integration, (3) the late system development phases which would include system design, physical database design and system implementation, and finally (4) the system lifecycle operational phases which would include system operations and maintenance. Two IRD's could be used to divide the phases: one for super phases (1) and (2), and the second for super

phases (3) and (4). For a further discussion on lifecycle approach to IRD applications see [LAW88].

Layer 3 describes the actual environment being modeled. This is the level where information models and the descriptions of production data, often referred to as "real world data," are maintained. If an IRD schema at Layer 2 supports multiple databases, the schemas of these working databases are defined by Layer 3.

Finally Layer 4 is the production environment. It is the layer where actual production data is stored. The production data of Layer 4 should be described and defined in Layer 3 of an IRDS, but otherwise Layer 4 is outside of the realm of the IRDS standard. The IRDS could communicate with Layer 4, but the level of interface and control between the IRDS and Layer 4 is dependent on the amount of activity or passivity built into the IRDS package.

Table 2 presents a set of examples of the type of content that would exist in the different layers of an IRDS. Included in the table is an indication that the actual IRDS ends after layer 3 and that application system data begins at layer 4.

3. SCHEMA EXTENSIBILITY

Perhaps the most important feature of the IRDS standard is the required ability to extend the IRD schema definition at any time. In this section we will examine the concept of schema extensibility, see why it is a critical feature in any data dictionary application and use the IRDS command language to show examples of schema extensibility.

3.1 What is Schema Extensibility?

Extensibility is the ability to stretch and enlarge. For a Data Dictionary or repository application, schema extensibility is the capability to add to, delete from, and modify the structure of the Data Dictionary or repository database. For example, schema extensibility would entail the ability to add, delete or modify entity-types, relationship-types, relationship-class-types, attribute-types, and attribute-group-types or any other meta-entities or meta-relationships at Layer 2 of the IRDS architecture.

3.2 Why Would We Want to Extend the Schema?

Only "change" is a known constant. Organizational needs vary with changes in organization mission and scope. Requirements vary with time and the availability of resources. Design environments are characterized by continual change. Designs are constantly being revisited to fix bugs, accept changing user specifications,

and convert to new hardware platforms. Since the schema is the template of a design, as such it is likely to require change.

In the past, most commercial data dictionaries have offered a fixed schema architecture, often similar to the optional Basic

Table 2. Examples of IRDS Content by Layers

	Meta-Entity-Types		
Layer 1	ENTITY-TYPE	RELATIONSHIP -TYPE	ATTRIBUTE-TYPE ATTRIBUTE-GROUP -TYPE
Layer 2	(Entity-Types)	(Relationship-Type)	(Attribute-Type & Attribute-Group-Type
	ELEMENT RECORD GOALS ORGANIZATION-PLAN	RECORD-CONTAINS -ELEMENTS GOALS-SET-BY-ORGANIZATION	LENGTH PRODUCTIVITY FREQUENCY ALLOWABLE-RANGE
Layer 3	(Entities)	(Relationships)	(Attributes)
	PART-NO QTY-ON-HAND PART-REC DIVISION-GOALS STRATEGIC-PLAN	PART-REC-CONTAINS-QUANTITY-ON-HAND DIVISION-GOALS--SET-BY-STRATEGIC-PLAN	6 (digits of PART-NO) 000001 (Lo-val) 999999 (Hi-val) Yearly Double
	IRDS ends, Application system data begins		
Layer 4	(Instances of Entities)	(Instances of Relationships)	
	123456 200 PART-REC of 123456 Division A Plan FY90	PART-REC of 123456 contains 200 on hand	Attributes are descriptive. They do not appear as instances in production DBs.

Functional Schema module of the IRDS standard. For a full definition of the Minimal and Basic Functional Schema of the IRDS standard, see [GOLD88]. A fixed schema, if properly selected, may satisfy most initial modeling needs for entity-types, relationship-types, relationship-class-types, attribute-types, and attribute-group-types. However, a fixed schema does have two major drawbacks; (1) it only addresses certain phases of the System Life Cycle, and (2) it is not well suited to model the ever changing complex systems we face today.

3.2.1 Schema Extensibility in the System Life Cycle

Fixed schemas, especially those available in data dictionaries, usually address the system design phase of the System Life Cycle and/or the operational phase of the system. However, they do not usually portray the earlier phases, such as strategic planning and requirement analysis, which are needed for the data dictionary to function in the role of a repository.

The commonly predefined entity-types in most dictionaries are similar to the Basic Functional Schema's USER, SYSTEM, PROGRAM, MODULE, FILE, DOCUMENT, RECORD, ELEMENT. These entity-types are best suited for system design or operation but cannot be used to express GOALS, REQUIREMENTS, OBJECTIVES, PROBLEMS, CONSTRAINTS, etc. In most early commercial data dictionaries the entity-types in figure 1 would not have been available, and no new entity-types could be added.

For example, suppose that we want to show in figure 1, that an organizational function, such as purchasing new equipment, will help the enterprise achieve its goal of doubling productivity in fiscal year 1990. We may want to create a new entity-type, FUNCTIONS, a new entity, PURCHASING, and a new relationship-type, SUPPORT, to expand the Division Strategic Planning sub-schema. Figure 4 illustrates graphically how the sub-schema in figure 1 has been extended. Without schema extensibility the dictionary administrator cannot add any new entity-types or relationship-types and thus cannot add the specific entities and relationships needed to satisfy the new requirement. Once schema extensibility is available then the data dictionary crosses the boundary into the area now referred to as "repository."

3.2.2 Schema Extensibility for Complex Systems Modeling

There is another reason why extensibility of schemas is necessary. Fixed schemas may provide an excellent data dictionary support for specific applications, such as financial management. However, a schema that is oriented toward a specific application, such as financial management, does not lend itself to the modeling of other types of applications such as scientific applications or other complex applications which manipulate large data objects and capture change of data over time. These other complex applications

can be found in visualization systems like computer assisted design (CAD) and chemical concentration modeling over time. This problem also exists in Image Processing systems like in Geographic Information Systems (GIS) and Land Management Systems (LMS) where new entity-types are needed to define objects such as geographic layers.

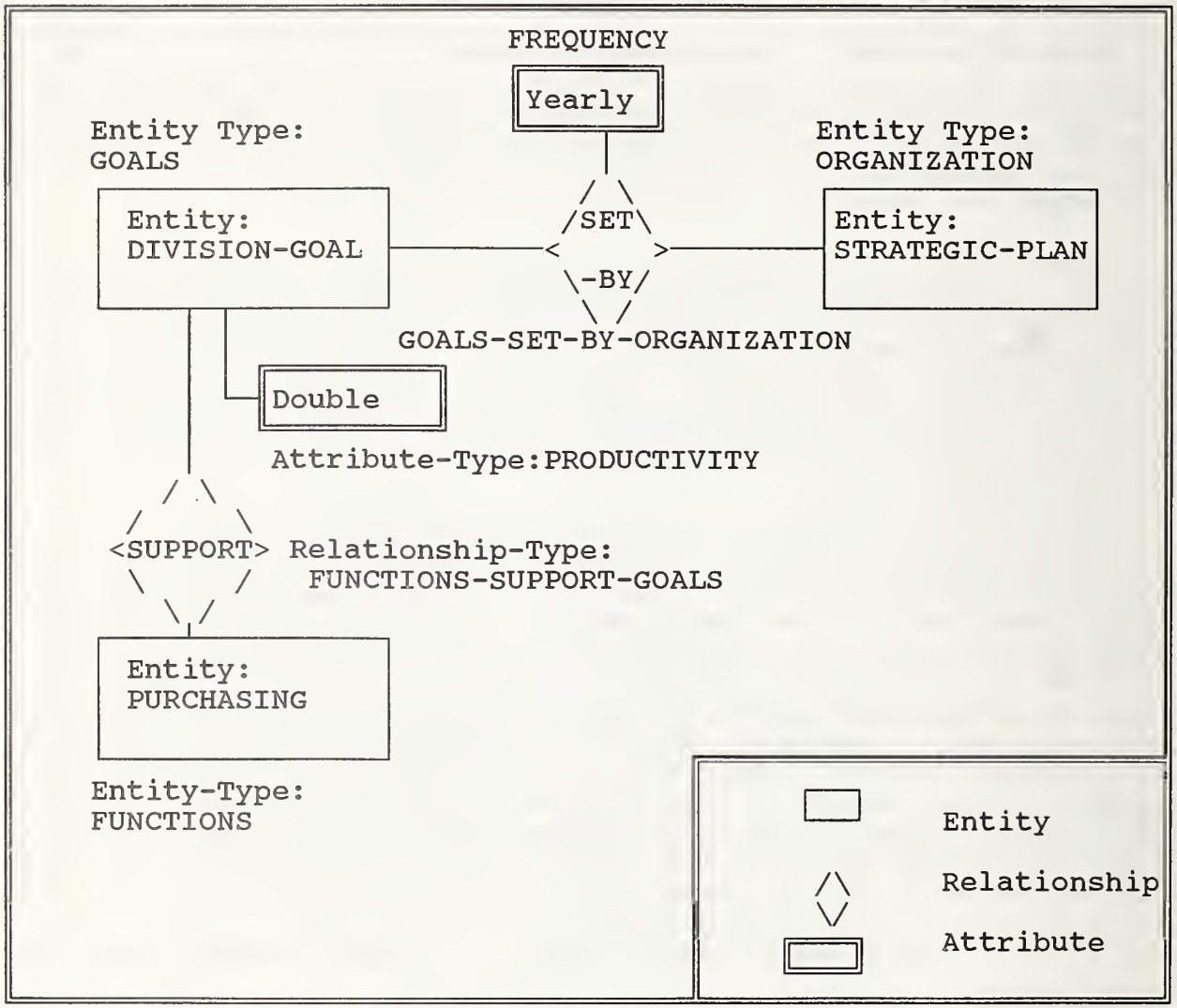

Figure 4. ERA Model of an Extended Division Strategic Planning Sub-Schema.

3.2.3 Schema Extensibility to Model the Entire Organization

A data dictionary/repository must serve as the communication tool between the major elements and forces which shape, manage, monitor, and use the organization (fig. 5). USERS need documentation tools and transparent access to distributed systems. CORPORATE MANAGEMENT needs integrated planning tools and project management tools. The DATA PROCESSING department must support all software life cycles and needs configuration management facilities. DATA PROCESSING must also support source and object libraries as well as data element standardization management systems. Finally, the AUDITORS will want administration and reporting tools.

The data dictionary/repository must know no boundaries since it must cross them all. Therefore the conceptual schema is never concluded. It is a dynamic process which must follow the organization's growth and evolution.

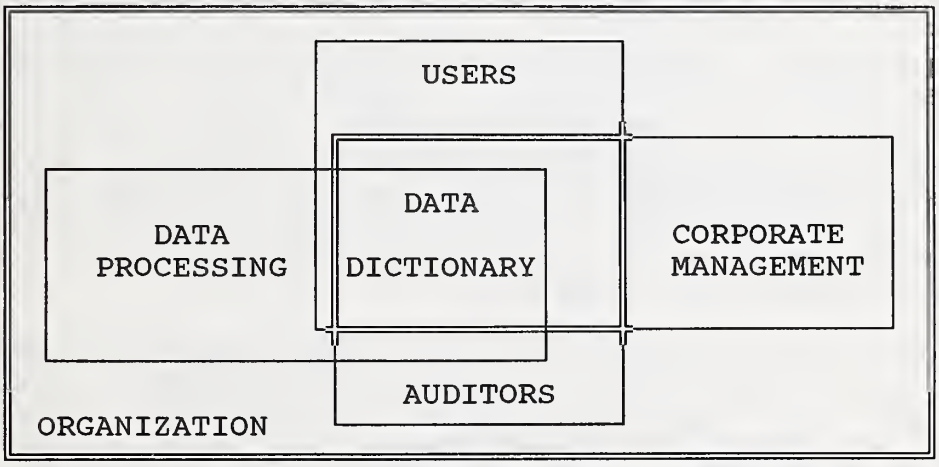

Figure 5. Dictionary as Communication Tool.

3.3 Background on the IRDS and Schema Extensibility

As stated in X3.138 [ANSI88], the IRDS provides many things for many different individuals and purposes. Applications of an IRDS can include the usage or support of:

- Documentation tools
- Software life cycle and project management tools
- Data element standardization and management systems
- Planning tools
- Administration tools for database and data

- Source and object library management systems
- Configuration management facilities
- Tools for supporting distributed processing
- Computer Aided Software Engineering (CASE) tools

None of these areas of usage are static in nature. The development committee for the IRDS standard recognized that it was impossible to fully anticipate future requirements. They therefore sought to make the IRDS standard flexible enough to satisfy recognized current requirements and accommodate future requirements. The standard itself does not address all of the requirements for each of the usage areas mentioned above. It does, however, provide a foundation upon which any implementor of the standard can build in order to meet part or all of the requirements needed to perform the above listed applications. This capability is possible partly because of the schema extensibility feature of the IRDS standard.

3.4 IRDS Implementation of Schema Extensibility

The FIPS IRDS mandates that the implementor provide Layer 1 of the architecture, the minimal schema, the IRD Schema maintenance functionality, the IRD maintenance functionality, and the Basic Functional Schema. The Basic Functional Schema includes the entity-types used in most business applications: USER, SYSTEM, PROGRAM, MODULE, FILE, DOCUMENT, RECORD, ELEMENT. For further details on the IRDS standard see [GOLD88] or [ANSI88]. Using the IRD Schema maintenance functionality a user of the IRDS can create an individual schema. For additional details on the attribute-types, attribute-group-types, relationship-types, and relationship-class-types available see [GOLD88].

To create a new IRD application the user always has access to the Basic Functional Schema and the IRD Schema maintenance functionality. If the new application does not exceed the structure of the Basic Functional Schema, the user is then ready to enter the metadata for the specific application. If the Basic Functional Schema does not meet the modeling needs, the user can **extend** the schema by adding, modifying, or deleting the entity-types, attribute-types, attribute-group-types, and relationship-types required by the application. Maintenance of the IRD schema is done at Layer 2 of the IRDS architecture. Maintenance of the IRD, describing information models and production information assets, is done at Layer 3.

The IRDS standard specifies the format and content of a command language that can be used to perform maintenance on the IRD schema and the IRD. Both maintenance procedures use a similar syntax.

We will use the sub-schema for Division Strategic Planning Model, as presented in figure 4, to illustrate how to use the IRDS

command language to extend the Basic Functional Schema. First let us identify which elements must be added to the IRD schema versus those to be added to the IRD. None of the needed entity-types, relationship-types, attribute-types and relationship-class-types are part of the Basic Functional Schema. Table 3 displays the elements of the Division Strategic Planning Model which could not be found in the Basic Schema of the IRDS standard. Again, let us clarify the meaning of the two columns. The left hand side column identifies the meta entities which must be added to the IRDS schema. In other words, it expands the IRD-schema which structures the metadata of an IRD application. The right hand side column identifies instances of these meta-entities, that is, the entities, relationships and attributes which must be added to this particular IRD application. In other words, it expands the description of the schema of the production database.

Table 3. Missing Elements of Division Strategic Planning Model

IRD SCHEMA	IRD
Metaschema of the metadata of an IRD	Production Database Schema
Entity Type: FUNCTIONS GOALS ORGANIZATION-PLAN	**Entity:** PURCHASING DIVISION-GOAL STRATEGIC-PLAN
Relationship-Class-Type: SET-BY SUPPORT	
Relationship-Type: FUNCTIONS-SUPPORT-GOALS GOALS-SET-BY-ORGANIZATION	**Relationship:** PURCHASING-SUPPORT-DIVISION-GOALS DIVISION-GOALS-SET-BY-STRATEGIC-PLAN
Attribute-Type: FREQUENCY PRODUCTIVITY	**Attribute:** Yearly Double

3.4.1 Example of IRD Schema Extensibility

The IRDS standard specifies two user interfaces: the Command Language Interface and the Panel Interface. To be in conformance with the standard an IRDS implementation may include either or both. The Command Language supports the user's interaction in both batch and interactive modes and assumes a certain level of computer literacy and the acquisition of an extensive syntax knowledge. The Panel Interface is more "user friendly." It can be implemented as a menu driven system that does some "hand holding". The example below illustrates the use of the IRDS command language as implemented in the National Institute of Standards and Technology (NIST) IRDS Command Language prototype. The example demonstrates how an IRD schema is extended using the NIST Command Language prototype.

To extend the IRD schema (Layer 2) we will first examine the left hand side of table 3. Since all meta-entities in the left column except for FREQUENCY are new, we will use the **"ADD"** command. The bold words are the variables being added to extend the schema.

Adding New Entity-Types

```
ADD meta-entity FUNCTIONS meta-entity-type = Entity-Type;

ADD meta-entity GOALS meta-entity-type = Entity-Type;
ADD meta-entity ORGANIZATION-PLAN
     meta-entity-type = Entity-Type;
```

Adding New Relationship-Class-Type

```
ADD meta-entity SET-BY
     meta-entity-type = Relationship-Class-Type;
ADD meta-entity SUPPORT
     meta-entity-type = Relationship-Class-type;
```

Adding Relationship-Types

```
ADD meta-entity FUNCTIONS-SUPPORT-GOALS
    meta-entity-type = Relationship-Type;

ADD meta-entity GOALS-SET-BY-ORGANIZATION
    meta-entity-type = Relationship-Type;
```

Associating the relationship-types with their component entity-types

```
ADD meta-relationship FUNCTIONS-SUPPORT-GOALS
     Connects FUNCTIONS  Position = 1;
```

```
ADD meta-relationship FUNCTIONS-SUPPORT-GOALS
     Connects GOALS  Position = 2;

ADD meta-relationship GOALS-SET-BY-ORGANIZATION
     Connects GOALS  Position = 1;

ADD meta-relationship GOALS-SET-BY-ORGANIZATION
     Connects ORGANIZATION  Position = 2;
```

Adding Attribute-Types

```
ADD meta-entity PRODUCTIVITY meta-entity-type=
                              Attribute-Type;
```

Associating Relationship-Type with Relationship-Class-Type

```
ADD meta-relationship FUNCTIONS-SUPPORT-GOALS
                      Member-of SUPPORT;


ADD meta-relationship GOALS-SET-BY-ORGANIZATION
                      Member-of SET-BY;
```

Associating Attribute-Types with Entity-Type and Relationship-Type

```
ADD meta-relationship GOALS-SET-BY-ORGANIZATION
     Contains FREQUENCY;

ADD meta-relationship GOALS Contains PRODUCTIVITY;
```

3.4.2 Example of Populating an IRD

Once an IRD schema is established, it can be populated with metadata. If the IRDS implementation defines an active IRD, then any access to the production database would have to go through the IRD. In such a case, by adding to the IRD, the user would also be extending the production database's conceptual and physical schemas. How often in the system life cycle of an organization has a production database schema remained unchanged? Most large applications are usually designed and implemented with an expected production life exceeding a decade or more. The chance that the database schema defined during the System design phase will still meet current requirements 10 years or more after its design is highly unlikely.

The Command Language syntax used to build an IRD (Layer 3) is very similar to the one used to extend the IRD Schema. To populate the IRD we will examine the right hand side of the table 3. Again,

the entities, attributes and relationships are all new. We will therefore use the **"ADD"** command. The bold words are the variables to be added to populate the IRD description of the production database schema.

Adding Entities

ADD entity **DIVISION-GOAL** entity-type = **GOALS**
with attributes **PRODUCTIVITY = "Double";**

ADD entity **STRATEGIC-PLAN** entity-type = **ORGANIZATION-PLAN;**

ADD entity **PURCHASING** entity-type = **FUNCTION;**

Adding New Relationships

ADD relationship **DIVISION-GOALS SET-BY STRATEGIC-PLAN**
with attributes **FREQUENCY = "Yearly";**

ADD relationship **PURCHASING SUPPORT DIVISION-GOAL;**

3.4.3 Drawbacks of Schema Extensibility

While schema extensibility is an extremely important and powerful capability, it has certain drawbacks. First and foremost is that it requires the administrator of the dictionary/repository to become more knowledgeable about the organization in order to provide schema extensions that reflect the "real world." Also, since schemas can be modified, some form of organizational control must be put in place or the proliferation of different schemas will hinder interoperability and reuse of information resources. Finally, as schemas change over time, care must be taken to ensure that the integrity of dictionary/repository information in relation to older, still operational organizational information, is not lost. Thus it is clear that the price to be paid for the power of schema extensibility is the need for additional planning and control.

4. FUTURE SCHEMA EXTENSIBILITY APPLICATIONS

This final section looks at where the IRDS standard is going in the 1990's and the impact of CASE tools.

4.1 The Future of the IRDS Schema Extensibility

In this document we have defined and described what schema extensibility is and shown how the IRDS standard has implemented it. Only recently has "full" extensibility, as defined previously, become available commercially. Although the IRDS standard supports schema modifications, once a schema has been defined and the IRD

has been populated with the metadata, schema redesign, even small modifications, could be time consuming.

If organizations are to be encouraged to create new schemas and/or extend existing schemas to better support the type of descriptive information most appropriate to the application area, an effort must be made to facilitate the extension of existing schemas even after they have been populated. Possibly the emerging standard on IRDS Export/Import may provide some initial assistance in the integration of different schema models.

4.2 The Role and Influence of Computer Aided Software Engineering (CASE)

In May 1989, John Hagedorn and Christopher Durney of Chartway Technologies presented a paper at the 10th Annual Conference on Application of Computer Aided Software Engineering Tools [HAGE89]. They made some general observations about industry directions in the 1990's including: (1) the software industry will continue to evolve toward automation of the full life cycle; (2) standard interfaces between CASE tools will be adopted and the development of full function repositories (dictionaries) will provide the key to full CASE integration; (3) system life cycle costs will continue to shift from development to design activities; (4) no one software development methodology will dominate the market.

Each of these four points makes a case for schema extensibility. First, as the software industry continues to evolve toward automation of the full life cycle, there will be an increasing need to add new meta-entities to the IRD. As the CASE tool marketplace has expanded users of these tools have begun to demand that schema extensibility be easily supported without major efforts in terms of cost and manpower (download/reload).

Second, full function repositories means that multiple CASE products will be used to model the entire SDLC of the organization through an Integrated CASE (I-CASE) environment. These tools will have to span the early phases of Strategic Planning and Requirements Definition, the intermediate phases of Functional Specifications and Logical Design and finally the late phases of System Design and Implementation. These tools cannot use a proprietary or closed architecture. They must be integrated, working together not only conceptually but physically. In order to achieve such an integration, schemas will need to be modified, if only to establish linkage between the various tools. Schema extension will also be necessary each time a new CASE tool is added to the SDLC. Full function repositories will need full schema extensibility to make CASE tools truly reach their full potential.

Third, as costs and emphasis continue to shift from development to design activities, the need to be able to define, change and modify meta-entities at layer 2 will grow

proportionally. No vendor can implement a fixed IRD schema which will support all design activities of all organizations.

Finally, observation suggests that the choice of a methodology to support software engineering will remain an internal decision. Some organizations will opt for an internally developed methodology whereas others will use one that is vendor-based. Each methodology will choose, use and support a different set of entity-types, relationship-types and attribute-types. Only full schema extensibility can provide access and allow for an unlimited number of methodologies. This is especially critical in large organizations where various elements of the organization may be utilizing different methodologies. Only full schema extensibility would allow the results of these different methodologies to be integrated.

4.3 Prototype CASE to IRDS Information Transfer

In October of 1990 the National Institute of Standards and Technology completed a project done on behalf of the Defense Logistics Agency (DLA) to research communication between Data Dictionaries and CASE tools through the use of applicable standards. One of the tasks under this project was to create a method of transferring data from a CASE tool to an IRDS using the IRDS Export/Import File Format Standard which was then under development in ANSI Technical Subcommittee X3H4. The primary factor which allowed this transfer to be accomplished was the schema extensibility of the IRDS.

The first step in the accomplishment of the transfer was to establish an Information Resource Dictionary (IRD) that had the appropriate schema to which the fixed schema of the subject CASE tool, KnowledgeWare's Information Engineering Workbench (IEW), could then be mapped. The use of this particular product in the completion of this project does not imply any recommendation or endorsement by NIST or DLA for the product or the company.

A program was then developed that accepted as input a set of IEW export files, and then generated as output an equivalent file of information structured in accordance with the IRDS Export/Import File Format. The Export/Import structured file was then entered into the IRD through use of the IRDS Import command functionality. No attempt was made during this project to transfer information back to IEW from the IRDS since IEW does not have an extensible schema. It should be noted that while the Export/Import File Format Standard was still in the draft stage at the time this project was completed, no extensions to the format were necessary to perform the transfer.

5. CONCLUSIONS

This brief guide has attempted to frame the concepts behind schema extensibility in the context of both data dictionaries and the production databases and information models described by them. One conclusion has emerged from this study; it is the need for schema extensibility throughout the entire SDLC. In the past, schema extensibility was associated with the production database and addressed only at system design time. This is no longer the case. The ability to extend the schema has become a necessity of all phases of the SDLC.

In the early 1970's a schema was usually associated only with an individual program. By the late 1970's a schema could represent an entire application area such as account receivables, payroll, etc. In the 1980's the concept of schema expanded to an entire business process such as accounting, marketing, design. In the 1990's it is expected that the "conceptual schema" will expand even further until it serves as a way of modeling the entire organization. Daniel S. Appleton, Chairman and CEO of D. Appleton Company Inc. gives this definition of the "conceptual schema": it is the whole infrastructure of internally consistent rules that control the business' view of itself and its environment.

Hundreds, even thousands of business rules influence the storage of facts in the information structure and as a result control what information can be generated. Any business activity change must be validated against the existing business rules to evaluate its impact and cost. It is impossible for a single individual to know all these rules. The fact that most people see only a subset (sub-schema) of these rules results in a reliance on ADP information systems. As business activities change and new or modified rules are introduced, they must be validated against the total business rules or conceptual schema. Basically new or modified rules must be developed as extensions to the total conceptual schema of the organization. Full schema extensibility, throughout the SDLC, is no longer a "nice" thing to have. The need to model the rules, and support the information processing and management of organizations, makes it a necessity.

The 1990's will see the expansion of software engineering supported by sophisticated Integrated CASE tools. These tools will be available for all stages of the SDLC and should access a common dictionary or repository. This dictionary or repository should support the IRDS standard for schema extensibility and will eventually go one step further to directly support production database schema extensibility.

REFERENCES

[ANSI88] ANSI, American National Standard X3.138-1988, Information Resource Dictionary System, American National Standards Institute, New York, 1988.

[CHEN79] Chen, Peter P., editor, Proceedings of the International Conference on Entity-Relationship Approach to Systems Analysis and Design, December 10-12, 1979, North-Holland Publishing Co., Amsterdam, The Netherlands, 1980.

[GOLD88] Goldfine, Alan, and Patricia Konig, A Technical Overview of the Information Resource Dictionary System (Second edition), NBS 88-3700, National Bureau of Standards, Gaithersburg, MD, January 1988.

[HAGE89] Hagedorn, James R., and Christopher P. Durney, The Context for CASE: Building a Productive Development Environment, Chartway Technologies, Inc., May 1989.

[JARD77] Jardine, Donald A., editor, The ANSI/SPARC DBMS Model Proceedings of the Second SHARE Working Conference on Data Base Management Systems, North-Holland Publishing Company (US distributor, American Elsevier Publishing Co. Inc.), 1977.

[LAW88] Law, Margaret Henderson, Guide to Information Dictionary System Applications: General Concepts and Strategic Systems Planning, NBS Special Publication 500-152, National Bureau of Standards, Gaithersburg, MD, April 1988.

[MEAD90] Meador, Jo G., "Making the Case for Integrated Systems: Evaluating Dictionary Technology," Data Resource Management (DRM), Vol. 1, No. 2, Spring 1990, pp. 5-12.

[ROSE89] Rosen, Bruce K. and Margaret H. Law, Guide to Data Administration, NIST Special Publication 500-173, National Institute of Standards and Technology, Gaithersburg, MD, October 1989.

[TEOR86] Teorey, Toby J., Dongging Yang, and James P. Fry, "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model," ACM Computing Surveys, Vol. 18, No. 2, June 1986, pp. 197-222.

[WERT89] Wertz, Charles J., The Data Dictionary Concepts and Uses, Second Edition, QED Information Sciences, Inc., Wellesley, MA, 1989.

BIBLIOGRAPHY

[DOLK87] Dolk, D. R. and Kisch, R. A., "A Relational Information Resource Dictionary System," Communications of the ACM, January 1987, pp.48-61.

[DURE85] Durell, William R., Data Administration: A Practical Guide to Successful Data Management, McGraw-Hill Book Co., New York, NY, 1985.

[FONG89] Fong, Elizabeth N.,and Alan H. Goldfine, editors, Information Management Directions: The Integration Challenge, NIST Special Publication 500-167, National Institute of Standards and Technology, Gaithersburg, MD, September 1989.

[ISO82] ISO TC97/SC5/WG3, Conceptual Schema, Appendix D: The Entity Attribute - Relationship Approaches, ISO Publication, March 1982.

[LEON82] Leong-Hong, Belkis W., and Bernard K. Plagman, Data Dictionary/Directory Systems Administration, Implementation and Usage, John Wiley & Sons, 1982.

[MART77] Martin, James, Computer Database Organization, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1977.

[MART89] Martin, James, Information Engineering, Books I, II, III, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.

[MART90] Martin, James, Strategic Data-Planning Methodologies, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1990.

[ULLM88] Ullman, Jeffrey, Principles of Database and Knowledge Base Systems, Volume I, Computer Science Press, 1988.

[YOUR89] Yourdon, Edward, Modern Structured Analysis, Yourdon Press, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989

*U.S. G.P.O.:1991-311-891:40060

NIST-114A
(REV. 3-90)

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

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NIST/SP-500/197

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
November 1991

4. TITLE AND SUBTITLE

Guide to Schema and Schema Extensibility

5. AUTHOR(S)

Bruce K. Rosen; Isabella des Fontaines

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
Final

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

Same as item #6

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.)

This guide was developed to assist both the casual user of Information Systems (IS) as well as ADP professionals in understanding the concepts behind databases and data dictionary schemas and schema extensibility. It was developed in the context of its application and pertinence to the ANSI standard X3.138-1988, Information Resource Dictionary Systems (IRDS).

This guide begins with a set of definitions that provide an understanding of data dictionary and repository terminology. It then follows with a discussion of the IRDS. After establishing these basic definitions, the discussion of schema is initiated. The guide then takes the reader step by step through the concept of extensible schemas and how they are useful in performing the functions that should be accomplished in the areas of Information Resource Management (IRM) and Data Administration. The document also discusses the importance of schema extensibility to the exchange of information between Computer Aided Software Engineering (CASE) tools. Included is a discussion of the CASE tool to IRDS data exchange prototype that was developed at NIST.

The document concludes with a discussion of possible future impacts of schema extensibility on the development of standards in areas related to data dictionaries and repositories.

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

CASE; Computer Aided Software Engineering; databases; data dictionary; DBMS; directory; encyclopedia; Information Resource Dictionary System; IRDS; life-cycle; repository; schema

13. AVAILABILITY

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

14. NUMBER OF PRINTED PAGES
32

15. PRICE

ELECTRONIC FORM

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SYSTEMS TECHNOLOGY

Superintendent of Documents
Government Printing Office
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Institute of Standards and Technology Special Publication 500-.

Name _____

Company _____

Address _____

City _____ State _____ Zip Code _____

(Notification key N-503)

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences.
Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published bi-monthly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.
Order the **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*
Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Institute of Standards and Technology
Gaithersburg, MD 20899