FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

Open Archival Information Systems for Database Preservation

Carlos Filipe Pereira Aldeias

Report of Dissertation

Master in Informatics and Computing Engineering

Supervisor: Gabriel de Sousa Torcato David (PhD)

Second Supervisor: Maria Cristina de Carvalho Alves Ribeiro (PhD)

5^{th} July, 2011

Open Archival Information Systems for Database Preservation

Carlos Filipe Pereira Aldeias

Report of Dissertation

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: João Carlos Pascoal de Faria (PhD)

External Examiner: Daniel Coelho Gomes (Title)

Internal Examiner: Gabriel de Sousa Torcato David (Title)

05^{th} July, 2011

Abstract

Relational databases are complex digital objects and their preservation is a challenging problem. The *DBPreserve* project has proposed a solution that implements the migration from a relational model to a dimensional model, storing all the relevant data in a data warehouse. A data warehouse can be implemented in multi-dimensional structures, or in relational databases that represent the dimensional model concepts in the relational model, as was used in this project.

As a second step in the preservation process, the *Open Archival Information Systems for Database Preservation* proposed here complements the representation of information in the data warehouse, making it conform to the OAIS reference model and delivering a technologically-independent model for the preservation of complex digital records, such as those constituting the databases and data warehouses. A more technologically-neutral format for database representation is obtained using XML-based files as a long-term preservation format.

Regarding the OAIS reference model, this research reflects on the archival system entities and their characterization, focusing also on the structures of the *Submission Information Package* (SIP), the *Archival Information Package* (AIP) and the *Dissemination Information Package* (DIP), according to this project approach.

This work presents the definition of the XML structure that extends the existing SIARD format used for the description and archival of relational databases, enriching it with a metadata layer for the description of the data warehouse components. The *Data Warehouse Extensible Markup Language* (DWXML) is the XML dialect proposed to describe the data warehouse.

To acquire the relevant metadata for the data warehouse and build the archival format, the *DBPreserve Suite* application was developed. The *DBPreserve Suite* handles the migration process of the metadata and the primary data from the data warehouse to the extended SIARD format, and ensures the access to the primary data. The application integrates the *SIARD Suite* component which builds the SIARD format from a relational database, it allows metadata editing through graphical interfaces, displays visual representations of star or snowflake schemas as well as dimension, hierarchies and levels. The *DBPreserve Suite* also generates the DWXML dimensional model metadata layer and combines it with the SIARD format, enabling primary data browsing from the XML-based format files generated.

Resumo

As bases de dados relacionais são objectos digitais complexos e a sua preservação é um desafio. O projecto *DBPreserve* aborda este problema com uma solução que implementa a migração de um modelo relacional para um modelo dimensional, guardando todos os dados relevantes num armazém de dados. Um armazém de dados pode ser implementado em estruturas multidimensionais, ou em bases de dados relacionais que representam os conceitos do modelo dimensional no modelo relacional, como foi utilizado neste projecto.

Dando continuidade neste processo de preservação, o projecto *Open Archival Information Systems for Database Preservation* visa adaptar o projecto *DBPreserve* em conformidade com modelo de referência OAIS e disponibilizar um modelo tecnologicamente independente que complementa a representação da informação no armazém de dados, atendendo à preservação de documentos digitais complexos, tais como as bases de dados e os armazéns de dados. O formato tecnologicamente menos dependente para a representação de bases de dados é obtido utilizando ficheiros XML, que facilita a preservação a longo prazo.

De acordo o modelo de referência OAIS, esta investigação reflecte sobre as entidades do sistema de arquivo e sua caracterização, focando também as estruturas do *Submission Information Package* (SIP, *Pacote de Informação de Submissão*), do *Archival Information Package* (AIP, *Pacote de Informação de Arquivo*) e do *Dissemination Information Package* (DIP, *Pacote de Informação de Disseminação*), de acordo com a perspectiva deste projecto.

Este trabalho apresenta a definição de uma estrutura XML que estende o formato SIARD utilizado para a descrição e arquivo de bases de dados relacionais, enriquecendo-o com uma camada de metadados para descrição dos elementos de um armazém de dados. *Data Warehouse Extensible Markup Language* (DWXML) é o dialecto XML proposto para descrever o armazém de dados.

Para a aquisição dos metadados relevantes para o armazém de dados e a construção o formato de arquivo, foi produzida uma aplicação que controla todo o processo de migração do armazém de dados para o formato SIARD estendido, quer dos metadados quer dos dados primários, e assegura o acesso aos dados primários. Denominada *DBPreserve Suite*, esta aplicação integra o componente do SIARD Suite que constrói o formato SIARD a partir de uma base de dados relacional, permite a edição de metadados através de interfaces gráficas, apresenta visualmente as representações de esquemas em estrela e em floco de neve assim como dimensões, hierarquias e níveis, produz a camada de metadados do modelo dimensional (DWXML) e adiciona-a ao formato SIARD, e disponibiliza a navegação entre os dados primários a partir dos ficheiros XML gerados.

Acknowledgements

This dissertation was supported by several people who made themselves available to answer questions and guide my research in the context of relational databases preservation, as well as in the areas of dimensional modeling of a data warehouse and XML technologies.

I specially appreciate the support of my honorable supervisors, Prof. Dr. Gabriel David[1] and Prof. Dr. Cristina Ribeiro[2], of the Department of Informatics Engineering, Faculty of Engineering, University of Porto, Porto, Portugal, for their assistance in all the phases of this project, from the clear statement of project requirements, to the guidelines on research, the intensive review of the proposed format to cover the various scenarios that may occur in the dimensional model, in the revision of this document as well as the scientific paper published, and in the validation of the developed application.

I sincerely thank to my colleague Arif Ur-Rahman, a PhD student in the area of database preservation, for the exposure of the DBPreserve project progress, in which he was also involved, particularly in the migration model process, from the relational model to the dimensional model, which supported the my study, implementation and accomplish of the objectives of my project.

Carlos Filipe Pereira Aldeias

[1]E-mail: gtd@fe.up.pt
[2]E-mail: mcr@fe.up.pt

”The goal of any preservation program is to ensure long-term, ready access to the information resources of an institution.”

Abby Smith, ”Preservation in the Future Tense”

Contents

List of Figures

List of Tables

Abbreviations

ADT	Abstract Data Type
AIP	Archival Information Package
API	Application Programming Interface
BLOB	Binary Large Object
CAMiLEON	Creative Archiving at Michigan and Leeds
CCSDS	Consultative Comitee for Space Data Systems
DBML	Database Markup Language
DBMS	Database Management Systems
DDL	Data Definition Language
DGARQ	Direcção Geral de Arquivos
DIP	Dissemination Information Package
DML	Data Manipulation Language
DTD	Document Type Definition
DW	Data Warehouse
EAD	Encoded Archival Description
FCT	Fundação para a Ciência e a Tecnologia
FEDORA	Flexible Extensible Digital Object Repository Architecture
GUI	Graphical User Interface
ISO	International Organization for Standardization
JDBC	Java Database Connectivity
JDOM	Java Document Object Model
JISC	Joint Information Systems Committee
LDS	Logical Data Schema
LOB	Large Object
MD5	Message-Digest Algorithm 5
MOLAP	Multidimensional Online Analytical Processing
NSF	National Science Foundation
OAIS	Open Archival Information System
OJDBC	Oracle Java Database Connectivity
OLAP	Online Analytical Processing
OPF	Open Planets Foundation
PDF	Portable Document Format
PDI	Preservation Description Information
PLANETS	Preservation and Long-term Access through Networked Services
RDB	Relational Database
RODA	Repositório de Objectos Digitais Autênticos
ROLAP	Relational Online Analytical Processing

SGML	Standard Generalized Markup Language
SHA	Secure Hash Algorithm
SIARD	Software Independent Archiving of Relational Databases
SFA	Swiss Federal Archives
SIP	Submission Information Package
TEI	Text Encoding Initiative
UVC	Universal Virtual Computer
USA	United States of America
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	eXtensible Markup Language

Chapter 1

Introduction

> "Digital information comes in a range of types, and while over 80% of organizations already need to preserve documents and images, by 2019 over 70% will need to preserve databases, websites, audio and video files as well." [Sin10, chap. Key Findings]

The technological generation in which we live has gradually modified the method of creation, processing and storage of information, using compulsively digital means for this purpose. The institutions, enterprises and governments rely increasingly on systems that increase the availability and accessibility of information. These information systems typically use relational databases, transforming them into valuable assets for those entities.

However, rapid technological changes generate into rapid obsolescence of applications, file formats, storage media and even Databases Management Systems. If nothing is done, access to stored information will be impossible and it will be lost forever. So, it is important that entities which have major responsibilities in preserving information in digital form, to become aware of this problem and join to initiatives all over the world, seeking for the best methodology for digital long-term preservation, namely on preservation of relational databases.

1.1 Context

The *Open Archival Information Systems for Database Preservation* project is an inherent module of the DBPreserve project, a research project funded by the FCT, that aims to study the feasibility of using data warehouses technologies on preserving complex electronic records, such as those constituting the databases.

The DBPreserve project studies the use of a data warehouse to preserve a relational database after its model migration, in accordance with standards such as the Open Archives

Initiative and the development of rules for transformation of operational databases systems to data warehouses. The results of migration from a relational model to a dimensional model have been tested in a case study in SiFEUP.

The *Open Archival Information Systems for Database Preservation* project pretends to complement the work done so far in the DBPreserve project, with an OAIS model approach and using XML to obtain a technological independent format for long-term preservation.

Open archives are complex systems that requires lots of human resources and time consumption in researching and development. The scope of this project, regarding the Open Archival Information System (OAIS) reference model, aims to define the information packages that are delivered to the archive (Submission Information Packages), as well the archived packages (Archival Information Packages) and the packages that are retrieved from a search on the OAIS to a future consumer.

For the XML migration, it will be defined an XML dialect that characterizes and represents in XML the dimensional model. XML has become the main language of data interoperability when it comes to maintaining human readability and for information exchange between systems. The migration process from the dimensional model to the XML model will require an application that enables the migration between these models. This application is required to be a partial automated process that collects the metadata and the primary data, in order to create the XML representation of the database. The results of migration from a dimensional model to the XML model, will be tested in the same case study (SiFEUP[1]), used by DBPreserve project.

1.2 Motivation and Objectives

The growing interest and concern of the international community in the preservation of digital objects has allocated huge resources to study the feasible and effective preservation strategies. Initially the focus was on the simple digital objects preservation, such as documents and images. Given the increasing reliance on new information and communication technologies and the rising of different digital objects formats, many of them with complex structure, emerged from this research other concerns about these strategies, particularly regarding the long-term preservation of relational databases.

Relational databases are complex digital objects, with a well-defined and distinct data structure, organizing data by attributes, records and tables, possessing integrity constraints and relationships between attributes and tables, as well as routines that complement the processing of these data (e.g., triggers, functions, procedures) according to the business logic of the context in which it operates.

[1] Information System of Faculty of Engineering, University of Porto, Portugal

Businesses and institutions use relational databases and/or data warehouses to support its digital infrastructure. The data is an asset of utmost importance to those entities that have a strong interest in preserving them for queries, statistics analysis, decision support and historical reports.

DBPreserve project approaches the long-term preservation of relational database by means of a two step migration:

- a model migration, from the relational model to a dimensional model, using data warehousing concepts for model and analysis simplification [RDR10];
- an XML migration, from the dimensional model to an XML dialect that represents the data warehouse, to ensure a long-term preservation format.

The main objectives of my project, the *Open Archival Information Systems for Database Preservation* project, are to implement the second migration posted above, the XML migration, and to intervene in the DBPreserve project in order to conform it to the OAIS standard, regarding the information packages.

The specific objectives required for this work are listed below:

- Analyze the DBPreserve project results so far, including the properties of the data warehouse and the transformation rules for operational databases systems to data warehouses;
- Study about the relevant work done so far on database preservation;
- Define the Submission Information Package, the Archival Information Package and the Dissemination Information Package, conforming to the OAIS standard, and identify the metadata needed to ensure long-term preservation of the data itself, of the relational, dimensional and XML model structures. The access functionality should maintain compatibility with the language already used in the EAD files [Gro02];
- Develop an XML based format to represent the data warehouse structure and primary data;
- Implement an application to automate the migration between the data warehouse to the XML;
- Test the XML language and the created tool on the case study used in DBPreserve project;
- Write and submit a scientific paper with the project achievements and tests results.

1.3 Document Outline

Besides the introduction, this document contains 6 more chapters. At Chapter 2, is presented some concepts about databases, models and database management systems. At Chapter 3 is described the state of the art regarding databases preservation and related works are presented. Chapter 4 presents a draft for the metadata requirements of an OAIS, defining the stucture of the SIP, AIP and DIP. The presentation of the preservation format for data warehouses implemented with relational database technologies, is done at Chapter 5. Chapter 6 describes the implementation of the DBPreserve Suite application that manages the migration process and the primary data browsing. Chapter 7 summarizes all the effort in this project, presents the conclusions achieved and points to related future work. This report contains two appendixes: the appendix A replicates the "DWXML - A Preservation Format for Data Warehouses" paper presented at XATA2011 conference; the appendix B presents and describes the graphical user interfaces of the DBPreserve Suite application.

Chapter 2

Databases

Databases have assumed an important role in respect to high amount of information storage, business processes organization and definition or other usual activities on a daily basis. In fact, routine tasks like check your bank balance, book an hotel accommodation or booking a train or plane trip, consult a library digital catalog, make grocery shopping, search for a contact or schedule an appointment in an electronic agenda, involves the use of databases.

Elmasri and Navathe define database as a "collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning" [EN00]. Date's definition of database relies on a "collection of persistent data that is used by the application systems of some given enterprise" [Dat04].

A database is designed and structured to a well-specified purpose, having an identified group of users which populates and uses the database using implemented applications.

Database approach usually introduces the concept of centralized control of data, having several benefits for organizations, such as data sharing, redundancy decrease, inconsistency avoidance, transaction support, integrity maintenance, security enforcement and standards enforcement [Dat04].

2.1 Database Management Systems

A Database Management System (DBMS) is a set of programs that enables the creation and maintain a database, i.e., a software that handles all access to the database. The DBMS facilitates the process of defining, constructing and manipulating databases.

Databases store data organized by attributes, record and tables. Defining a database involves specifying the attributes data types, structures and constraints for the data. Constructing a database is the storing process of the data on a storage medium controlled by

the DBMS. Manipulating a database refers to functions like database querying and data retrieving, data updating and data reports generating.

A DBMS typically has a collection of functions that supports the database management. Figure 2.1 illustrates the major DBMS functions and components[1] [Dat04].

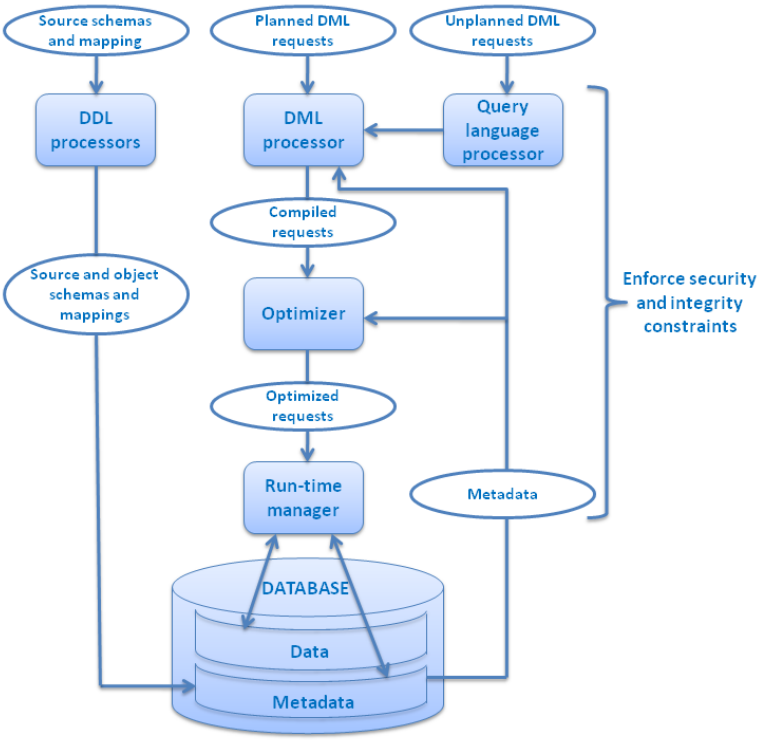

Figure 2.1: Major DBMS functions and components

- **Data definition** - The DBMS must include the Data Definition Language (DDL) or DDL compiler for each data definition languages;
- **Data manipulation** - The DBMS must include the Data Manipulation Language (DML) or DML compiler to deal with the data manipulation language. Generally, DML requests can be planned or unplanned;
- **Optimization and execution** - DML requests (planned or unplanned) must be processed by the optimizer, which determines an efficient implementation for the request;
- **Data security and integrity** - The DBMS or related systems must monitor user requests and reject any attempt against data security and integrity constraints;

[1] The functions and components described below are paraphrased in its original form in pages 45-48 from the book "An Introduction to Database Systems (Eight Edition)" by C. J. Date. Pearson, Addison Wesley, 2004

- **Data recovery and concurrency** - The DBMS or the transaction manager must enforce recovery and concurrency controls;
- **Data dictionary** - The DBMS must provide a data dictionary, i.e., the metadata needed to understand the database schemas and mappings;
- **Performance** - The DBMS must perform efficiently all of its tasks.

A DBMS improves data redundancy control, unauthorized access restriction, persistent storage of objects and data structures, allows inference and actions using rules, provides multiple user interfaces, represents complex relationships among data, enforces integrity constraints and provides backup and recovery tools [EN00].

The data models with major use in many commercial DBMSs are the **relational data model** and the **object data model**. Although, there are other database systems based on the **hierarchical** and **network data models**. From evolution of DBMSs emerged a new approach based on object databases, the **object-relational data model** [EN00].

2.2 Data Warehouses

Transactional systems are planned for predictable workloads, with small units of work, high utilization, with clearly defined performance requirements. The decision systems have typically unpredictable workloads with large units of work and sporadic use, with varying performance requirements. The differences between these systems make it difficult to have a combination of operational system and a decision system in a single system.

Often the decision support systems are fed with data from various operational systems and stored on a separate data store platform, the **Data Warehouse (DW)**.

W. H. Inmon defined a data warehouse as "a subject-oriented, integrated, nonvolatile, time-variant collection of data in support of management's decisions" [Inm92]. Data warehouses fulfills two major purposes: provides a single, clean and consistent source of data for decision support and unlinks the decision platform from the operational system [Dat04].

The characteristics of a data warehouses are presented below to understand the major differences between operational systems[2].

- multidimensional conceptual view;
- generic dimensionality;
- unlimited dimensions and aggregations levels;

[2]The characteristics list below were copied in its original form from the book of Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems (Third Edition). Addison-Wesley, 2000

- unrestricted cross-dimensional operations;
- dynamic sparse matrix handling;
- client-server architecture;
- multi-user support;
- accessibility;
- transparency;
- intuitive data manipulation;
- consistent reporting performance;
- flexible reporting.

The category of software tools that provides analysis of stored data is called Online Analytical Processing (OLAP) [ESC93, CD97], that enables users to analyze different dimensions of multidimensional data. There are mainly two different types of OLAP: the Multidimensional OLAP (MOLAP), where data is stored in a multidimensional cube; and the Relational OLAP (ROLAP), that relies on manipulating the data stored in relational databases. The ROLAP approach is the one in focus regarding the dimensional model of a data warehouse.

Data warehouses are often implemented using relational database technology, and thus they are made up of tables that stores data. A deeper inspection leads to the finding of facts, dimensions, bridges tables, hierarchies, levels, level keys, indexes and views. However, there are some key differences between a database used in an operational system and in a data warehouse.

2.2.1 Dimensional model metadata

The structure of a data warehouse is referred to as a **dimensional schema**, where the **fact tables** are surrounded by **dimensional tables**, forming **star schemas**. A **fact table** is often located at the center of a star schema and consists of facts of a business process (e.g., measurements, metrics). A **dimensional table** (part of a set of dimensional tables) contain attributes in order to define and group data for data warehouse querying.

The dimensions are characterized by a set of levels with defined hierarchies. Hierarchies are logical structures that use levels to organize and aggregate data, define navigation paths or establish a family structure [Inm92, KR02]. A common example is a time dimension: a hierarchy might aggregate data from the day level to the week level to the month level to the quarter level to the year level.

Figure 2.2 shows an example of a star schema related to a real world case study used in the project, a "Course Evaluation System", aiming to obtain general statistics about user satisfaction (anonymous students) in an academic environment scope, specifically on professor and class evaluation.

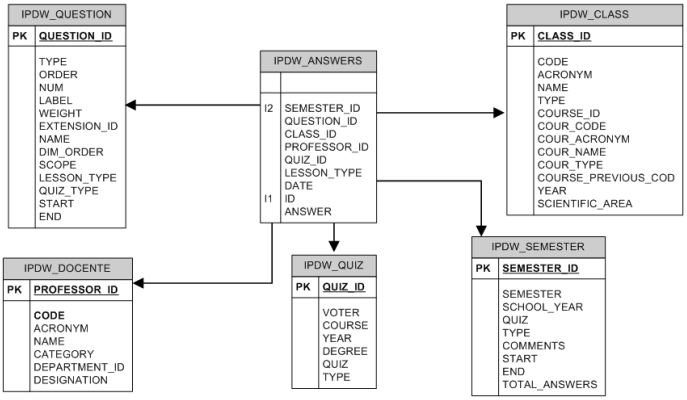

Figure 2.2: Star schema example

In the center, a fact table contains the submitted answers (`IPDW_ANSWERS`). As dimensional tables, there is the question table (`IPDW_QUESTION`), also the quiz table (`IPDW_QUIZ`), the semester table (`IPDW_SEMESTER`), the class table (`IPDW_CLASS`) and finally the professor table (`IPDW_PROFESSOR`). Because the answers are anonymous, there is no relationship towards the students, who actually answered the questions.

An important step in the data warehouse building process is to declare the dimensions. The next code sample shows a declaration of a dimension with the `CREATE DIMENSION` SQL statement [Ora03] using Oracle Database 11g.

Example of a dimension declaration

```
CREATE DIMENSION class_dim
  LEVEL class IS (IPDW_CLASS.CLASS_ID)
  LEVEL course IS (IPDW_CLASS.COURSE_ID)
  HIERARCHY class_rollup(
    class CHILD OF
    course)
  ATTRIBUTE class DETERMINES
    (IPDW_CLASS.CODE, IPDW_CLASS.ACRONYM,
     IPDW_CLASS.NAME, IPDW_CLASS.TYPE)
  ATTRIBUTE course DETERMINES
    (IPDW_CLASS.COUR_CODE, IPDW_CLASS.COUR_ACRONYM,
     IPDW_CLASS.COUR_NAME, IPDW_CLASS.COUR_TYPE,
     IPDW_CLASS.COURSE_PREVIOUS_COD);
```

This declaration defines a dimension (`class_dim`) with a hierarchy of two levels (`class_rollup`): the level `course` with `COURSE_ID` as level key, and a child level `class` with `CLASS_ID` as level key. This dimension is implemented by the `IPDW_CLASS` table. The `ATTRIBUTE` clause specifies the attributes that are uniquely determined by a

hierarchy level. Thus it is possible to analyze the data in a more global perspective, through the `course` level, or get a more detailed overview using the `class` level.

Another data warehouse concept is a bridge table. A bridge table is used to resolve a many to many relationship between a fact table and a dimension table and is also used to flatten out a hierarchy in a dimension table [KR02].

Storing snowflake schemas and data marts is also needed. The snowflake schema is similar to the star schema, but dimensions are normalized into multiple related tables. A data mart is a subset of a data warehouse [KR02, Hac97]. Figure 2.3 represents a snowflake. Comparing this image with the star schema shown before (figure 2.2), it is clear that the `IPDW_CLASS` in the star schema was normalize, resulting in a new table (`IPDW_COURSE`), also called as a sub-dimension.

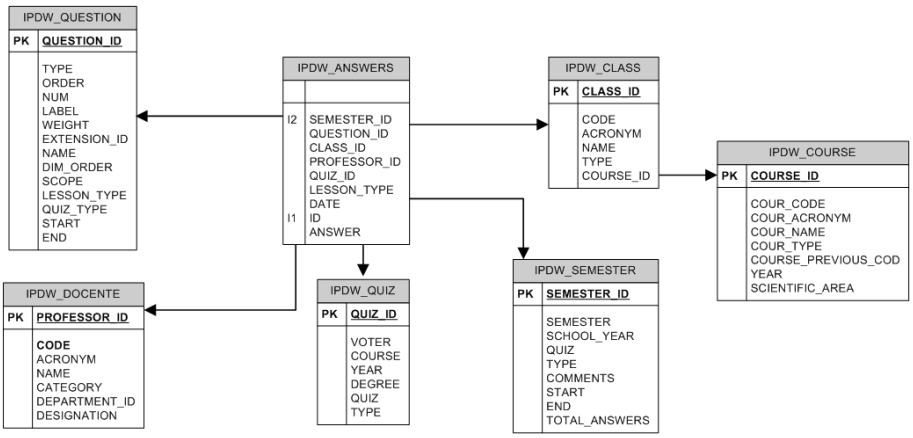

Figure 2.3: Snowflake schema example

2.3 Summary and Conclusions

This brief chapter aimed to introduce the basic concepts about databases. In the first section was presented the advantages of using databases and the improvements introduced by a DBMS.

The architecture of a database management system was intended to illustrate the composition and generic operation of this type of platforms.

The clear differences between operational systems and decision systems have been identified, thus there have been enlightened the concepts related to data warehouses, which uses dimensional data models.

Chapter 3

Database Preservation

The concern about preservation of conventional formats of digital objects (e.g., documents, images) has attracted several investments that study the different strategies, justify the choices and point out scenarios to better fit of each preservation strategy. However, the databases are different from conventional digital objects as they have an internal structure, include schemas and integrity constrains, which are vital for data interpretation.

PresDB'07 workshop report states that "existing preservation techniques for fixed digital objects are not suited for databases, thus some of our most critical digital assets are endangered - both economically and technically - in the long term" [CB07].

This section introduces the concepts, requirements and strategies for digital preservation in the long term. "Long term is long enough to be concerned with the impacts of changing technologies, including support for new media and data formats, or with a changing user community. Long term may extend indefinitely" [fSDS02]. The concepts of physical object, logical object and conceptual object are introduced, as well as the terminology and architecture of an *Open Archival Information System*. It will be described the digital preservation strategies and pointed out its adaptability and suitability for databases preservation. Several projects have been developed to ensure the preservation of digital objects, being posted the projects that stand out in the context of relational databases preservation.

3.1 Digital objects

In a generation of growing dependence on information technology and communication, the production of information is performed using mainly digital formats, such as articles, documents, books, images, videos, databases and software applications. These objects are known as digital objects, represented by a sequence of bits. According to Thibodeau,

"a digital object is an information object, of any type of information or any format, that is expressed in digital form" [Thi02].

Thus, whether the objects are created from the ground in digital formats, or those that are digitized from analog contexts, both are covered by this definition.

These digital objects are stored on physical media (e.g., hard disks, CD, DVD, tapes), defining the physical objects that require specific devices to read them. The information stored in these media is characterized by a set of symbols, which are organized and controlled by different rules, depending on the physical medium used [MMvPW06, TJC06, Ass07, Bhu00, CTI10, CGOZ99]. These data structures are the logical level of abstraction of the digital object, also known as a digital object format [Thi02].

In order to a digital object to be presented to the user, it needs the right technological environment, i.e., it must verify the requirements of hardware and software for its processing and display. In the presence of these conditions, the object assumes a level of conceptual abstraction, showing itself with the appearance/shape expected. From the human point of view, the conceptual object is what should be the target of preservation.

Figure 3.1 illustrates the levels of abstraction of a digital object, particularly when it assumes as a database.

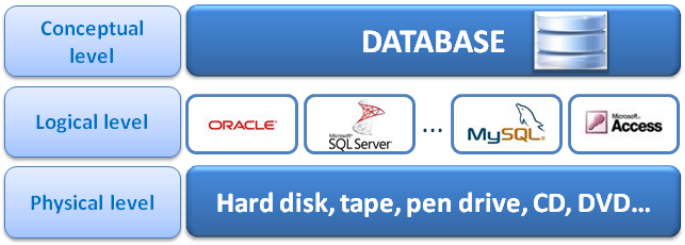

Figure 3.1: Levels of abstraction of a digital object, when it assumes as a database

However, a conceptual object may be interpreted differently by each person. This individual interpretation that each person acquires is called the experienced object. In theory, it is possible to preserve the experienced object [Fer06].

This translation process between levels of abstraction of a digital object is also reflected in the opposite direction. When a person modifies a conceptual object (e.g., insert a record in a table in a database), this modification should be mirrored or stored in a suitable physical support, through an intermediate conversion into binary codes that encodes the input made by the user.

Digital preservation is a process or a set of processes that must follow a concrete plan of activities, with allocation of adequate resources and use of technologies and practices that ensure access to a digital object, in the long-term perspective.

To be able to preserve a digital object, the levels of abstraction described above must be accessible and interpretable, otherwise, it is no longer possible to use that object. For

example, if an institution has a database in a particular DBMS and that DBMS by discontinued, or later versions do not guarantee backward compatibility, the access to this database will be prevented, losing certainly the greatest good that an institution produces, the information.

This overview of the concepts inherent in digital objects allows a better understanding about the preservation strategies presented later in this report.

3.2 Requirements for Long-term Preservation

In the context of databases, there are some key requirements to achieve success in long-term preservation perspective:

- **Integrity** - Data integrity must ensure that data stored in the database is correct and consistent, remaining intact, with no changes or corruptions so that its meaning is no longer clear. For example, if there is a reference to a specific object or entity, that object must exist in the database and its data must be accurate [Des90].
- **Authenticity** - The authenticity of the data must be viewed as a key concept from the perspective of long-term preservation, which means that the preserved data must not be tampered with or corrupted. The digital object must be the very thing it claims to be and have been created by the specified organization or person [For02].
- **Intelligibility** - The intelligibility of a database is defined by the ability to perceive and interpret the data formats and the relationships between the tables and what they represent in reality [RDR10].
- **Originality** - For better definition of database originality, Deveci[1] looks at the issue from different perspectives [Dev04]: considering the originality as a requirement of *copyright*, supporting the definition on the Database Directive [Mar96], comparing originality and investment as requirements to copyright and by interpretation *sui generis* of right risks outreaching *copyright*.
- **Accessibility** - Accessibility to a database is characterized by the possibility of using the data in open formats that do not require specific vendor-software, thereby ensuring access to data from a perspective of long-term preservation [RDR10].

3.3 Open Archival Information System Model Reference

In 1990, the *Consultative Committee for Space Data Systems* (CCSDS) found itself the need to create a set of directives aimed at the long-term storage of digital information

[1] MA, LLM, Dip. in Intellectual Property, GCert.Ed., Barrister Formerly lecturer at South Bank University

produced in space missions. These standards were discussed and developed in a public open forum and culminated with the definition of an *Open Archival Information System* (OAIS) reference model. This model was approved as an ISO standard in 2003 (ISO 14721:2003) [fSDS02].

One of the best contributions of this standard was the definition of specific terminology for the communication between the concerned parties in the long-term preservation of digital objects. Particularly in this study, this model will be analyzed and discussed in the preservation of relational databases and data warehouses built on relational database technologies.

3.3.1 OAIS Concepts

An **Open Archival Information System** is "an organization of people and systems that has accepted the responsibility to preserve information and make it available for a designated community" [fSDS02]. The term 'Open' is just to emphasize the fact that it has been developed in an open public forum, in which any interested party was encouraged to participate.

An **OAIS archive** is one that intends to preserve information for access and use for a designated community.

The term **designated community** is related to an identified group of consumers who should be able to interpret a particular set of information. This group may be composed by multiple user communities.

Figure 3.2 identifies the roles outside of an OAIS: Producer, Consumer and Management [fSDS02].

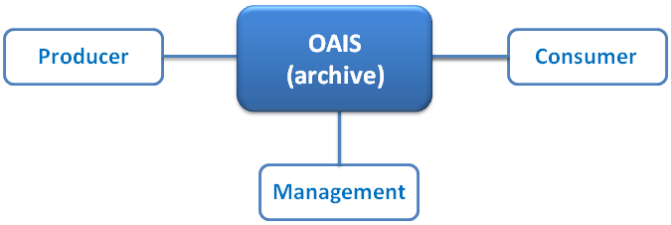

Figure 3.2: Environment Model of an OAIS

The **Producer** is the role assigned to people or system clients which provide information to be preserved. Within this study, they will be those who have relational databases to be preserved in the long-term perspective.

The **Consumer** is the role played by individuals or system customers that interact with OAIS services to find and request preserved information of interest. A subgroup that assumes this role is the designated community. In this project scope, the Consumer will be those who need to consult preserved data and associated metadata. From the standpoint of

portability, they may be entities that will need in the future to migrate the data to another DBMS or another data format.

The **Management** is performed by those who define the overall OAIS policy.

In the OAIS reference model, **information** is defined as "any time of knowledge that can be exchanged, and this information is always expressed by some type of data" [fSDS02].

In databases, the data is scattered and represented by tables, records and attributes. When data from databases is processed, it retrieves information [Ack89].

Representation Information corresponds to the syntax and grammar, i.e., a set of symbols and rules that allows the processing and interpretation of data, obtaining an information object.

One of the definitions with greater emphasis is the concept of **Information Package**. Information Packages are the objects submitted to the OAIS, in one or more submissions, and after processing they are subsequently archived. A future search to the OAIS also returns one or more information packages. An information package is defined as a conceptual container consisting of two types of information, called **Content Information** and **Preservation Description Information (PDI)**, which are encapsulated and identified by **Packaging Information**. The resulting package is labeled and recognized by the **Descriptive Information**.

Figure 3.3 presents a visual definition of the Information Package.

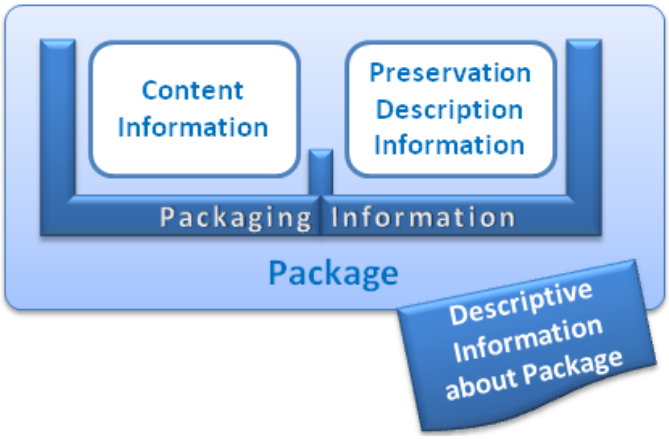

Figure 3.3: Information Package Concept

The **Content Information** is the information that is the initial target of preservation. It consists on the Data Object (physical or digital object) associated with its Representation Information, to allow the interpretation of the Data Object by the Designated Community.

The **Preservation Description Information** is a set of information needed for the

preservation of the Content Information, ensuring its correct identification and contextualization into the environment in which the Content Information was created. The Preservation Description Information is divided into four types of preserving information:

- **Provenance** - describes the source of the Content Information, who it belonged since its creation and its history;
- **Context** - describes how the Content Information relates with other information outside the Information Package (e.g., the reason for the creation of the Content Information);
- **Reference** - provides one or more identifiers for uniquely identifying the Content Information;
- **Fixity** - provides a way to protect the Content Information against undocumented changes (e.g., it may contain a checksum of the Content Information for integrity checks).

The **Packaging Information** is the information that allows the Content Information binding and associating with the PDI.

The **Descriptive Information** is the information used to identify the package containing the desired Content Information. It may consist of a set of labels that describes the Information Package or a set of attributes that facilitates the search.

The definition of Information Package calls for a distinction between the Information Packages that are submitted, those which are archived and those that are disseminated by the OAIS. Thus, the OAIS reference model identifies three variants of Information Packages: the Submission Information Package (SIP), the Archival Information Package (AIP) and the Dissemination Information Package (DIP).

The **Submission Information Package (SIP)** is the package that is submitted to the OAIS by a Producer. The form and detail of the SIP is negotiated between the Producer and the OAIS.

After the submission, one or more SIPs will be processed and transformed into one or more **Archival Information Packages (AIP)** to be preserved, consisting on the Content Information and the associated PDI.

When a Consumer makes a request to an OAIS, it responds with a **Dissemination Information Package (DIP)**, which is derived from one or more AIPs [fSDS02].

3.3.2 OAIS Responsibilities

An organization wishing to implement an OAIS archive, must comply with a mandatory set of responsibilities[2]:

[2]The responsibilities listed below are reproduced from their original form in page 3-1 of the OAIS Reference Model documentation

- Negotiates for and accepts information from Producers;
- Obtains sufficient control to ensure long-term preservation;
- Determines the Designated Consumer Community;
- Ensures that information is Independently Understandable to the Designated Community;
- Follows established preservation policies and procedures;
- Makes the information available to the Designated Community.

3.3.3 OAIS Functional Model

This section aims to present the OAIS functional entities and how the information is handled by the OAIS. Figure 3.4 specifies in detail the OAIS functional model, identifying six entities and the major information flows[3].

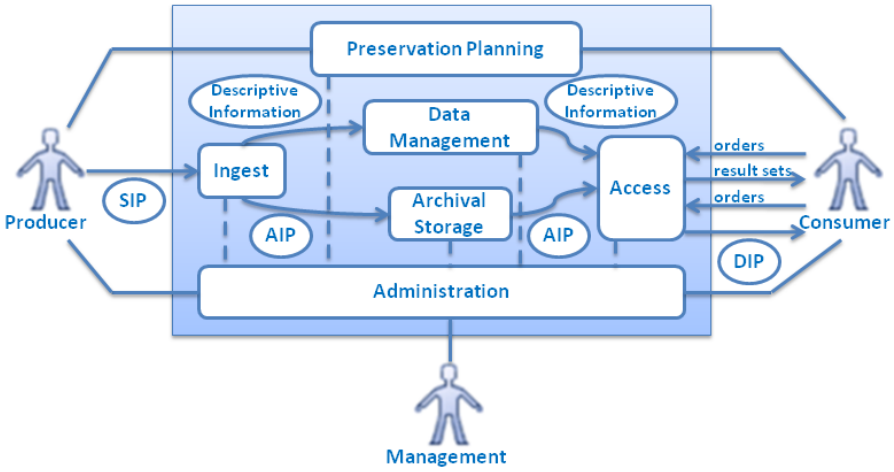

Figure 3.4: OAIS Functional Entities

The **Ingest** is the OAIS entity that contains the services and functions for acceptance of SIP submitted by Producers, prepares the AIP for storage and ensures that the AIP and its Descriptive Information is established within the OAIS.

The **Archival Storage** is the OAIS entity that contains the services and functions for storage and retrieval of AIP.

The **Data Management** is the OAIS entity that contains the services and functions for populating, maintaining and accessing a wide variety of Information (e.g., catalogs, consumer billing, security controls).

[3]The definitions of the entities listed below are reproduced from their original form in Section 1.7.2 of the OAIS Reference Model documentation

The **Administration** is the OAIS entity that contains the services and functions for controlling the other OAIS functional entities.

The **Preservation Planning** is the OAIS entity that contains the services and functions for monitoring the OAIS environment and providing recommendations to ensure that the OAIS stored information remains available to the Designated Community, in a long-term perspective, even if the original computer environment becomes obsolete.

The **Access** is the OAIS entity that contains the services and functions to enable the Consumers to view and use the related services with the preserved information.

3.4 Preservation Strategies

On Digital preservation, the research projects have a concern on defining which is the best strategy that is sustainable and efficient for the long-term preservation of digital objects. There are already many efforts and projects developed under this scope. Projects such as CAMiLEON [HL01], InterPARES [For02] or FEDORA [LPSW06] contributed to the study of requirements, strategies and proposals for preserving digital objects and its authenticity. Regarding complex digital objects, such as databases, projects like SIARD [IP08], Chronos [BKM07] or RODA [RFFC07], analyzed in detail the preservation of relational databases.

Among the studies conducted over the past two decades, several strategies are identified for preservation and they can be grouped into two distinct domains: strategies with emphasis on technology preservation and strategies for information preservation.

Thibodeau's organization of the different digital preservation strategies relates them to their applicability and objective [Thi02]. Figure 3.5 shows a simplified version of this bidimensional mapping, according to Ferreira' perspective [Fer06], that is sufficiently clear and synthesized for this research purpose. As Thibodeau's organization, this viewpoint arranges in its left extreme the strategies focusing on preservation of the physical/logical object, and at the right side, the strategies focused on preserving the conceptual object.

To ensure the persistence of a digital object, it is necessary that these objects are stored in a physical media (e.g., hard disk, pen drive, CD-R). However, the risk of deterioration, which damages the integrity of this supports, causes crashes, possibly critical errors, on interpreting the information stored therein. Moreover, the fact that a media storage becomes obsolete, without hardware equipment that allows the reading of those supports, could lead to the lost of the information contained [LSL+02].

Not so much as a strategy but more than a pre-condition for digital preservation, entities must copy the digital information to a new storage medium before the former one becomes so obsolete that does not allow the access to its information. This process is

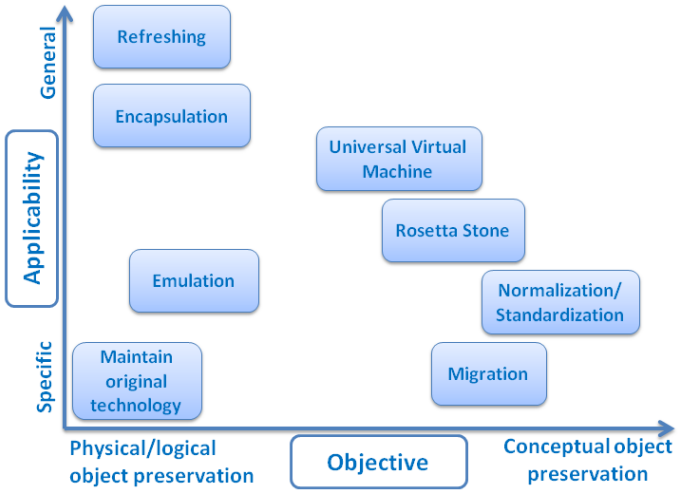

Figure 3.5: Digital Preservation Methods

called **refreshing** or **copying** [Hen98]. Garrett and Waters describe the techniques for preserving the integrity of digital information within this context [GW96].

3.4.1 Technology Preservation

One of the methods studied initially regarding digital preservation pointed to the preservation of the technological environment in which the conceptual object was designed. This would need to preserve and maintain the hardware and software required to access that object, i.e., it would be like a museum of technology, both hardware and software. Of course, the object to be archived must also be preserved and stored on a stable digital medium. The advantages of this method are clearly concerned in preserving the original appearance of the conceptual object, its *look-and-feel*. Moreover, this process would be extremely costly given the variations of systems and environments to preserve, with greater difficulties in maintaining the hardware operational (e.g., hardware faults, obsolete spare parts) and constraints on access and reuse of information [Hen98]. "Over the long term, keeping original technology is not practicable and may not be feasible" [Thi02].

Having the conceptual object as a database, it would be necessary to preserve the DBMS, its correct or compatible version with the database, the operating system where this DBMS was compatible, the front-end application that allows to use the database and associated features, a compatible browser if it was an *Web* interface, the hardware required to run the operating system, applications and browsers, among other requirements like programs, updates, patches, drivers. In fact, although it would be important to maintain this requirements for a strategy of this kind, it would be almost unsupportable from the point of view of creating a repository of relational databases, due to the panoply of available DBMS, operating systems, browsers and all of their versions.

3.4.1.1 Emulation

The Thibodeau's definition of emulation, where he states that "Emulation strives to maintain the ability to execute the software needed to process data stored in its 'original' encodings" [Thi02], introduces a description of the preservation process using this technique. This strategy simplifies the problem of data preservation, since it allows the use of formats in their original encoding, while maintains the look, feel and behavior. To make this possible, it is necessary to have an emulator. An emulator is a specific software that allows technological recent computers to replicate the behavior of obsolete computers [Rot00a]. This technique decouples application programs from the platform via a virtual machine. So, applications can run on different platforms by migrating the virtual machine to those platforms (e.g., Java virtual machine). Rothenberg states that emulation can be performed at three levels: at application software level, at system software (operating system) level and at hardware level [Rot00b].

Hoeven et al. state that emulation in the digital preservation context, can be generally identified as *Stacked Emulation*, *Migrated Emulation* and *Emulation Virtual Machine* [vdHvW05].

According to Hendley, emulation of technology is regarded as the leading option when there are digital resources that are not able to be converted to open formats. He sees the emulation as a strategy in a short to medium term or as a specific strategy when the focus of preservation sits on the look and feel of the original digital resource [Hen98].

Granger says that "emulation is not a complete digital preservation solution but the partial one" [Gra00]. Ensuring that there are clear advantages in re-creating the look and feel of a resource, also points out to possible disadvantages arising from the uncertainty of the direction of technology evolution and complexity in specifying emulators.

Waugh et al. worried about the effects that a computer virus can produce when it pollutes the software application, which can lead to loss of information over time. They believe that emulation is a viable solution for such cases where the objective is to preserve the software as an artifact itself and future user organizations lack sufficient knowledge to understand the format of the digital information [WWHD00].

Russell points out to emulation as the best solution for very long-term preservation of digital objects, mainly for resources with unknown value and where future use of the material is unlikely [Rus00].

In summary, emulation strategy presents a strong point: it preserves the look and feel, recreating the technological environment of the original digital object. As negative aspects are pointed the complexity in the construction of emulators, emulators themselves are subject to obsolescence, do not guarantee the reuse of information and users must know the systems and applications already abandoned.

The strategy of emulation can be applied in the context of long-term preservation of relational databases. However, considering the advantages and disadvantages outlined above and designing a method to fit in an OAIS, this strategy would be extremely complex, because it will need to emulate multiple operating systems, so they could execute the multiple DBMS and applications to access and to use archived databases. Furthermore, the preservation of databases intended primarily to maintain long-term access to data stored there, without much need to keep the look and feel of the original application/interface.

3.4.1.2 Encapsulation

The strategy of encapsulation addresses the problem of technology obsolescence of digital formats through the preservation of the information along with details of how to interpret this information. This technique aims at re-creating the original application technological environment in the future, which provides access to create or preserve the digital object.

The encapsulation of the digital object with the information necessary for its preservation, including metadata, can be done through structures called containers or wrappers [Day98].

Aiming to create a standard format, regardless of application, operating system and the hardware, packaging metadata for digital preservation to the object, came the Universal Preservation Format (UPF) [SM98].

This method is oriented towards objects that are accessed in the distant future, deferring the responsibility of preservation. However in future, it must be built applications that allow visualization, emulation or migration of these objects.

Taking into account this method regarding preservation of databases, together with the encapsulation of information required for specification of a DBMS, this approach does not appear at all suitable for this purpose. An incorrect specification of complex objects could culminate in disastrous effects, particularly in the integrity and access to digital objects preserved.

3.4.1.3 Universal Virtual Computer

UVC stands for *Universal Virtual Computer*. "The UVC is a Computer in its functionality; it is Virtual because it will never have to be built physically; it is Universal because its definition is so basic that it will endure for ever" [Lor01].

The concept of Universal Virtual Computer was developed in 2000, by Raymond Lorie. This approach uses coupled migration and emulation that allows digital objects to be reconstructed in its original form. To implement this strategy, it is necessary, besides the UVC itself, the logical data scheme with type description, the UVC program (format decoder) and the logical data viewer [Lor02].

To better understand the workings of this process, let's look at an example. Assuming you want to archive a file, it will be archived in its original form and the program (format decoder) that allows its interpretation is also preserved, but this program is written for the UVC compatibility. In the future, the only thing required would be to emulate that UVC, which then would be able to run the program (format decoder) that allows the access to and interpretation of the archived file. Thus, information can be returned to a future client, according to its logical view, which is quite similar to XML. This UVC "is a general purpose computer, complete yet basic enough as to remain relevant for a very long time" [Lor05]. The simplicity of the UVC allows a simple writing of a program that emulates it.

Figure 3.6 presents the Universal Virtual Computer and its components.

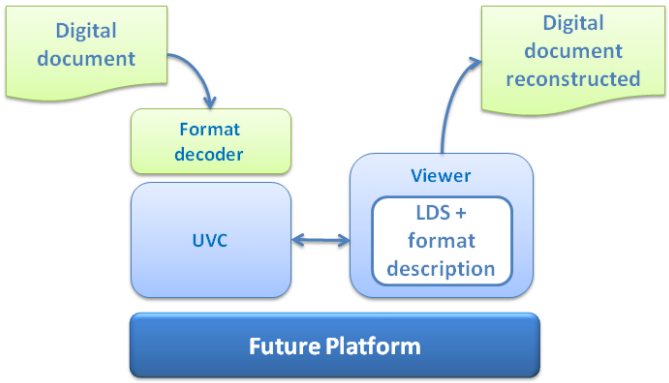

Figure 3.6: UVC and its components

There are a proof of concept for this approach on the archiving of PDF documents [Lor02]. Lorie and Diessen also promoted this technique in the context of the long-term preservation of complex processes [LvD05].

3.4.2 Information Preservation

Regarding on information preservation, strategies that emphasize on the conceptual object preservation are mainly migration techniques. Rather than focus on the technology, information migration focuses on the conceptual object and on ensuring its accessibility using available technology [Rus00].

The strategies that are based on preserving the conceptual object are Migration, Backward compatibility, Interoperability, Conversion and XML.

3.4.2.1 Migration

Migration is a periodic transfer of an object from a technological environment to another. There are many variants of digital migration. It can be done from digital to analog

formats [GW96], from a digital format into a more recent technological format or even migrate to a concurrent format [Thi02], migration-on-request [MWS02] and distributed migration [Fer06, WT04]. Migration has been one of the most successful techniques for preservation of digital objects [Fer06, GW96, Tes03].

The main purpose of migration is to maintain the integrity of the preserved object, providing access and viewing by users. Furthermore, migration can be practiced only with the aim of using the latest formats so that they can be used in applications technologically more advanced [GW96].

This strategy is widely used to transfer information between systems or applications, including for the replacement of information systems. So, the information created on one platform can be easily used by a substitute platform, keeping untied application and information [WWHD00].

However, the migration process must be preceded by a study, examining the compatibility of the new format with the original format, to preserve the authenticity and integrity of the digital object, and ensure that the main properties of the original object are reflected in the new object [Tes03].

The OAIS reference model [fSDS02] fragments migrating into four categories, Refreshment, Repackaging, Replication and Transformation[4]:

- **Refreshment** - A Digital Migration where the effect is to replace a media instance with a copy that is sufficiently exact that all Archival Storage hardware and software continues to run as before.
- **Repackaging** - A Digital Migration in which there is an alteration in the Packaging Information of the AIP.
- **Replication** - A Digital Migration where there is no change to the Packaging Information, the Content Information and the PDI. The bits used to represent these Information Objects are preserved in the transfer to the same or new media instance.
- **Transformation** - A Digital Migration in which there is an alteration to the Content Information or PDI of an Archival Information Package. For example, changing ASCII codes to UNICODE in a text document being preserved is a Transformation.

3.4.2.2 Backward Compatibility

Backward Compatibility characterizes the new versions of software applications that are able to read digital objects that were created in an earlier version of that application, without sacrificing features or introducing substantial changes in the visualization [Tes03].

[4]The categories listed below were copied in its original form from the Section 1.7.2 of the OAIS reference model documentation, to avoid the use of different terminologies

Nowadays, most of the current software products guarantee backward compatibility for files created with earlier versions, as well as portability of digital objects between systems. Despite being a simple and low-cost strategy, shall be made periodical migrations of digital objects to recent versions in order to prevent that the format of the digital object becomes too obsolete [Hed97].

This approach is viewed as a short-term strategy and viable for simple digital objects. However, it is not reliable for complex objects such as databases, particularly in preserving at the medium and long-term [LSL+02].

3.4.2.3 Interoperability

Interoperability is related to using a digital object across multiple platforms, while maintaining its functionality and its usual visualization. The term interoperability may be applied to digital objects created and used by applications that runs on different operating systems, or may be related to the compatibility of two similar applications that are able to interpret correctly the same digital object. Another form of interoperability resorts to an interim program to synchronize different applications.

The databases which have interest in being preserved, usually have a very high complexity, often running on a central server and containing own tools for import and export data between databases, which requires some care in its use, with the risk of losing data. The interoperability, seen as a strategy for long-term preservation of databases, is not recommended [Tes03].

3.4.2.4 Standardization

Standardization is a migration from proprietary formats, usually closed, to open formats, reducing the dependency on the technological environment and the risk of obsolescence of the original system. This strategy can use two types of standards: *de jure* standard (created in open processes involving an standardization officially accredited organization, like ISO or W3C; e.g., XML) or *de facto* standards (created in closed processes and widely used, e.g., PDF) [Tes03].

The conversion to standards may be used as a database preservation strategy, as it will achieve both backward compatibility and interoperability. Digital Preservation Testbed has examined the standards ASCII, SQL, and XML [Tes03].

3.4.2.5 XML

XML stands for eXtensible Markup Language and it is an open standard defined by the World Wide Web Consortium (W3C). This standard is a very flexible text format derived from SGML (ISO8879) [SMB94], and it is widely used to structure, exchange and store data [Con08].

This strategy is basically a type of migration - standardization. XML is a simple text format, independent of platforms and applications, and it can be interpreted with a simple word processor [Tes03]. It is also human readable, but it can be sometimes verbose. This method stands out as a good option for digital preservation. XML is seen as a good storage format, with good portability and technology neutrality.

XML also introduces the concept of separation of content, structure on one side and appearance on the other. Another positives aspects in using XML focused by Verdegem[5] is the fact that XML is royalties free [Ver03]. He also points out the need of complex processes to access and analyze complex information representations.

3.5 Related Work

This section summarizes the projects that outstand in the scope of digital preservation of relational databases. Much of these efforts are also useful regarding the preservation of simple digital objects.

3.5.1 CAMiLEON

CAMiLEON stands out for *'Creative Archiving at Michigan and Leeds Emulating the Old on the New'*. The CAMiLEON Project[6] is an international research project, funded by the Joint Information Systems Committee (JISC) in the United Kingdom and the National Science Foundation (NSF) in the United States of America (USA). This research addresses to study the feasibility of technology emulation [HL01] for digital preservation strategy and developed some techniques for software longevity [HW01].

CAMiLEON also developed an alternative approach for migration, called Migration on Request. This technique maintains the original byte stream along with a tool to migrate the object at the point of use. They claim that this strategy can be more accurate and cost effective [MWS02, Whe01].

As reported before, emulation (aims at providing programs that mimic a certain environment) and migration strategies are suitable for database preservation, besides their drawbacks. CAMiLEON UK Project Manager, Paul Wheatley, stated that "Migration will be crucial for the preservation of more simple data objects and emulation will undoubtedly be essential for preserving complex objects that incorporate software elements" [Whe01].

3.5.2 Digital Preservation Testbed

The **Digital Preservation Testbed** is an initiative of the Dutch National Archives and the Dutch Ministry of the Interior and Kingdom Relations. It aims at researching and

[5] Project manager, Digital Preservation Testbed, The Netherlands
[6] Project's official site: http://www2.si.umich.edu/CAMILEON/index.html

testing the applicability of different strategies of preserving government and other digital information and keeping it accessible for the future. The Digital Preservation Testbed is part of the ICTU foundation, a non-profit organization intended to contribute to the structural development of the digital government.

This research project studies the effectiveness, limits, costs and application potential of three different approaches to long-term digital preservation: migration, emulation and XML [Sla02].

However, regarding database preservation, Digital Preservation Testbed recommends the use of XML as a preservation strategy. Testbed developed a rapid and simple conversion tool for compatibility with Microsoft Access and Oracle databases, using an XML file structure suited to digital preservation [Tes03].

3.5.3 PLANETS

The PLANETS project (Preservation and Long-term Access through NETworked Services[7]) is funded by the Information Society Technologies (IST) R&D Programme of the European Union. This project is an international major effort that gathered some important partners in Europe, like national archives, national libraries, research institutes and technology companies with digital preservation experience.

Focusing on long-term preservation strategies research, in order to help organizations find the best alternative to their context, ensuring the value of their digital content and managing costs with the preservation processes, the PLANETS project aims to provide a framework enabling the preservation of digital content over the long term. This main objectives of this project are listed below[8]:

- Preservation Planning services that empower organizations to define, evaluate, and execute preservation;
- Methodologies, tools and services for the characterization of digital objects;
- Innovative solutions for Preservation Actions tools which will transform and emulate obsolete digital assets;
- An Interoperability Framework to seamlessly integrate tools and services in a distributed service network;
- A Testbed to provide a consistent and coherent evidence-base for the objective evaluation of different protocols, tools, services and complete preservation plans;

[7] Project's official site: http://www.planets-project.eu/

[8] The PLANETS project' objectives listed were copied in their original form from the official project's website at http://www.planets-project.eu/about/

- A comprehensive Dissemination and take-up program to ensure vendor adoption and effective user training.

Among other studies, the PLANETS project researched on emulation approach [vdH07], on migration strategies [ZvW08], on integrated services for digital preservation [FHY07] and metadata requirements for successful long-term preservation [DF09]. It has used the OAIS model as a basis for its digital preservation framework, that is extensible to meet organizations' requirements [Sin10].

Hoeven et al. researched on emulation strategy and built an open source software modular emulator for digital preservation named Dioscuri. With limited resources, they shown that it is feasible and it execute old application better than a modern computer platform. Being multi-platforms compatible, it is more durable than other emulators [vdH07].

Zierau et al. focused on migration approaches and, rather than build new applications for migration, they reuse and enhance existing migration tools. Although, they also emphasized on differing opinions on the scope of the tool development and quality control [ZvW08].

Dappert et al. summarized the effort by Information Society Technologies (IST) Programme of the European Sixth Framework Programme on researching about metadata for digital preservation services, defining a data dictionary for key digital preservation metadata concepts. The conceptual model also supports dynamic preservation processes, has implications for implementations of preservation metadata dictionaries, property registries and preservation services [DF09].

Farquhar et al. focused on the requirements specification and requirements analysis for preservation services. They reported on the status of some key components for PLANETS architecture, like "registries for storing and accessing information about file formats, implementation of file migration tools as web services so that they can be included in distributed preservation workflows, and prototypes of the Plato preservation planning tool, the Testbed and the PLANETS Interoperability Framework" [FHY07].

PLANETS Project started in June 2006 and finished in May 2010. Nevertheless, the work and research done so far will be maintained and developed by a not-for-profit company, the Open PLANETS Foundation (OPF)[9]

PLANETS framework provide tools and services for digital preservation and a set of integrated conversion or migration tools. Regarding database preservation, it deals with Access, MS SQL Server, Oracle databases, as well the SIARD format [PLA09].

[9]Open PLANETS Foundation official website: www.openplanetsfoundation.org.

3.5.4 SIARD

The Swiss Federal Archives (SFA) has developed an open storage format for relational databases called **SIARD**[10] (Software Independent Archiving of Relational Databases), as well a set of conversion tools named the SIARD Suite, in order to convert relational databases (e.g., Access, Oracle and SQL Server) into the archival SIARD format.

The **SIARD format** is a nonproprietary and published open standard, based on open standard (e.g., ISO norms Unicode, XML, SQL1999) and the industry standard ZIP64. In May 2008, the European PLANETS project accepted SIARD format as the official format for archiving relational databases [IP08]. This format will be carefully analyzed at Section 5.2.2.

3.5.5 FEDORA

In 1998, Payette and Lagoze presented a digital object and repository architecture in order to ensure a reliable and secure architecture for archiving and accessing digital libraries, as well providing the extensibility and interoperability. Called **Flexible Extensible Digital Object Repository Architecture (FEDORA)**, this architecture supports multiple data types, adaptation to new types to appear, aggregation of mixed, possibly distributed, data into complex objects, multiple content disseminations specification and rights management scheme association with these disseminations [PL98].

In 2001, the University of Virginia in partnership with the Digital Library Research Group of Cornell University developed the first digital object repository management system based on the Flexible Extensible Digital Object Repository Architecture (FEDORA). The FEDORA Project was funded from the Andrew W. Mellon Foundation [SWP03].

Although not OAIS compliant, FEDORA project[11] is an open-source digital content repository service, which provides a flexible foundation for managing and delivering complex digital objects and the basis for ensuring long-term preservation of the information, while making it directly available to consumers [LPSW06].

FEDORA repository system is implemented as a set of web services that provide full management of digital objects and search and access to multiple representations of objects. FEDORA APIs use Web Service Description Language (WSDL). WSDL an XML language that describes Web services based on abstract models [CCMW01].

There are some FEDORA-based applications that extends the concepts of initial architecture. One of them is RODA, an OAIS-compliant and service-oriented digital repository system designed to preserve government authentic digital objects.

[10] Project's official site: http://www.bar.admin.ch
[11] Project's official site: http://fedora-commons.org/

3.5.6 RODA

The portuguese **Repository of Authentic Digital Objects project**[12] (in portuguese RODA means Repositório de Objectos Digitais Autênticos) was launched by the National Archives of Portugal (Direcção Geral de Arquivos - DGARQ) in partnership with the University of Minho, Portugal. This project aims to study the issues related to continuous management of digital objects in order to develop processes, tools and resources capable of meeting the needs of preservation of digital information produced by the Public Administration [BCF+07].

The most important objective is to create a prototype system repository that complies to the OAIS reference model [fSDS02] and to ensure the preservation and authenticity of the archived information. In order to prove the viability of the prototype, the initial phase of the project limited the type of files to be accepted by the archive, being only possible the ingestion of text documents, still images and relational databases [RFFC07].

Regarding the archiving of relational databases, RODA sets that only the structure of the database (tables and relationships between tables) and the data itself would be considered objects of preservation [BCF+07].

To ensure the long-term preservation, RODA uses Database Markup Language (DBML), an XML language that migrates the database into a single XML file, defining the database structure and data stored on it. This format will be carefully analyzed at Section 5.2.1. However, typically a database sizes have high enough that the transformation process of XML file in an added problem. "RODA's architecture is Service Oriented and Web Services perform poorly with large volumes of data" [RFFC07].

RODA is based on FEDORA repository, which already includes library management systems, multimedia production systems; archival repositories, institutional repositories, educational digital libraries [RFC+07].

3.5.7 Chronos

In Germany, the cooperation between the Department of Computer Science of the University of Applied Sciences in Landshut and the CSP company originated the '**Chronos Archiving**' project which intended to respond to the need to preserve the contents of relational databases in the long term. This project was funded by the government of Bavaria (Germany) via a program for innovation in computer science and communication, between the years 2004 and 2006. The project aimed to use open formats, independent of the original system for easy archived data retrieval or agile data portability to another version or another database. The search engine uses indexes to improve the efficiency on returning the stored information.

[12] Project's official site: http://portal.roda.dgarq.gov.pt/

In 2007, the work dedicated to research and development in Chronos Archiving project resulted in a CSP commercial enterprise solution, **Chronos**[13]. Chronos "provides an easy-to-manage Producer-Archive interface for relational databases, and it implements all components of an OAIS (ISO 14721) compliant archive system based on clearly documented procedures, open technology standards, and separation of components for ingest, data management, archival storage, and access" [BKM07].

In order to ensure future access and interpretation of database archives, Chronos extracts the data from the database and creates archives in text based format (ASCII/UTF to the primary data, XML to the metadata). Thus, the data stored in the archive can be accessed without any DBMS and even without the Chronos software. To access and retrieval of archived data can be used any Web browser, using full-text search as well as SQL queries.

Given the changes to the database schema and the need to allow continuously extraction from the production system over years, Chronos uses incremental archiving approach that applies the method of building a hierarchy of 'archive slices' where data is archived in multiple subsequent runs, thus archiving databases that continue to be in use.

3.6 Data warehouse Preservation

As the DBPreserve project, where this work is inserted, provides a model migration from the relational model of a database to a dimensional model, before being created a preserving format for the long-term, the analysis on data warehouse preservation is also considered here.

The research produced around digital preservation of databases does not provide the concepts introduced in the dimensional model. Concepts like facts, dimensions, bridges, hierarchies, levels, data marts, star schemas or snowflake schemas are essential for the full description of a data warehouse.

However, given the implementation of a data warehouse using technologies of relational data- bases, much effort has already produced can be used to describe a dimensional model. However, it is necessary to add a metadata layer that allows the complete characterization the data warehouse.

3.7 Summary and Conclusions

An understanding about the concepts of abstraction levels of a digital object (physical, logical and conceptual levels) gives a useful knowledge to contextualize the focus of each strategy for digital preservation.

[13] Product official site: http://www.csp-sw.de

Regarding to long-term preservation of relational databases, the key requirements identification allow the definition of the essential characteristics that an archive should contain to ensure data integrity and authenticity, database intelligibility, accessibility, processibility and originality.

The OAIS reference model introduces the appropriate terminology in the context of long-term preservation, as well as defining the functional components necessary to an archive implementation.

Digital preservation attracted many organizations to research and implement several preservation strategies according to its applicability (general or specific) in relation to its objective (physical/logical object preservation or conceptual object preservation). Among all the studied strategies, the one which has been most feasible and successful in databases preservation is data migration to a standard XML format. This format, being platform and application independent, simple text format and human readable, assumed as an effective method for long-term preservation of relational databases.

The presentation of the projects that focused on relational databases preservation provides a clarification about the problems and issues encountered, the most successful and effective methods, as well as the suitable long-term formats. As data warehouses can be implemented using relational database technologies, there are some efforts produced regarding relational database preservation that can be used and extended in order to fully characterize the dimensional model.

Chapter 4

DBPreserve OAIS Compliance

Section 3.3 presented the basic concepts associated with an OAIS. These systems have several entities with specific functions within the archive. This chapter is intended to outline some guidelines in order to conform the DBPreserve project, especially in the definition of the OAIS' information packages, the Submission Information Package (SIP), the Archival Information Package (AIP) and the Dissemination Information Package (DIP), to the OAIS model.

4.1 OAIS Functional Model Description

Although the creation of an OAIS prototype is an extensive job and is not an objective of this work, the study on OAIS regarding database preservation identified some important elements for its definition and so they are briefly described below.

The implementation of a digital archiving system capable of providing all the functionalities contained in the OAIS model (ingestion, management and access) [fSDS02], should be supported by metadata schemas that provides these functions. Metadata is categorized into descriptive, administrative and structural metadata [Pre04]. Other studies state that metadata can also have a technical category [Com11, BCF+07]. However, categories overlap and shift over time, and individual standards often address more than one category.

4.1.1 Descriptive Metadata

The descriptive metadata contains data that is vital for resource identification and discovery. There are several standards for descriptive metadata: MARC21 [oC00], Metadata Object Description Schema (MODS) [oC10], the Dublin Core Metadata Element Set [Cor10], the Content Standard for Digital Geospatial Metadata [Com98], the VRA Core

4.0 [Com07], the Data Documentation Initiative (DDI) [Ini09] and the Encoded Archival Description (EAD) [Gro02, GDO03].

As part of the objectives of this work, the access functionality should maintain compatibility with the language already used in the EAD files. Encoded Archival Description is an XML-based standard for preserving the hierarchy and describing the content of the objects held in the archive. It enables the consumers to categorize and locate the required information [Gro02].

4.1.2 Administrative Metadata

Administrative metadata provides information to support the management of a resource (creation date, type, grants...). Administrative metadata have several layer of administrative data, including the rights management metadata (relative to the intellectual property rights), and the preservation metadata (information needed to archive and preserve a resource) [Pre04].

In May 2005, the PREMIS (Preservation Metadata: Implementation Strategies) working group released a Data Dictionary for preservation metadata as well as a set of XML schemas to support implementation of the Data Dictionary in digital archiving systems. PREMIS Data Dictionary is now at version 2.1 [Com11]. The PREMIS Data Dictionary defines preservation metadata as the information a repository uses to support the digital preservation process (maintaining viability, renderability, understandability, authenticity, and identity in a preservation context. Thus, preservation metadata spans over administrative metadata (including rights and permissions), technical metadata and structural metadata. The entities in the PREMIS data model are shown in figure 4.1 and described below.

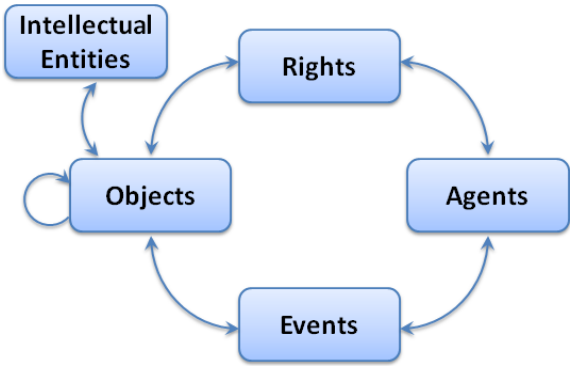

Figure 4.1: PREMIS data model

- Intellectual Entity - a set of content that is considered a single intellectual unit for purposes of management and description: for example, a particular book, map, photograph or database.

- Object (or Digital Object) - a discrete unit of information in digital form.

- Event - an action that involves or impacts at least one Object or Agent associated with or known by the preservation repository.

- Agent - person, organization or software program/system associated with Events in the life of an Object, or with Rights attached to an Object.

- Rights - assertions of one or more rights or permissions pertaining to an Object and/or Agent.

PREMIS does not detail the characteristics of Agents. However, standards such as MARC21 [oC00], vCard [Con97], MADS [oC11] and others, can be used to describe people, organizations and other entities that can act as Agents.

PREMIS Data Dictionary has the information needed to verify fixity, i.e., that an object is unchanged since some earlier point in time. In the PREMIS model, verifying the integrity of an object is considered an Event. If the representation includes the structural metadata, it can be used to test that all files are present and correctly named. Otherwise, the integrity of a representation may have to be verified by special programs that understand the structure of the representation. Authenticity includes both technical and procedural aspects. Regarding technical aspects, they can include maintenance of detailed documentation of digital provenance (object's history), the preservation of a version of the object that is, bit-wise, identical to the content as submitted, and the use of digital signatures [Com11].

4.1.3 Structural Metadata

The Structural Metadata describes the internal structure of digital resources and the relationships between their parts, indicating how compound objects are put together. Structural metadata can be used to represent the physical or logical structure of a complex object, enabling navigation and presentation of that object.

The Metadata Encoding & Transmition Standard (METS) is an XML standard that supports descriptive, administrative and structural metadata for digital objects. It provides the necessary metadata for the management of digital objects within a repository and for the exchange of such objects between archives or between archives and their consumers [Fed10]. It can be used to identify the compounds of an information package within an OAIS as well.

4.1.4 Technical Metadata

The technical metadata describes the physical rather than intellectual characteristics of digital objects. As a digital object is format-specific, technical metadata is clearly necessary for implementing most preservation strategies. Regarding databases, it describes the technology, physical characteristics of a database, table name, column name, data type, data dictionary, etc.

PREMIS has a restricted technical metadata, but it allows an external technical metadata schema reference if necessary. Regarding relational databases there are no common standards for technical metadata. However, this work uses formats that contain a metadata layer and contributes to the technical characterization for these kind of digital objects.

4.2 An Information Package

An information package is defined as a conceptual container consisting of two types of information, the *Content Information* (the information that is the initial target of preservation) and the *Preservation Description Information* (a set of information needed for the Content Information preservation), which are encapsulated and identified by *Packaging Information* (the information that allows the Content Information binding and associating with the PDI). The resulting package is labeled and recognized by the *Descriptive Information* (the information used to identify the package containing the desired Content Information). Regarding preservation metadata, a PDI have four aspects to consider: Provenance, Context, Reference and Fixity.

Meeting the needs to characterize an OAIS and assessing the suitability of the major metadata standards described above, as well as the formats for database preservation used in this project and described at Chapter 5 (SIARD and DWXML), it is possible to idealize what these information packages would be.

4.3 The Submission Information Package

The SIP is the package that is submitted to the OAIS by a Producer. In the scope of the DBPreserve project, the information content of a SIP is the relational database that the producer wants to preserve, both at the primary data and metadata levels. Metadata relates to primary data by describing and referencing it, making it meaningful and useful. Figure 4.2 shows the described draft of the SIP.

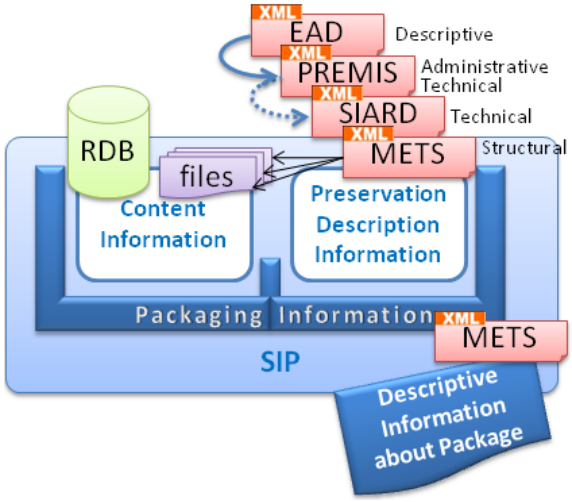

Figure 4.2: SIP proposed draft

The content information can be related to the relational database. The database can be a single file (e.g. MSAccess) or a database dump. The binary files corresponding to blobs (binary large objects) should also be submitted to the OAIS. Other files can be submitted to the archive by the producer that helps the interpretation and preservation process of the associated database (like reports, documents, UML schemas), and they can be identified by a METS file.

To describe the Content Information, the EAD standard can be used. The PREMIS standard is used for administrative metadata, to manage the preserving process and rights metadata, ensuring as well the Provenance, Reference and Fixity of the database. To collect the technical metadata of the database, the PREMIS standard can also point to a metadata SIARD representation of the original database. The structural information is implemented by the METS standard, referencing all the files involved in the submission. The wrapping and description of the SIP can also be achieved using a METS file.

However, the ingest process should comprised a set of tools to handle the submission of databases from different DBMS and related files, in order to enable the creation and packaging of the SIP.

4.4 The Archival Information Package

After the submission, one or more SIPs will be processed and transformed into one or more AIPs to be preserved. The DBPreserve project approach comprises a migration from the relational model to dimensional model, and then a migration to an XML-based model. During the project it was decided to maintain a SIARD representation of the original relational model. After the migration process to the data warehouse, it will be

represented with the extended SIARD format (described at Chapter 5). Figure 4.3 shows the described draft of the AIP.

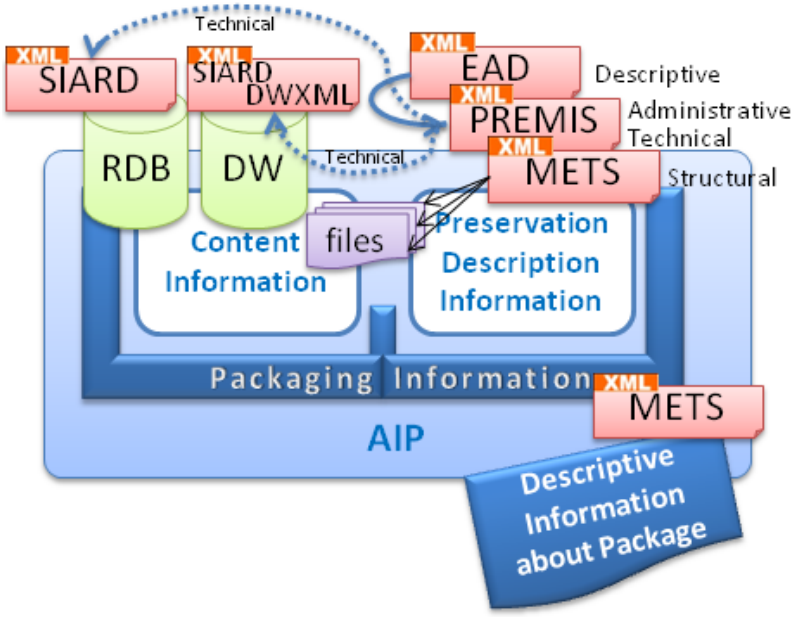

Figure 4.3: AIP proposed draft

So, the EAD describes the Content Information, as the PREMIS handles the administration metadata and technical metadata. The technical metadata can be enriched with the metadata layer from the SIARD representation of the relational database, as well as the dimensional metadata layer from the extended SIARD representation of the data warehouse. The related files will be described with METS, which also handles the packaging of the AIP.

Regarding the preservation of the original relational database with a SIARD format, another approach could be as follows. Instead of having an AIP with both representations, the SIARD for the relational database and the extended SIARD for the data warehouse, there might be two separated AIPs, one for each representation. Since the access to the primary data should be done through the XML-based representation of the data warehouse, the relational model will not be used for dissemination purposes.

4.5 The Dissemination Information Package

One of the most important functions of an OAIS is to ensure access to the preserved resources. So, the access and browsing of the primary data is essential to the success of a preservation project. Every request of a Consumer to an OAIS will be answered with a DIP, which can be derived from one or more AIPs. However, the output object can assume a large range of types (from an attribute value, to a record, a set of records, a table,

the whole database, or a SQL representation of the database to ensure its reactivation). These outputs should be described using the EAD standard. Nevertheless, having technical metadata can clarify the complexity of the retrieved package. This information can be found at the extended SIARD metadata level of the proposed AIP. Figure 4.4 shows the described draft of the DIP.

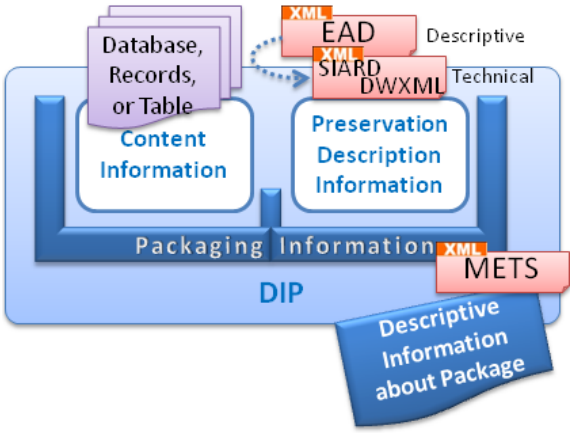

Figure 4.4: DIP proposed draft

Granting access to a resource only for those who should get it, is another problem that an OAIS should handle. Though PREMIS ensures the authenticity and provides some rights metadata, it isn't enough to ensure authentication functions. This enforcement should be implemented on the Access entity of the OAIS.

4.6 Summary and Conclusions

This chapter disserts on the OAIS requirements, first analyzing the genres of metadata needed to support an archival system of this type, and then defining possible information packages drafts for the various preservation stages: ingest, archive and access. An OAIS requires four different types of metadata: descriptive, administrative, structural and technical metadata.

The ingest process should be handled by an application that supports the submission and building of the SIP, packaging the content information to preserve and describing it with metadata. The archival storage will manage the AIPs, built from one or more SIPs. The proposed AIP preserves the relational database through its SIARD representation, as well as the extended SIARD representation of the data warehouse.

The access to the primary data should be secured properly to the people allowed to get the resources, and ensure the correct DIP interpretation, allowing the description with AED standard, helped with some metadata layer from the AIP.

This analysis aims to determine the emerging needs to implement an OAIS. However, not all entities or requirements are evaluated here. Only those that contribute to the definition of the SIP, AIP and DIP, were referenced. Though the information packages are identified and described with published standards, a real implementation of an OAIS can impose other important requirements for the characterization of the SIP, AIP or DIP.

Chapter 5

Data Warehouse Preservation Format

The DBPreserve project approaches the long-term preservation of relational databases by means of a two step migration:

- a model migration, from the relational model to a dimensional model, using data warehousing concepts for model and analysis simplification;
- an XML migration, from the dimensional model to an XML based format that represents the data warehouse, to ensure a long-term preservation format.

Focusing on the second step of the DBPreserve project approach on long-term preservation of relational databases, we define the *Data Warehouse Extensible Markup Language (DWXML)*, an XML-based neutral format that fully describes the dimensional model produced in the first step. This format should hold all the relevant metadata regarding the dimensional model as well the primary data stored in relational tables.

5.1 The data warehouse as a digital object

Databases and data warehouses are different from conventional digital objects. Data warehouses are often implemented on relational databases (ROLAP), keeping the data in tables, views and schemas. Data warehouses are indeed complex digital objects, based on a dimensional model, where star and snowflake schemas, facts, dimensions with levels and hierarchies, bridges tables and datamarts can be identified.

Regarding databases and data warehouses implemented using relational database technologies, there is the concern to preserve the structure (metadata about tables, attributes, primary and foreign keys, constraints) and the primary data (the data stored in the tables). The access to the metadata that characterize the structure depends on the DBMS used to

implement the database or the data warehouse, because each DBMS has a singular approach on storing the structural description. This process requires different mediators to access databases implemented with different DBMS. Collecting the primary data is an easier task as almost every DBMS allows the conversion of the data into pure text files for interoperability purposes. However, using mediators to connect to the database, it is possible to query the tables and collect the data for the migration process.

Databases and data warehouses have the data stored in tables, columns (attributes) and rows (records). This organization is an example of structured information, where accessing and reusing the data is efficient. However, the lack of information about the order of the records can be an issue if the order of the data is to be preserved.

5.2 Analysis of preservation formats for relational databases

In order to propose a preservation format for data warehouses implemented using relational databa- se technologies, a format that preserves the metadata, both at the relational and dimensional levels, as well as the primary data of a data warehouse, a detailed inspection on the established formats for relational database preservation took place, in order to determine if these formats could be used or extended to accommodate the dimensional model concepts. The analysis focused in the DBML and SIARD formats.

5.2.1 Relational Database Preservation using DBML

The DBML is an XML representation of a relational database for preservation purposes. This format represents the structure and the primary data in a single XML file and was used on the RODA project [RFFC07, BCF+07]. DBML simplifies the data traversal, as just a single file needs to be searched for dissemination purposes. However, it is common that databases and data warehouses have a high volume of data. Migrating large volumes of data produces even larger XML files. Collecting the data into a single XML file brings the usual problems about large XML document processing and querying efficiency [WTF03].

The following example is a sample of a DBML file. It is visible that structural definition is apart from the data itself, but all the database fits into an XML file. The `STRUCTURE` element holds the definition of the schema of the database, adding metadata for tables, columns and keys, using element and attributes.

Example of a DBML file

```
<?xml version="1.0" ?>
<DB>
  <STRUCTURE>
    <TABLE NAME="IPDW_PROFESSOR">
      <COLUMNS>
        <COLUMN NAME="PROFESSOR_ID" TYPE="decimal" SIZE="22" NULL="no"/>
```

```
        <COLUMN NAME="CODE" TYPE="decimal" SIZE="22" NULL="no"/>
        <COLUMN NAME="ACRONYM" TYPE="varchar2" SIZE="8" NULL="no"/>
        ...
      </COLUMNS>
      <KEYS>
        <PKEY TYPE="simple">
        <FIELD NAME="PROFESSOR_ID"/>
      </PKEY>
      </KEYS>
    </TABLE>
    <TABLE NAME="IPDW_QUIZ">
      ...
    </TABLE>
  </STRUCTURE>
  <DATA>
    <IPDW_PROFESSOR>
      <IPDW_PROFESSOR-REG>
        <PROFESSOR_ID> 145 </PROFESSOR_ID>
        <CODE> 208741 </CODE>
        <ACRONYM> GTD </ACRONYM>
        ...
      </IPDW_PROFESSOR-REG>
      <IPDW_PROFESSOR-REG>
        ...
      </IPDW_PROFESSOR-REG>
    </IPDW_PROFESSOR>
  ...
  </DATA>
</DB>
```

The element `DATA` contains the primary data of each row of each table. The table elements are named after the tables as well each record element, identified by the suffix `-REG` after the name of the table where it belongs. Inside this element there are a sequence of elements that represents the columns of the table and the text of the element is the value of that attribute in that record. Using the approach of labeling the elements using the names of the tables and columns makes the XML file easier to read but this introduces a validation problem of the XML schema, because the XML schema depends on the schema of the database. So, for each DBML file, a new XSD schema has to be created according to the database's structural description.

5.2.2 Relational Database Preservation using SIARD

The SIARD format is a nonproprietary and published open standard, based on other open standards (e.g., ISO standard Unicode, XML, SQL1999, and the industry standard ZIP. It was developed by the Swiss Federal Archives. In May 2008, the European PLANETS project accepted SIARD format as the official format for archiving relational databases [IP08].

The SIARD file is a ZIP64[1] [PKW07] uncompressed package based on an organizational system of folders, storing the metadata in the `header` folder and table data in the `content` folder. This organization is shown in figure 5.1.

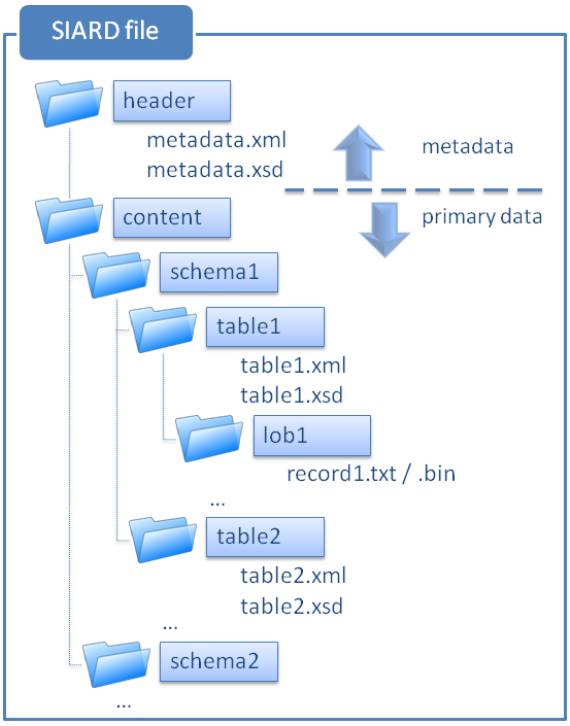

Figure 5.1: Structure of the SIARD Archive File

The metadata is gathered in a single file that fully describes the database structure: schemas, tables, columns, keys, views, functions, users, privileges and roles. The folder `metadata` also contains the XSD schema for `metadata.xml` XML file validation.

The following example shows an extract of the `metadata.xml` file.

Example of the SIARD metadata file

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="metadata.xsl"?>

<siardArchive
 xmlns="http://www.bar.admin.ch/xmlns/siard/1.0/metadata.xsd"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0"
 xsi:schemaLocation="http://www.bar.admin.ch/xmlns/siard/1.0/metadata.xsd metadata.xsd">
  <dbname>inqueritos</dbname>
  ...
  <databaseUser>CALDEIAS</databaseUser>
  <schemas>
    <schema>
      <name>CALDEIAS</name>
      <folder>schema0</folder>
```

[1]The use of ZIP64 standard was due to exceed the limit of 4GB package maximum size in the standard ZIP. Nowadays, JDK7 already supports the ZIP64 standard.

```
      <tables>
        <table>
          <name>IPDW_PROFESSOR</name>
          <folder>table4</folder>
          <description/>
          <columns>
            <column>
              <name>PROFESSOR_ID</name>
              <type>DECIMAL(22)</type>
              <typeOriginal>INTEGER</typeOriginal>
              <nullable>false</nullable>
            </column>
            <column>
              <name>CODE</name>
              <type>DECIMAL(22)</type>
              <typeOriginal>INTEGER</typeOriginal>
              <nullable>false</nullable>
            </column>
            <column>
              <name>ACRONYM</name>
              <type>CHARACTER VARYING(8)</type>
              <typeOriginal>VARCHAR2(8)</typeOriginal>
              <nullable>true</nullable>
            </column>
            ...
          <columns>
          <primaryKey>
            <name>SYS_C0025947</name>
            <column>PROFESSOR_ID</column>
          </primaryKey>
          <rows>557</rows>
        </table>
        ...
      </tables>
    </schema>
  </schemas>
  <users> ...
  </users>
  <roles> ...
  </roles>
  <privileges> ...
  </privileges>
</siardArchive>
```

As to the primary data, each database schema is stored in a separated folder sequentially numbered, as well as the tables of each schema. The data from each table is stored in an XML file with simplified structure (only rows and columns) and its XSD. If there are Large Objects - LOB (BLOB - Binary Large Objects and CLOB - Character Large Objects), these data are stored in binary files or text, within a folder for each attribute of these types, as is referenced by its path in the XML for the corresponding table.

The following example illustrates how the primary data is stored in XML format. The table on quest is `table4` from `schema0` (zero). So the pathname to the `table.xml` inside the ZIP64 SIARD file is `/content/schema0/table4/table4.xml`.

Example of a SIARD primary data file

```
<?xml version="1.0" encoding="utf-8"?>
<table
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.admin.ch/xmlns/siard/1.0/schema0/table4.xsd"
  xsi:schemaLocation="http://www.admin.ch/xmlns/siard/1.0/schema0/table4.xsd table4.xsd">
  <row><c1>145</c1><c2>208741</c2><c3>GTD</c3><c4>Gabriel David</c4></row>
  <row><c1>185</c1><c2>209566</c2><c3>MCR</c3><c4>Cristina Ribeiro</c4></row>
  ...
</table>
```

This simple representation of the primary data resembles the data in tables, where each record is a row with several columns, the attributes. XML schemas for the tables (`tableN.xsd`) is similar with each other, varying only on the number and name of the columns. One of the major benefits of this segmented archiving of the primary data is that it will reduce the size of each XML file, because the data will be distributed into their respective XML table files. This will increase the efficiency of parsing and querying of the XML data and can be extremely useful for parsing and querying of simultaneous XML table files, in order to solve to a query involving more than one XML file (table).

5.3 Extending the SIARD format with dimensional model metadata

The inherent modularity of data warehouses, with independent stars sharing some dimensions, led to the choice of a format that also supports a segmented structure of the primary data. The SIARD format proved to be the most appropriate starting point for the representation of a data warehouse in XML. This modular format is appropriate for the handling of data. The XML file sizes will be manageable, reflecting an increase on parsing and querying efficiency. The suitability for handling several XML files helps processing more complex queries.

Thus, reusing the effort to define an archive format that stores the definition of the relational tables and their primary data, this work proposes to extend the SIARD format, adding a metadata layer for data interpretation according to the data warehouse perspective. This extra metadata layer must accommodate the dimensional model concepts described at Section 2.2.1.

Another reason for using the SIARD format is the existence of a tool which already allows us to create these packages from relational databases created in Oracle, MSAccess and MSSQLServer. The SIARD project produced a set of tools named SIARD Suite. One of them, the `SIARDfromDB`, migrates the relational metadata and the primary data from a supported DBMS to the SIARD format. Thus, the effort in this work will focus on the description of the dimensional model, complementing the existing one for the relational metadata format. The process of migration the primary data into an XML format according to SIARD has also to be ensured.

However, the reuse and expansion of an existing open format like SIARD should not prevent the use of the applications that supports it, the SIARD Suite. Existing applications should be executed as if no changes to the format were made. Thus, using SIARD Suite it must be possible to manage the relational level metadata and the primary data.

To accomplish the goal of adding dimensional model metadata to the SIARD format, we identified two different approaches: alter the `metadata.xml` and associated schema at the `hea- der`folder, adding the necessary dimensional model metadata; or adding a new metadata file that describes the dimensional model. None of these hypotheses will interfere with the integrity of the primary data, certified by an hexadecimal digest message code over the `content` folder with a prefix which indicates the type of the Digest-Algorithm (MD5[2] or SHA1[3]) [IP08]. This digest message is stored in the `messageDigest` element at the database level metadata, within the `metadata.xml` file.

The first approach seemed promising, because changing just the metadata file would concatenate the characterization of the relational model and the dimensional model to a single file. So, after adding some new XML elements for testing in the SIARD metadata file, the SIARD Suite was executed with no problems. As expected, the new elements were ignored, and it was still possible to use all the features of this application. Nevertheless, after applying some changes to the metadata, a direct inspection of the XML metadata file verified that the newly-added XML elements had vanished. So, altering the metadata via SIARD Suite, triggers the full compilation of the metadata file according to the related schema, where there is no representation of the dimensional model. This path had to be abandoned.

The option to add a new XML file that describes the dimensional model of a data warehouse was then the path adopted in this work. Similarly to the previous approach, an XML file for testing was added to the `header` folder to analyze the behavior of SIARD Suite. This application, that supports the creation, metadata editing and reactivation of the database from its representation in the SIARD format, functioned normally, and even after changing some relational metadata the extra file was not deleted.

Thus, to provide dimensional metadata to the SIARD format, a new XML file will be added that characterizes a data warehouse and provides the concepts associated with the dimensional model, not covered by the standard SIARD format. The XML schema (XSD) will also be added for validation of the XML file produced. This new XML representation was named Data Warehouse Extensible Markup Language (DWXML) [ADR11]. The language is described in the following section.

Figure 5.2 shows an excerpt of the extended SIARD format, bearing the description of a data warehouse.

[2]MD5 - Message-Digest Algorithm 5
[3]SHA - Secure Hash Algorithm

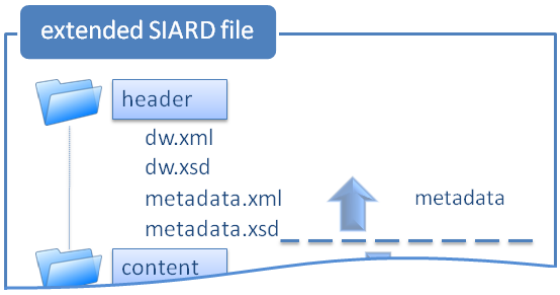

Figure 5.2: Extended SIARD Archive File

5.4 DWXML Schema Definiton

As XML is widely used across several platforms, a research on data warehouse XML representations was done. Indeed, there are some works in this area [HBH03, JMP01]. These works concern a multidimensional schema representation, i.e. data cubes. The XCube [HBH03] is a data cube XML representation and it was developed to exchange data warehouse data over any kind of network. The XCube was designed for MOLAP systems for interoperability purposes. The representation of the cube is divided to several XML documents to characterize each entity involved in the multidimensional system. This approach is interesting in the context in which it was developed, as it allows slicing the cube and sending small packets of information over the network, just what is requested by the client. Even trying to adapt it for dimensional models, the diversity of documents produced would obscure the representation of the data warehouse. Moreover, it doesn't have any reference to tables (which store the facts and the dimensions in ROLAP systems), views, star or snowflake schemas.

Regarding the SIARD format extension for archiving data warehouses in ROLAP systems, the proposed XML accounts for the description of dimensional model, adding a metadata file (`dw.xml`) and its schema definition (`dw.xsd`[4]). The Data Warehouse Extensible Markup Language intends to be an independent representation of the metadata of a data warehouse. However, in this project's context it will be inserted into the SIARD format, resulting in a long-term preservation format for data warehouses implemented using relational database technologies (ROLAP).

The data warehouse is characterized as a set of stars and a set of dimensions, represented in tables and views organized in schemas. We also envisage a representation of data marts as a set of stars. Figure 5.3 characterizes the DWXML basic structure and the `star` element.

The attribute `version` represents the version of the DWXML definition. The element `dwBinding` supports the description of the DWXML file, the information related to the

[4] https://www.fe.up.pt/si/wikis_paginas_geral.paginas_view?pct_pagina=42633

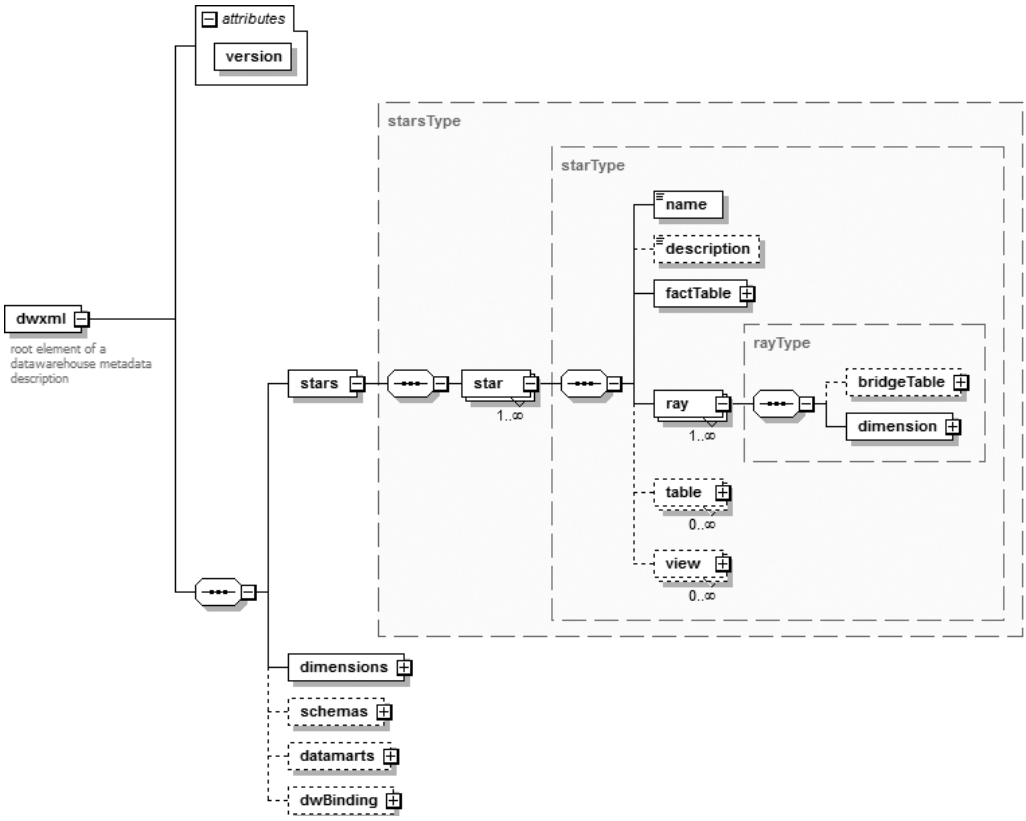

Figure 5.3: DWXML schema showing the star element

owner of the data, the credentials of the connection to the data warehouse and the names and versions of the applications involved in the DWXML creation, including the DBMS where the data warehouse was working and the migration date.

Table 5.1 describes the identifiers of the data warehouse metadata. The column *Opt.* just indicates if the identifier is optional or not.

Table 5.1: Data warehouse metadata description

Identifier	**Opt.**	**Description**
version	no	DWXML format version
stars	no	List of stars in the data warehouse
dimensions	no	List of dimensions, dimensional tables and views in the data warehouse
schemas	yes	List of schemas in the data warehouse
datamart	yes	List of datamarts in the data warehouse
dwBinding	yes	Additional metadata for data warehouse connection description and DWXML file generation

Table 5.2 describes the identifiers of the data warehouse binding metadata.

Table 5.2: Data warehouse binding metadata description

Identifier	Opt.	Description
description	yes	Description of the data warehouse's meaning and content
dataOwner	no	The owner of the data, who has the right to grant the access to the data
xmlApplication	yes	Name and version of program that produced the DWXML from the data warehouse
migrationDate	yes	Date when the DWXML was produced from the data warehouse
dwProduct	no	The data warehouse product and version of the data warehouse that contains the dimensional model
dwUser	no	The user to the data warehouse who carried out the XML migration
dwConnection	no	The connection string to the data warehouse that contains the dimensional model

5.4.1 Stars and Facts

A star is composed of a fact table and a set of rays which establish relationships to dimensions and possibly bridge tables. The `factTable` element references the respective table description in the `schemas` element, indicates the columns responsible for the joins between fact tables and bridge tables or dimensions and contains information about its granularity and about the facts. Regarding the facts, these elements indicate the table's column that represents them, as well as their measure type: non-additive, semi-additive or additive. Table 5.3 describes the identifiers of the star metadata.

Table 5.3: Star metadata description

Identifier	Opt.	Description
name	no	Name of the star
description	yes	Description of the star's meaning and content
factTable	no	Fact table of the star
ray	no	Ray of the star connecting the fact table with a bridge table and/or a dimension (referenced by schema and name).
table	yes	List of extra tables to accommodate unexpected special cases (referenced by schema and table name)
view	yes	List of extra views to accommodate unexpected special cases (referenced by schema and view name)

Figure 5.4 shows the schema of a fact table element, its facts and join column definition.

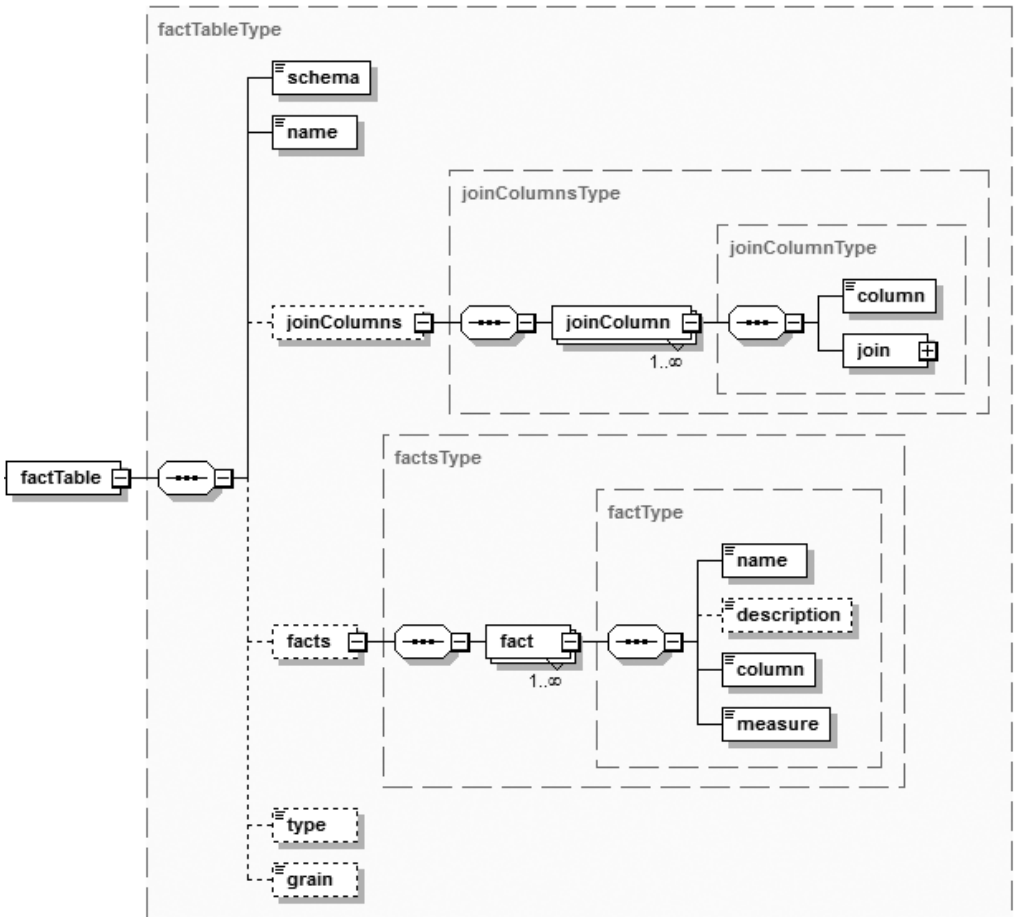

Figure 5.4: Schema of a fact table element

Table 5.4 describes the identifiers of the fact table metadata.

Table 5.4: Fact table metadata description

Identifier	**Opt.**	**Description**
schema	no	Schema of the fact table
name	no	Name of the fact table
joinColumns	yes	List of columns used in the join between a fact table and a bridge table (if it exists)
facts	yes	List of facts represented in the fact table
type	yes	The type of the fact table (CUMULATIVE or SNAPSHOT)
grain	yes	The grain of the fact, the meaning and content of a row in the fact table

Join columns are useful to indicate which column of the fact table is responsible for the relationship with a column of a bridge table. Table 5.5 describes the identifiers of the join column metadata.

Table 5.5: Join column metadata description

Identifier	Opt.	Description
column	no	Name of the column of fact table used in the join
join	no	Column of the bridge table referenced on the join (schema, table and column)

Table 5.6 describes the identifiers of the fact metadata.

Table 5.6: Fact metadata description

Identifier	Opt.	Description
name	no	Name of the fact
description	yes	Description of the fact's meaning and content
column	no	Column of the fact table where the fact is stored
measure	no	Measure type (constrained to ADDITIVE, NON ADDITIVE or SEMI ADDITIVE)

In a star, each `ray` element represents a relationship between the fact table and the dimension. If there is a many to many relationship between the fact table and the dimension table, a bridge table may be added. In this case, the `ray` element would be composed by a `bridgeTable` element that references the related table, followed by the `dimension` element that represents a reference to the dimension. Table 5.7 describes the identifiers of the ray metadata.

Table 5.7: Ray metadata description

Identifier	Opt.	Description
bridgeTable	yes	Reference to the bridge table (schema and name)
dimension	no	Reference to the dimension (schema and name)

The following example shows a star definition using DWXML. The `IPDW_ANSWERS_STAR` is composed by the `IPDW_ANSWERS` fact table, that hold the data of the additive fact `ANSWER_F` represented on the fact table's column `ANSWER`, and by two ray elements. One of them establishes a connection to the dimension `QUESTION_DIM`.

Example of a DWXML star definition

```
<?xml version="1.0" encoding="UTF-8"?>
<dwxml version="1.0" xsi:noNamespaceSchemaLocation="dw.xsd"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <stars>
   <star>
     <name>IPDW_ANSWERS_STAR</name>
     <description>Star related to the answers</description>
     <factTable>
       <schema>CALDEIAS</schema>
       <name>IPDW_ANSWERS</name>
       <facts>
         <fact>
           <name>ANSWER_F</name>
           <column>ANSWER</column>
           <measure>ADDITIVE</measure>
         </fact>
       </facts>
     </factTable>
     <ray>
       <dimension>
         <schema>CALDEIAS</schema>
         <name>QUESTION_DIM</name>
       </dimension>
     </ray>
     <ray>
       ...
     </ray>
   </star>
 </stars>
 ...
</dwxml>
```

As snowflake schemas are quite similar to star schemas, their representation starts as a star schema, but the dimensions of a snowflake schema are implemented by tables (dimension tables) that are partially normalized, resulting in relationships with other tables (sub-dimension). So, inspecting the dimension table's foreign keys, it is possible to differentiate between a snowflake schema and a star schema. If a foreign key of a dimension table refers a sub-dimension, the schema is a snowflake schema. Datamarts are subsets of data warehouses, i.e. sets of star or snowflake schemas.

Table 5.8 describes the identifiers of the datamart metadata.

Table 5.8: Datamart metadata description

Identifier	**Opt.**	**Description**
name	no	Name of the datamart
description	yes	Description of the datamart's meaning and content
stars	no	List of the name of the stars that defines the datamart

5.4.2 Dimensions

The dimension element was defined following the sintax of the `CREATE DIMENSION` Oracle SQL statement [Ora03]. The metadata related to the dimensions is stored in sep-

arated `dimension` elements and allows the categorization and description of the facts and measures in order to support meaningful answers to the requested questions. Each `dimension` element describes the levels and respective level keys, the level hierarchies and the attributes defined at each level. The `tables` and `views` elements contain the references to the tables and views (schema and name) that support the declared dimensions; their structure is described in the `schemas` element.

Figure 5.5 displays the `dimensions` element schema.

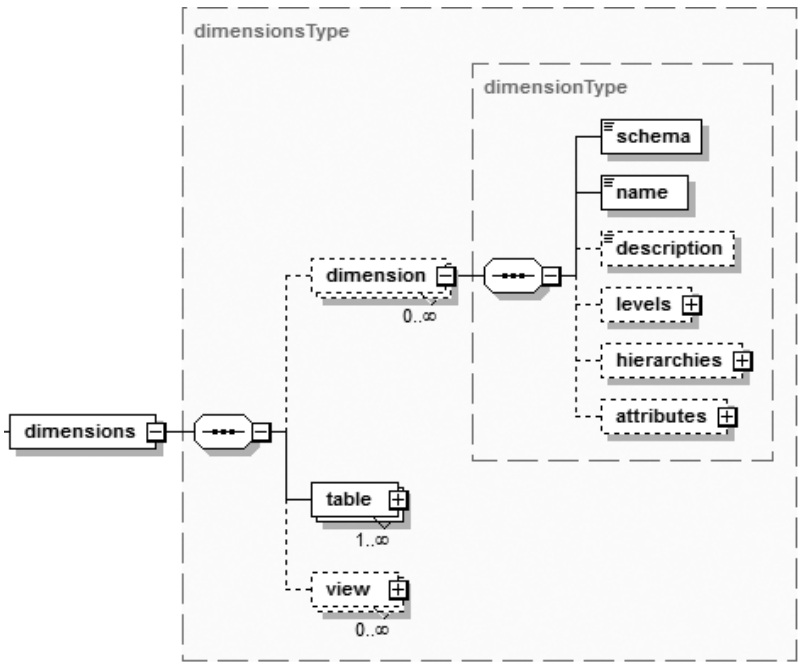

Figure 5.5: Dimensions element schema

Table 5.9 describes the identifiers of the dimension metadata.

Table 5.9: Dimension metadata description

Identifier	Opt.	Description
schema	no	Schema of the dimension
name	no	Name of the dimension
description	yes	Description of the dimension's meaning and content
levels	yes	List of levels in the dimension
hierarchies	yes	List of hierarchies in the dimension
attributes	yes	List of attributes in the dimension

Figure 5.6 displays the `level` element schema in a dimension. Levels have a level key that identifies each level. This key is typically just one column of the dimension table that represents the dimension in the data warehouse.

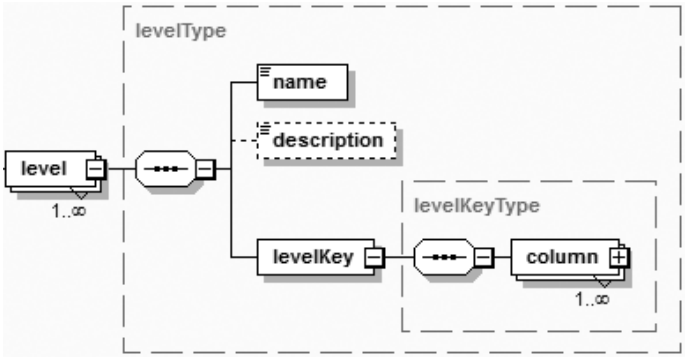

Figure 5.6: Level element schema

Table 5.10 describes the identifiers of the level metadata.

Table 5.10: Level metadata description

Identifier	Opt.	Description
name	no	Name of the level
description	yes	Description of the level's meaning and content
levelKey	no	Key of the level (one or more columns in the dimension table)

Figure 5.7 displays the `hierarchy` element schema in a dimension.

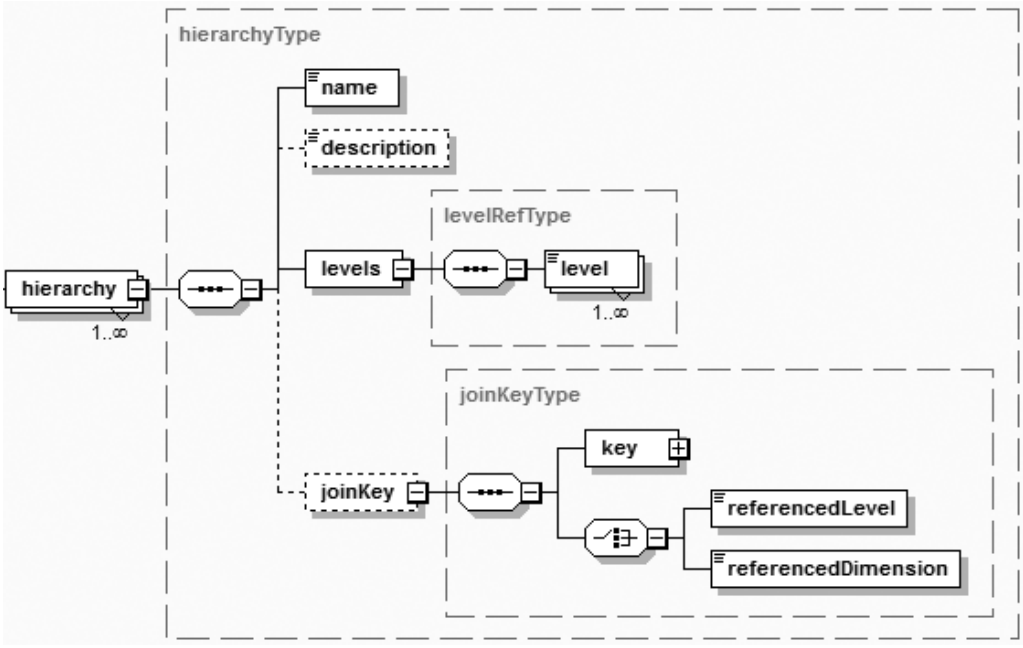

Figure 5.7: Hierarchy element schema

Table 5.11 describes the identifiers of the hierarchy metadata.

Table 5.11: Hierarchy metadata description

Identifier	Opt.	Description
name	no	Name of the hierarchy
description	yes	Description of the hierarchy's meaning and content
levels	no	List of the levels of the hierarchy. The hierarchy is settled according to their order of appearance of the levels (the first level is parent of the second that is parent of the third...).
joinKey	yes	The key that joins the levels of the hierarchy when they are implemented using different dimension tables.

Table 5.12 describes the identifiers of the attribute metadata. Attributes are characteristics of a level, identified by its level key.

Table 5.12: Attribute metadata description

Identifier	Opt.	Description
attributeName	yes	Name of the attribute
level	no	Identifies the level and the attributes determined by its level key. There have to be at least one level element

Figure 5.8 displays the `attribute` element schema in a dimension.

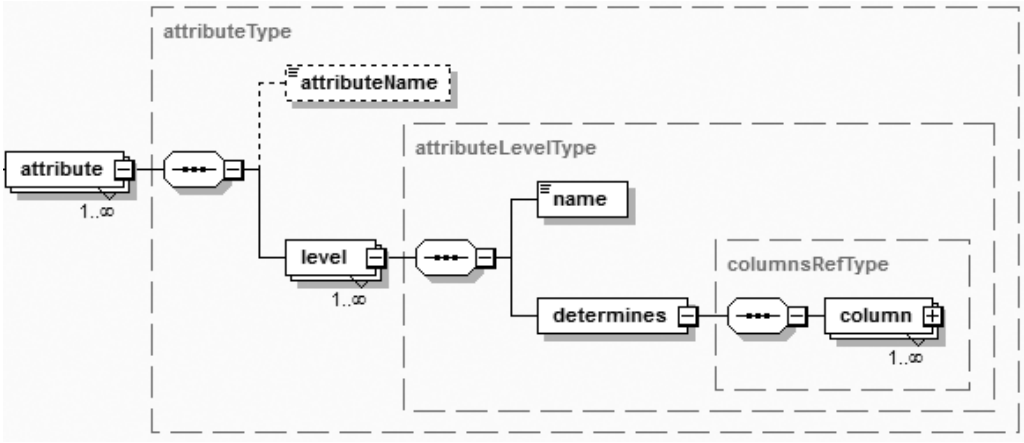

Figure 5.8: Attribute element schema

Table 5.13 describes the identifiers of the level referenced by an attribute.

Table 5.13: Attribute levels metadata description

Identifier	Opt.	Description
name	no	Name of the level
determines	no	Atribute determined by the level

The next example shows a dimension definition using DWXML. This dimension (CLASS_DIM) is composed by the levels COURSE and CLASS, identified by their level keys COURSE_ID and CLASS_ID, respectively. Both these levels are implemented by the IPDW_CLASS relational table. The dimension has a defined hierarchy of levels named (CLASS_ROLLUP) that states the CLASS level is child of the COURSE level, according to the order of appearance (the first is parent of the second and so on). The attribute element defines which attributes are defined by each level. So, the attributes COUR_PREVIOUS_COD, COUR_TYPE, COUR_NAME, COUR_ACRONYM belong to the COURSE level and the attributes TYPE, NAME, ACRONYM belong to the CLASS level.

Example of a DWXML dimension definition

```
<?xml version="1.0" encoding="UTF-8"?>
<dwxml version="1.0" xsi:noNamespaceSchemaLocation="dw.xsd"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  ...
  <dimensions>
    <dimension>
      <schema>CALDEIAS</schema>
      <name>CLASS_DIM</name>
      <levels>
        <level>
          <name>CLASS</name>
          <description />
          <levelKey>
            <column>
              <schema>CALDEIAS</schema>
              <table>IPDW_CLASS</table>
              <column>CLASS_ID</column>
            </column>
          </levelKey>
        </level>
        <level>
          <name>COURSE</name>
          <levelKey>
            <column>
              <schema>CALDEIAS</schema>
              <table>IPDW_CLASS</table>
              <column>COURSE_ID</column>
            </column>
          </levelKey>
        </level>
      </levels>
      <hierarchies>
```

```
        <hierarchy>
          <name>CLASS_ROLLUP</name>
          <levels>
            <level>COURSE</level>
            <level>CLASS</level>
          </levels>
        </hierarchy>
      </hierarchies>
      <attributes>
        <attribute>
          <attributeName>COURSE</attributeName>
          <level>
            <name>COURSE</name>
            <determines>
              <column>
                <schema>CALDEIAS</schema>
                <table>IPDW_CLASS</table>
                <column>COUR_PREVIOUS_COD</column>
              </column>
              <column>
                <schema>CALDEIAS</schema>
                <table>IPDW_CLASS</table>
                <column>COUR_TYPE</column>
              </column>
              ...
            </determines>
          </level>
        </attribute>
        <attribute>
        ...
        </attribute>
      </attributes>
    </dimension>
    ...
  </dimensions>
  ...
</dwxml>
```

5.4.3 Tables and Views

The schemas, tables and views follow a simplified representation to the SIARD format and some elements are replicated in this description to permit the full characterization of a data warehouse metadata regardless of whether there is a SIARD package or not. However, this DWXML version does not contemplate the representation of the primary data in XML, since it is used in conjunction with the SIARD format, which already performs the primary data migration to XML format.

A schema contains group of tables and a group of views. Figure 5.9 displays the `schema` element.

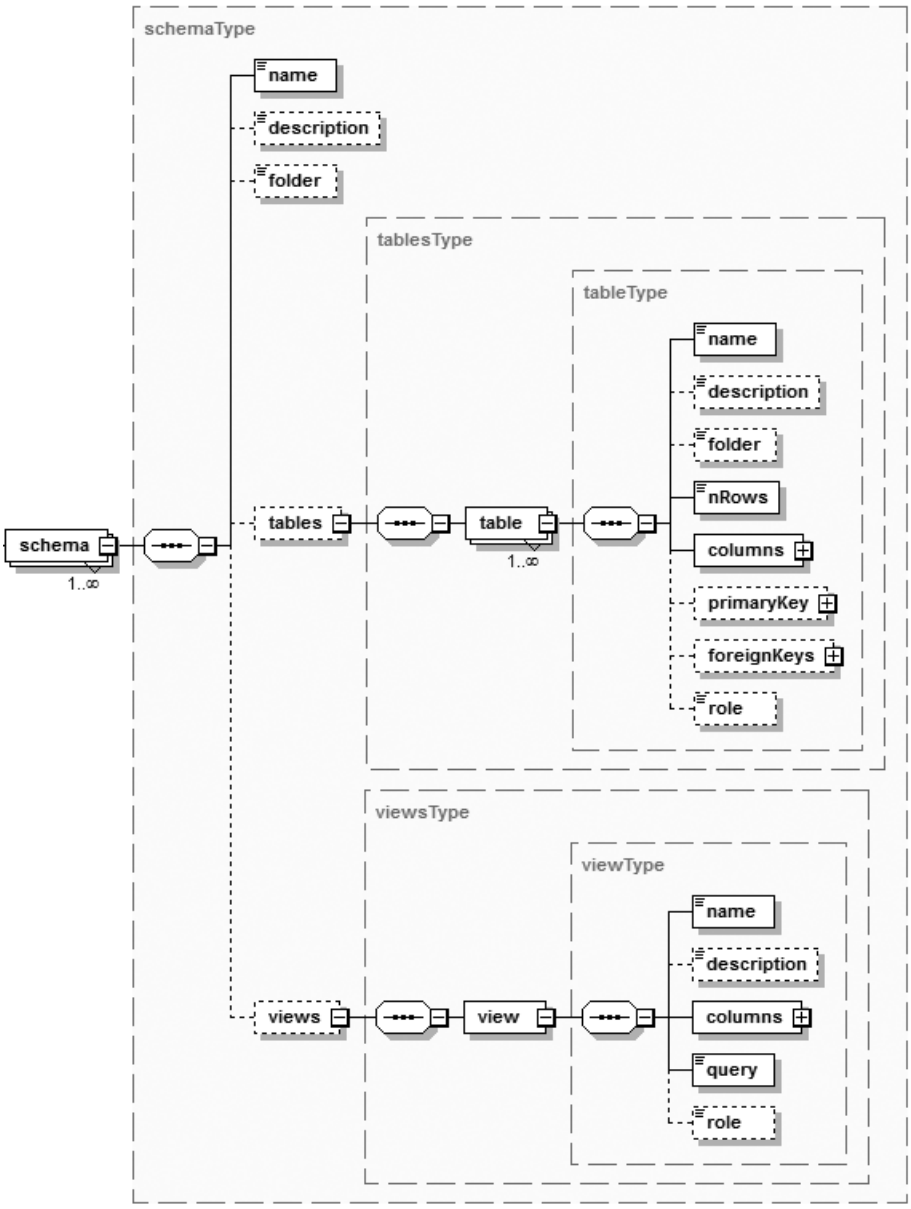

Figure 5.9: The schema element

Table 5.14 describes the identifiers of the schema metadata.

Table 5.14: Schema metadata description

Identifier	Opt.	Description
name	no	Name of the schema
description	yes	Description of the schema's meaning and content
folder	yes	The name of the folder in the SIARD format
tables	yes	List of tables of the schema and their definition
views	yes	List of views of the schema and their definition

Table 5.15 describes the identifiers of the table metadata.

Table 5.15: Table metadata description

Identifier	Opt.	Description
name	no	Name of the table
description	yes	Description of the table's meaning and content
folder	yes	The name of the folder in the SIARD format
nRows	no	Number of rows of the table
columns	no	List of columns of the table and their definition
primaryKey	yes	The primary key of the table
foreignKeys	yes	List of foreign keys of the table and their definition
role	yes	The role of the table in the dimensional model

Table 5.16 describes the identifiers of the column metadata.

Table 5.16: Column metadata description

Identifier	Opt.	Description
name	no	Name of the column
description	yes	Description of the column's meaning and content
folder	yes	The name of the folder in the SIARD format for LOBs storage
type	yes	Data type of the column in the data warehouse
defaultValue	yes	Default value of the column
nullable	no	Indicates if the column value can be null

Table 5.17 describes the identifiers of the primary key metadata.

Table 5.17: Primary key metadata description

Identifier	Opt.	Description
name	no	Name of the primary key
description	yes	Description of the primary key's meaning and content
column	no	Column that belongs to the primary key. There have to be at least one column element

Table 5.18 describes the identifiers of the foreign key metadata.

Table 5.18: Foreign key metadata description

Identifier	Opt.	Description
name	no	Name of the foreign key
description	yes	Description of the foreign key's meaning and content
referencedSchema	no	Name of the schema of the referenced table
referencedTable	no	Name of the referenced table
reference	no	Name of the referencing column and referenced column. There have to be at least one reference element

Table 5.19 describes the identifiers of the view metadata.

Table 5.19: View metadata description

Identifier	Opt.	Description
name	no	Name of the view
description	yes	Description of the view's meaning and content
columns	no	List of the columns of the view (schema, table and column names)
query	no	Query that represent the view
role	no	The role of the view in the dimensional model

5.5 Summary and Conclusions

The DBPreserve project migrates relational databases to a dimensional model in order to use the data warehouse technologies for relational database preservation. However, the use of these platforms maintains a technological dependence, as data warehouses implemented using relational databases tools are supported by DBMS. Thus, the choice of a representation of the data warehouse using XML allows the existence of a text-based, common format, a digital preservation format in the long term.

Looking at the data warehouse as a digital object, there is no intention to archive its look and feel, but there is the need to store its representation (stars, dimensions, levels,...), i.e. the data warehouse metadata, along with the primary data.

To achieve a data warehouse preservation format, we analyzed the existing formats for relational databases preservation, and studied some XML representations of multidimensional OLAP. The chosen path was the expansion of the SIARD format, a format for preservation of relational databases, adopted by the European project PLANETS. The proposed XML dialect, the DWXML, intends to add a metadata layer to adapt the SIARD

format to accommodate the data warehousing concepts at a dimensional model level (ROLAP). The DWXML schema description is presented in this chapter and was published in a paper on the XATA2011 workshop proceedings, on June 1 2011[ADR11] (see appendix A).

Chapter 6

DBPreserve Suite Application

After the study and definition of a preservation format for a data warehouse, another objective of this project was to develop an application that eases the migrating process of metadata and primary data to the proposed XML based format - the SIARD format extended with the DWXML.

This chapter reports the work done on the implementation of the DBPreserve Suite application. To implement this tool, recent technologies have been chosen to maintain it fully operational in the current technological context. However, the focus of this project is merely on relational databases preservation while support for dimensional models, allowing future access to the preservation format used in the archive. The SIARD format extended with the DWXML holds the relational model metadata, the dimensional model metadata as well as the primary data. Thus, there is no concern about keeping this tool alive in the long-term, but just to keep it running during the technological context in which it was developed. Through analysis of the produced documentation on the proposed preservation format, it is possible to create or replicate later on an application that allows management and access to this format.

6.1 General Requirements

In the initial phase of the project the requirements for the application to build were rather vague and broad, because there was still a long way to follow in defining the preservation format. In general, this application should migrate the primary data and the dimensional model metadata to an XML-based format, yet to define. The study about the preservation of relational databases and the developing of a format for long-term preservation that would accommodate the relational model and the dimensional model concepts, enabled the clarification and definition of the application to develop. So, after the decision of

extending the SIARD format with the DWXML, the requirements of the DBPreserve Suite application were clearly defined. The DBPreserve Suite application should:

- Migrate the data warehouse model implemented using relational database technologies to the SIARD format;
- Acquire the metadata to describe the dimensional model of the data warehouse;
- Automate the process of propose a DWXML representation, upon the metadata collected;
- Enable metadata editing supported by graphical interfaces;
- Generate the DWXML from the metadata collected/edited and embed it into the generated SIARD format;
- Enable primary data browsing using the preservation format proposed.

As well as in the DBPreserve project, this work uses a real world case study to test the migration process, in order to validate all the features implemented in the DBPreserve Suite application.

6.2 Case Study: SiFEUP Academic Surveys

This research uses the DBPreserve Project's case study. This case study started as a relational database that was used in an academic context to evaluate the satisfaction of the students relatively to the classes and the professors.

The study of the relational model migration to the dimensional model is not part of this work. This model migration process is being developed and refined in a parallel work [RDR10]. According to the scope of this project, the case study assumes already as a data warehouse, a dimensional model. This data warehouse was built on Oracle Database 11g Enterprise Edition Release 11.1.0.7.0 - 64bit Production[1].

The data warehouse has 10 fact tables (one with over 2 million records) and 7 dimension tables. The estimated size of the data warehouse is 115 MB, with a total of 2,598,428 records. This data warehouse is hosted in a server at FEUP. Table 6.1 illustrates the data warehouse table's statistics.

[1] http://www.oracle.com/us/products/database/enterprise-edition-066483.html

Table 6.1: Case study tables statistics

Table	Role	Rows	Size (KB)
IPDW_AGREG_COURSE_QUESTION	Fact table	4,320	384
IPDW_SEMESTER	Dim. table	12	40
IPDW_AGREG_FACULTY_GLOBAL	Fact table	55	32
IPDW_EXTENSION	Dim. table	33	40
IPDW_PROFESSOR	Dim. table	557	40
IPDW_AGREG_EXTENSION	Fact table	34,833	1,288
IPDW_QUIZ	Dim. table	17,096	464
IPDW_AGREG_FACULTY_QUESTION	Fact table	633	88
IPDW_AGREG_COURSE_GLOBAL	Fact table	357	56
IPDW_AGREG_FACULTY_EXTENSION	Fact table	177	40
IPDW_AGREG_QUESTION	Fact table	158,608	9,216
IPDW_COURSE	Dim. table	25	64
IPDW_ANSWERS	Fact table	2,365,189	105,256
IPDW_QUESTION	Dim. table	198	40
IPDW_CLASS	Dim. table	1,252	224
IPDW_AGREG_COURSE_EXTENSION	Fact table	1,036	104
IPDW_AGREG_GLOBAL	Fact table	14,047	552

6.3 Data Warehouse Metadata Enrichment

Each DBMS uses different ways to store database metadata. Since the DBPreserve project uses a model migration from the relational database to preserve the initial database, the dimensional model allows the abstraction of the initial database's DBMS. However, the use of Oracle Database to create the data warehouse during this initial migration, introduces a dependence on this technology but limits the scope within the existing DBMSs. Thus, using Oracle Database to create the dimensional model, there was a concern to study how the metadata is stored and what is the potential of this technology for the description of dimensional models.

The metadata relevant to the objects of an Oracle database are stored in a repository called *Data Dictionary*, which is a set of tables that provides information about the database. The structure of the data dictionary is composed by *Base Tables* that stores information about the associated database, *User-Accessible Views* that summarizes and displays the information stored in the base tables of the data dictionary and Oracle user *SYS* that owns all base tables and user-accessible views of the data dictionary [Ora02]. In many cases user-accessible views consists of three views containing similar information and distinguished from each other by their prefixes: `USER` (what is in the user's schema),

ALL (what the user can access) and DBA, the Database administrator's view (what is in all users' schemas). The metadata can be retrieved using SQL Data Definition Language (DDL) or through the DBMS_METADATA package that provides interfaces for extracting complete definitions of database objects.

To increase the volume of metadata for migration process is necessary to define primary and foreign keys. Those will be useful to build the star or snowflake schemas and inferring on the table role in the dimensional model (fact table, dimension table, bridge table...). Another complement for enriching the metadata is to add a description to each table, view and column, using the comments tables of the data dictionary (USER_TAB_COMMENTS for table and views and USER_COL_COMMENTS for columns).

Regarding dimensional models, Oracle Database has already the notion of the object *dimension*, allowing its definition through the CREATE DIMENSION clause [Ora03]. A key step in the process of the data warehouse creation is to declare the dimensions, so that the data dictionary contains this metadata and enables its future extraction. It eases the process of identifying the dimensions, levels and hierarchies, as well as tables and views that support them. An example of a dimension declaration is posted at Section 2.2.1.

6.4 DBPreserve Suite Architecture

The DBPreserve Suite application is a Windows desktop application that has a modular and extensible architecture, composed by 5 major modules as shown in the overall architecture of the application in figure 6.1. It has been developed using the NetBeans IDE 7.0 RC1 and Netbeans Platform[2], with support for Java 1.7[3], using the JDOM[4] library [Hun02] for XML processing of metadata. This application integrates a tool from the SIARD Suite that manages the migration of a relational database to the SIARD format, the SIARDfromDB application.

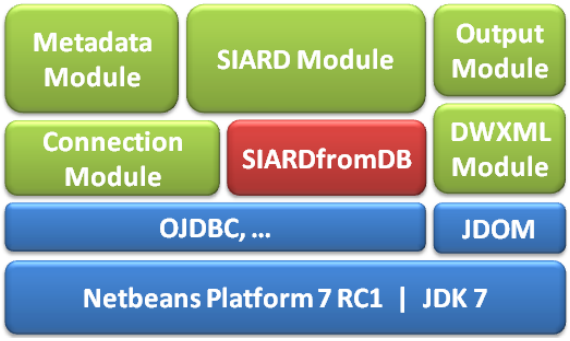

Figure 6.1: DBPreserve Suite general architecture

[2] http://netbeans.org/features/platform/
[3] http://download.java.net/jdk7/docs/api/
[4] http://www.jdom.org/index.html

The *Connection Module* enables the abstraction of the data warehouse connection, using Java Database Connectivity (JDBC) [Ora]. The application already supports connections to Oracle database using Oracle JDBC (OJDBC), due to the DBMS used in the proposed case study. However, this module is prepared for an easy extension to connect with other DBMS, adding just a file that rewrites some metadata retrieval methods.

The *Metadata Module* handles with all the metadata imported from the DBMS and with the metadata import process itself. The metadata imported is related with schemas, tables, views, columns, primary keys, foreign keys, dimensions, levels, level keys, hierarchies, attributes, table comments and column comments. Through the analysis of the acquired metadata, this module proposes a possible DWXML that describes the dimensional model.

The *SIARD Module* allows the integration of the `SIARDfromDB` tool from SIARD Suite [Tho09], that creates the SIARD format of the data warehouse. This format still lacks the dimensional model description. It looks to the data warehouse just at a relational model level. This module also manages the generated SIARD format, accessing to the relational metadata, enabling the primary data browsing and embedding the DWXML with the dimensional model description.

The *DWXML Module* handles with the dimensional metadata, creating the DWXML file to embed it into the SIARD format or reading it from the SIARD format if already added.

The *Output Module* manages all the graphical interfaces, such as connection management, SIARD format generation through `SIARDfromDB` integration, table and view roles visual editing, graphical representation of star or snowflake schemas and dimensions, hierarchical viewing of schemas, star and dimension, editing of the DWXML through several graphical user interfaces, viewing of the DWXML file added to the SIARD format and browsing of the primary data when selecting a star schema or dimension.

6.5 Development Highlights

The application deployment introduced several challenges as the application was taking shape, which were being overcome with relative ease through comprehensive analysis of the technologies used. These challenges ranged from the integration of external tools, through the metadata acquisition from the DBMS, the graphical editing of metadata, the inference on the acquired metadata for generating a proposal for the DWXML, the graphical display of schemas and dimensions, the use of the ZIP64 format that encapsulates the SIARD format, ending on the primary data browsing. The graphical user interfaces are presented in appendix B.

6.5.1 SIARD Suite tool integration

The SIARD Suite[5] component that extracts the database and builds the SIARD format is a command line application, the `SIARDfromDB`. This application is integrated into the DBPreserve Suite via a thread responsible for the process that manages the execution of the command `SiardFromDb` [Tho09], as well as the log of the migration execution progress. At this stage, the object to migrate is the relational database implementation of a data warehouse. The DBPreserve Suite application controls the `SIARDfromDB` process and catches the log to display it in the graphical interface.

In order to use DBPreserve Suite application, the SIARD Suite must be installed in the computer. DBPreserve Suite application just needs to know the path where the application is stored. The DBPreserve Suite application has an interface that manages this pathname. After that, the use of `SIARDfromDB` is completely transparent to the user. Using only the DBPreserve Suite it is possible to execute the whole migration of data warehouse to the SIARD format and to extend it with DWXML for description of the dimensional model.

6.5.2 Metadata Acquire Process

The process of extracting the metadata that describes the data warehouse is undertaken through the dictionary data analysis. This process is conducted by OJDBC driver, once the data warehouse from the project's case study was implemented using Oracle Database 11g. Thus, it is possible to directly extract some metadata from the database. This process allows to extract the definition of schemas, tables, primary keys and foreign, as well as the definition of views. However, to extract the definition of the dimensions, levels, hierarchies and comments of tables, views and columns, it is needed to query directly the respective tables in the data dictionary because the OJDBC have not such features.

This process can automatically assign a role to each table or view found, identifying them as fact tables, dimension tables, bridge tables or views. The *Fact table* role is set to those tables which have not a defined primary key and has at least one foreign key which refers to a dimension table or to a bridge table. The *Dimension table* role is defined to those tables which have a defined primary key and are referred within a level of a dimension declaration. The *Bridge table* role is assigned to the tables which have a reference to a fact table and to a dimension table. The *View* role is assign directly from the metadata that settles if the object is a table or a view. However, views can be used to implement dimensions. For those tables which do not fit any of these roles, they are define as *Unsorted tables*.

This role assignment is displayed to the user as a tree view of the existing roles, hanging the tables in the correspondent role node. If the auto assignment process returns some

[5]This application was gently sent by Johannes Bader from SIARD project

incorrect object role, it can be manually changed just by dragging and dropping the table in the correct role node.

6.5.3 DWXML Proposal

Through the analysis of the acquired metadata, a significant part of the metadata needed to build the DWXML file is automatically filled in, directly or by inference. Nevertheless, it is still necessary some manual input of small metadata details, such as objects' descriptions and facts' definitions. The schemas, tables, views, columns, primary keys, foreign keys, dimensions, levels, hierarchies and attribute levels are fully defined if the data dictionary has these metadata. The proposal builds a star for each fact table it founds. For every dimension used in the star, a ray element is added to the star which can contain also a bridge table reference, besides the dimension reference. However, it is necessary to add the facts' characterization and to define the fact table type (cumulative or snapshot).

The processing of metadata XML files is managed by the JDOM library, a straightforward, fast and lightweight API, optimized for Java programming, performing Java class representations of an XML document.

6.5.4 Editing the metadata

The metadata acquisition process contributes to a better understand of data warehouse, speeding up the process of creating the DWXML, maximizing the amount of metadata produced automatically. However, this process does not do the entire job in filling the DWXML with all the required metadata.

The metadata can be edited using the several graphical user interfaces built for an easy and visual editing of the DWXML. This feature has a major impact if the metadata from the data dictionary is poorly filled (e.g., no dimension declarations, no comments, no primary keys, no foreign keys). In these cases, the user must insert the metadata using these interfaces. Defining facts and datamarts is only possible using this feature, as there are no related metadata imported in the metadata acquire process.

6.5.5 Star schemas and dimensions graphical representation

Through graphical interfaces for editing the DWXML or by viewing directly the XML file created, a user can have a general vision of a star or snowflake schema, or even an overview of a dimension. However, this process can become rather complex, especially if stars schemas has many rays (references for dimensions) or even several dimensions with various levels and hierarchies.

Thus, the DBPreserveSuite application has a graphical representation of a star or snowflake schema as well as a graphical representation of a dimension. This enables

the user that is editing the metadata or browsing the primary data to have a clear view of the dimensional schema archived in XML-based format. The graphical representation was built using the NetBeans Visual Library 2.0 API[6]. Figures B.9 and B.10 show these graphical representations.

6.5.6 Preservation Format Packaging

For the SIARD format encapsulation, the SIARD Suite used at that time (2009) a proprietary application (PKZIP[7]) to create the uncompressed ZIP64, which extends the ZIP format to overcome the 4 GB size limit in the standard ZIP. However, DBPreserve Suite application handles the access and the integration of DWXML into the SIARD format is performed using the JDK7 (yet in Developer Preview Release version), that improves the `java.util.zip` library with support for ZIP64 format extensions [She09] defined by the PKWARE ZIP File Format Specification [PKW07]. Thus, access to the SIARD format and the integration of the DWXML file and its XSD schema is done as if it were a standard ZIP file.

6.5.7 Browsing the data

A major objective of preserving a digital object is to ensure the access to it on the long term. This work achieves this objective through the migration of data warehouse to XML-based files, as well as the description of the relational and the dimensional model.

The primary data acquisition was initially tested using the same library used to process the metadata XML files, the JDOM. Since JDOM loads the entire document into memory, loading and processing the primary data files was compromised for those with large sizes, restricted by the memory of the computer where the DBPreserve Suite application was being executed. An advantage in using JDOM is the ability to use XPath expressions to filter the data to process, supported by this library.

In fact, the implementation of the previous approach in the case study used in the project, verified that the processing of larger files was not possible, given the limited memory of the laptop that I often work. Thus, given the impossibility to process large XML files using JDOM and to use a similar processing method for the primary data files, we used the JDK XMLStreamReader interface. The XMLStreamReader allows a read-only access to XML and is designed to be the most efficient method to read XML data [Ora11]. The browsing of the data is restricted to a group of records at a time per page, to improve processing efficiency. This number of rows parameter can be configured by the user.

[6] http://platform.netbeans.org/graph/documentation.html
[7] http://www.pkware.com/software/pkzip

6.6 Technologies Involved

The support for digital repositories is largely associated with the use of XML standards, which guarantee the preservation and long-term access. This work calls the use of XML in many domains: XML is used in the drafting of information packages for descriptive, administrative, structural and technical metadata, as well as the use of SIARD which is based on XML files to preserve the primary data and metadata from a relational database, and DWXML that supports the description of a data warehouse. XML derived from an older standard format called SGML (ISO 8879) [SMB94]. XML is a W3C recommendation, based on a simple text-based format for representing structured information (documents, data, configurations, ...) [Con08].

The DBPreserve Suite application was implemented using Java, a programming language and computing platform first released by Sun Microsystems in 1995, which is used throughout the world, in computers, mobile devices, game consoles, etc. Java is everywhere! The developed application uses a specific Java version, the JDK7 Developer Preview Release [Ora11]. The choice of this not final release has mainly due to the ZIP64 support for files not available in earlier versions. Remember that the SIARD format is encapsulated by an uncompressed ZIP64 file.

For XML metadata processing, the JDOM library was used. The Java Document Object Model (JDOM) is a Java representation of an XML document, providing an easy and efficient reading, manipulation, and writing. It's an alternative to DOM [Con04] and SAX [Bro02], although the integration with both is possible. JDOM supports XPath expressions to filter the XML files, enabling the querying [Hun02, HM09]. Due to some hardware limitations for processing large XML files, the handling of the primary data XML files is done by the JDK XMLStreamReader interface [Ora11].

Finally, the DBPreserve Suite application was developed using the NetBeans IDE 7.0 RC1 and Netbeans Platform [Net11]. The NetBeans IDE is an open-source integrated development environment that enables developers to rapidly create web, enterprise, desktop, and mobile applications using the Java platform, among others. The NetBeans Platform architecture is modular, easing the creation of robust and extensible applications, containing several APIs that simplifies the handling of windows, actions, files, and many other things typical in applications [Net11].

6.7 Case Study: The Results

The main basis for the DBPreserve Suite application is to enable the migration to the proposed format, the extended SIARD format, and allow access to primary data archives using this format. The case study presented at Section 6.2 was been used since the beginning of the implementation of the application to verify this main objective. Before

start this migration, it is important to fill in the relevant metadata, both at relational and dimensional level, in order to allow an automated building of the XML file dimensional layer, the DWXML.

The process of creating the initial SIARD format, which consists of the relational metadata and the primary data, was done within the facilities of FEUP, to minimize delays in network connections. To generate this SIARD format, the creation of the `metadata.xml` file took about 4 minutes, while it took around 2:30h to migrate the primary data to this format. The structure of the final SIARD format is illustrated at Table 6.2 (XSDs not included for simplification).

Table 6.2: Case study extended SIARD format structure

Filename	**Size (KB)**	**Description**
`/header`		Metadata folder
`/metadata.xml`	71	SIARD relational model metadata
`/dw.xml`	86	Dimensional model metadata
`/content`		Content folder
`/schema0`		`CALDEIAS` schema folder
`/table0/table0.xml`	1,094	`IPDW_AGREG_COURSE_QUESTION` data
`/table1/table1.xml`	3	`IPDW_SEMESTER` data
`/table2/table2.xml`	15	`IPDW_AGREG_FACULTY_GLOBAL` data
`/table3/table3.xml`	7	`IPDW_EXTENSION` data
`/table4/table4.xml`	53	`IPDW_PROFESSOR` data
`/table5/table5.xml`	3,455	`IPDW_AGREG_EXTENSION` data
`/table6/table6.xml`	1,067	`IPDW_QUIZ` data
`/table7/table7.xml`	168	`IPDW_AGREG_FACULTY_QUESTION` data
`/table8/table8.xml`	88	`IPDW_AGREG_COURSE_GLOBAL` data
`/table9/table9.xml`	45	`IPDW_AGREG_FACULTY_EXTENSION` data
`/table10/table10.xml`	31,429	`IPDW_AGREG_QUESTION` data
`/table11/table11.xml`	4	`IPDW_COURSE` data
`/table13/table12.xml`	331,376	`IPDW_ANSWERS` data
`/table13/table13.xml`	63	`IPDW_QUESTION` data
`/table14/table14.xml`	264	`IPDW_CLASS` data
`/table15/table15.xml`	251	`IPDW_AGREG_COURSE_EXTENSION` data
`/table16/table16.xml`	1,383	`IPDW_AGREG_GLOBAL` data

The extraction of metadata from the dimensional model and the production of DWXML based only on the imported metadata data dictionary, took only 3 seconds. Graphical representations of stars schemas and the dimensions in Figures B.9 and B.10, rely only on the metadata collected in this process. However, the user can add or edit the associated

metadata. Combining the collected metadata to the segmentation of folders that characterizes the SIARD format, it is possible to access and display the primary data stored in XML with a more familiar presentation for tables (with rows and columns), as shown in figure B.11.

6.8 Future Features

The produced application already allows the access and navigation between the primary data stored in the extended SIARD format. A feature to add to the application would be to provide mechanisms to perform specific queries, more elaborate, using the potential of XPath and XQuery applied to primary data. The use of XPath supported by JDOM was tested in the case study, particularly in the metadata files processing. Its application to the primary data files, potentially extensive files, would need an analysis about the existing methods in processing large XML files. However, this feature would allow a better primary data navigation, browsing between hierarchy levels of the dimensions filtering the facts, and would enable potential *Ad hoc* queries using XQuery, similar to SQL for relational databases.

Another new feature would be the creation of a module that does the opposite route, i.e. based on the dimensional model archived using the extended SIARD format, the module would create a data warehouse using relational database technologies (DBMS), would add the relational and dimensional metadata to the data dictionary, and would load the primary data into the tables, reactivating the data warehouse.

The DBPreserve Suite application depends on a SIARD Suite tool for the generation of the initial SIARD format file. The process of primary data migration is very time consuming. Java platform is getting faster and faster[8]. To untie the DBPreserve Suite application from external applications and to improve the performance with recent Java versions, a new module could be developed to handle with the migration of the primary data and relational metadata.

The DBPreserve Suite application was tested just in one real world case study. A new one related to *SiFEUP Human Resources* is being prepared, with the migration from the initial relational model for dimensional model, the first step in the preservation approach of the DBPreserve project. Testing the application in a new case study will contribute for the refinement of the tool and bugs cleaning.

6.9 Summary and Conclusions

This chapter reports the development of the application that controls the entire migration process of a data warehouse to the proposed preservation format. The requirements for the

[8] Source from http://inebium.com/post/java-7-new-release-performance-code

application, called DBPreserve Suite, are presented as well as the case study that supports the implementation and testing of the tool. Related to an academic context to evaluate the satisfaction of the students relatively to the classes and the professors, the case study is a data warehouse built on Oracle Database 11g.

An important task to prepare the migration process is to verify that all metadata are present in the data warehouse. Thus, the volume of the acquired metadata will allow a more assisted production of the dimensional model XML representation, the DWXML.

The DBPreserve Suite has a modular architecture, composed by five major modules: the *Connection Module* allows the abstraction of the data warehouse connection; the *Metadata Module* handles with all the metadata imported from the DBMS and its analysis; the *SIARD Module* enables the integration of the `SIARDfromDB` tool from SIARD Suite and manages the generated SIARD format, the access to the relational metadata, and primary data browsing; the *DWXML Module* creates and handles the DWXML dimensional metadata file; and the *Output Module* manages all the graphical interfaces of the application. The application was developed using NetBeans IDE 7.0 RC1 and Netbeans Platform, with support for Java 1.7, using the JDOM library for XML processing of metadata. Using NetBeans Platform boosted the development graphical interfaces, with high level, and provided that all the objectives proposed for this application were met.

The interpretation of the data stored in XML format fulfils one of the essential requirements of a digital preservation process, the access to the data. However, the application can be extended to achieve other useful features, improving the results so far. In addition to the development of this application, the study of techniques for large XML files processing could lead to implement efficient filtering techniques for queries, with XPath or XQuery. The reactivation of an archived data warehouse from the extended SIARD format to a DBMS, and the implementation of the module to generate the SIARD format, are other features that could be develop in the future.

The test of the developed application in the case study corresponded to the expectations. In fact, the migration and storage of the dimensional model is guaranteed by its representation through DWXML, allowing the identification and characterization of star schemas, snowflake schemas, datamarts, facts, dimensions, hierarchies, attributes and levels. DWXML also supports the graphical representation of star or snowflake schemas and dimensions, as well as the browsing of the primary data. This application does not alter the primary data content folder. The final format is still compatible with SIARD Suite programs.

Chapter 7

Conclusions and Future Work

The research around relational databases preservation suggests several strategies to ensure the access to primary data in the long term. Databases are different from conventional digital objects (e.g., documents, images), as they have an internal structure, include schemas and integrity constrains, which are vital for data interpretation. The concern in the preservation of complex digital objects, such as relational databases or data warehouses implemented using relational database technologies (ROLAP), lies not just in the preservation of primary data (the data that is stored in tables), but also focuses on the preservation of metadata in the database, i.e. the technical and structural description of the database.

However, there is a clear trend to use standardized and technological independent formats, based on text files. The relational databases are supported by DBMS where most of them already enable the export of primary data into text formats, mainly for operability purposes. Given the diversity of existing DBMS, each one has its own way to create and represent these text files, so there is not a uniform representation of data.

Standardized representations of relational databases are essential in order to ensure the digital preservation in the long-term. These formats are based on XML (e.g. DBML and SIARD), being platform and application independent, simple text format and human readable, currently widely used in various application domains, assumed as an effective method for long-term preservation of relational databases. A good documentation on preservation formats ensures the future access and interpretation of data, either by direct consultation of the created XML files, or preferably, by developing an application that performs the parsing of these files and shows the data in a more user friendly appearance.

The research community has been engaging to define a common terminology and a reference functional model for an archive system of digital objects, which are described by the Open Archival Information System (OAIS) Reference Model. Though the implementation of a digital repository for relational databases is not the scope of this work,

an analysis was made regarding the metadata needs of an OAIS for databases, in order to build a draft representation of the Submission Information Package (SIP), which the Producer delivers to the OAIS, also of the Archival Information Package (AIP), the package to preserve, and the draft of the Dissemination Information Package (DIP), what a Consumer receives when access and requires a resource from the OAIS.

The DBPreserve project where my work is inserted, approaches the relational database preservation issue with an approach of migrating the relational model of the initial database to a dimensional model (not the focus of this report), using a data warehouse implemented by relational databases technologies, and then achieve a preservation format upon that data warehouse (one of the objectives of this work).

Data warehouses are used in many application domains, and there is the established method for their preservation. This work studies the existing XML-based formats for relational database preservation (DBML and SIARD) and identifies which one will be better to a data warehouse implemented using relational database technologies. However, none of these formats allows the characterization of the dimensional model concepts (star schema, snowflake schema, fact, dimension, hierarchies, levels, level keys, data marts). The inherent modularity of data warehouses, with independent stars sharing some dimensions, lead to choose the format that also supports a segmented structure of the primary data, the SIARD format. The SIARD format is an uncompressed ZIP64 package based on an organizational system of folders, storing the metadata and the primary data files into separated folders. The SIARD project also developed a set of tools, the SIARD Suite. One of them, the `SIARDfromDB` is a command line application that builds the SIARD format from a relational database.

In order to represent a data warehouse in a technological independent and long-term preserving format, my approach proposes to extend the SIARD format, adding a metadata layer for data interpretation according to the data warehouse perspective. This extra metadata layer must accommodate the dimensional model concepts not covered by the standard SIARD format. To achieve the dimensional model XML-based representation, it is presented the DWXML. This XML dialect was defined trying to cover all the concepts and all the specific cases of the dimensional model. As documentation of preservation formats is vital to ensure the success of the preservation archive, this report contains a full description of the DWXML, as well as how to extend the SIARD format. The DWXML was also presented at XATA2011 conference and published in the workshop proceedings as a paper named “DWXML - A Preservation Format for Data Warehouses” (see appendix A).

To support the entire migration process of the data warehouse and associated metadata to the SIARD format and extend it with the DWXML dimensional model metadata layer, an application was implemented, the DBPreserve Suite application. Supported by NetBeans IDE 7.0 RC1 and Netbeans Platform and JDK7, this application has five modules:

the *Connection Module* enables the abstraction of the data warehouse connection; the *Metadata Module* handles with all the metadata imported from the DBMS and its analysis; the *SIARD Module* allows the integration of the `SIARDfromDB` tool from SIARD Suite and manages the generated SIARD format, the access to the relational metadata, and primary data browsing; the *DWXML Module* creates and handles the DWXML dimensional metadata file; and the *Output Module* manages all the graphical interfaces of the application (see appendix B).

The proposed preservation format and the tool deployed for the migration process were tested in a real world case study, a data warehouse implemented using Oracle Database 11g, containing data from an academic context in order to evaluate the satisfaction of the students relatively to the classes and the professors. To major the metadata acquisition from data warehouse, it is essential to enrich the data dictionary with the maximum possible description of the objects (tables, keys, dimensions, levels, hierarchies...). The DBPreserve Suite application will have a better understand of data warehouse, speeding up the process of creating the DWXML, maximizing the amount of metadata produced automatically. The remaining metadata are added through graphical user interfaces created for this purpose.

7.1 Satisfaction of Objectives

The objectives proposed for this project have been met, however with inherent challenges that were overcome during the study, development and verification of the project objectives, being here highlighted the key aspects to achieving them.

The knowledge of the OAIS associated terminology and the verification of an OAIS needs has allowed a better understanding of these archive systems, and had assisted in the identification of different types of metadata for the OAIS resources characterization, and contributed to the sketch of the information packages: SIP, AIP and DIP. The implementation of an OAIS should be supported by metadata schemas which provide descriptive, administrative, structural and technical metadata. The descriptive metadata is achieved using the EAD standard. PREMIS standard is pointed for administrative metadata and technical metadata, while METS is referenced for structural metadata. PREMIS has a restricted technical metadata and there are no specific technical metadata standard for relational databases or data warehouses implemented using relational database technologies. But, the metadata acquired from the data dictionary can be used as a complement for technical metadata. However, the complete acceptance of these proposed packages would have to be verified with the development of an OAIS repository, where perhaps the choices in the packages representation will require a refinement process.

Regarding the proposed format for data warehouse preservation, the decision of combining the DWXML (that describes the dimensional model), with the SIARD format (for

relational model description and primary data storage), proved to be a useful way to represent a data warehouse in XML-based files. In fact, the DBPreserve Suite application uses this extended SIARD format for star and snowflake schemas representation, as well as dimension structures (hierarchies, levels and attributes), and enables the browsing of primary data from dimensional model perspective (through stars and dimensions). Thus, extending the SIARD format with a DWXML dimensional model metadata layer, a long-term preservation format for data warehouses is achieved.

Looking to the major implemented features of DBPreserve Suite application, the integration of the `SIARDfromDB` command line application from the SIARD Suite enables the standard SIARD format generation, migrating all relational metadata and primary data according to that format, with a total control of this process from the developed application. So, reusing that effort, the essence of the application focused on the description of the dimensional model, by importing the metadata from the data dictionary, automating the creation of DWXML after analysis of the imported metadata, providing user interfaces for a friendly DWXML editing, embedding it into the SIARD format and enabling the access to primary data through the dimensional model perspective. All the implemented features were tested and refined using the case study of project.

7.2 Future Work

The import process of the dimensional model description of a data warehouse, together with their primary data, using the proposed preservation format, the extension of the SIARD format with the DWXML, is supported by the tool developed, the DBPreserve Suite application. Although this application had been tested in a real case study during its development, it is appropriate to apply other scenarios to check its behavior. A new case study is being prepared to use with the application, and so cleaning some edges, where justified.

To improve the generated application, other new features can be introduced, such as the implementation of the reverse migration process, i.e. starting from the final preservation format in XML, reactivate the data warehouse through its reconstruction in a DBMS and then loading with the primary data. Another new feature could be the implementation of a module that generates the initial SIARD format, untying the DBPreserve Suite application from the SIARD Suite tool, making it the completely autonomous. One of the most important improvements is to provide to the application methods to query the primary data XML files. Therefore, it is necessary to analyze the efficiency of techniques for large XML documents processing, as well study which user query language will be better applied.

The preservation format proposed is not itself a guarantee of success regarding long-term digital preservation. Being a preservation format for data warehouses implemented

with relational databases technologies, it fulfils only one requirement among many others to ensure the success of an OAIS. An OAIS is a platform to support digital preservation, with several stages and with well-defined functional entities. All functional entities (Ingest, Archival Storage, Access, Preservation Planning and Administration) are fundamental to the success of the archive system. This work points to possible representations of the information packages involved in the various stages of preservation (SIP, AIP and DIP), which should be checked, improved and accepted upon the implementation of the OAIS. The implementation of an OAIS is a wide and complex process, but no doubt crucial to ensure the long term preservation of the archived resources. This document reports a milestone in the process of implementing an OAIS, which has numerous iterations.

References

[Ack89] Russell L. Ackoff. From Data to Wisdom. *Journal of Applied Systems Analysis*, 16:3–9, 1989.

[ADR11] Carlos Aldeias, Gabriel David, and Cristina Ribeiro. DWXML - A Preservation Format for Data Warehouses. In *XATA 2011 - XML: Aplicações e Tecnologias Associadas*, pages 115–126, June 2011.

[Ass07] Blu-Ray Disc Association. Blu-ray Disc, March 2007.

[BCF+07] F. Barbedo, L. Corujo, L. Faria, R. Castro, M. Ferreira, and J. C. Ramalho. RODA: Repositório de Objectos Digitais Autênticos. In *9º Congresso Nacional de Bibliotecários, Arquivistas e Documentalistas, Ponta Delgada, Portugal*, 2007.

[Bhu00] Bharat Bhushan. *Mechanics and Reliability of Flexible Magnetic Media*. Springer, 2000.

[BKM07] Stefan Brandl and Peter Keller-Marxer. Long-term Archiving of Relational Databases with Chronos. In *First International Workshop on Database Preservation - PresDB'07*, 23 March 2007.

[Bro02] David Brownell. *SAX 2*. O'Reilly Media, January 2002.

[CB07] Vassilis Christophides and Peter Buneman. Report on the First International Workshop on Database Preservation, PresDB'07. *SIGMOD Record*, Vol. 36, No. 3:55–58, September 2007.

[CCMW01] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web Services Description Language (WSDL) 1.1, March 2001.

[CD97] Surajit Chaudhuri and Umeshwar Dayal. An Overview of Data Warehousing and OLAP Technology. *SIGMOD Rec.*, 26:65–74, March 1997.

[CGOZ99] P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni. *Flash Memories*. Springer, 1999.

[Com98] Federal Geographic Data Committee. Content Standard for Digital Geospatial Metadata, 1998.

[Com07] VRA Core Oversight Committee. VRA Core 4.0, September 2007.

[Com11] PREMIS Editorial Committee. Data Dictionary for Preservation Metadata: PREMIS version 2.1, January 2011.

[Con97] Internet Mail Consortium. vCard: The Electronic Business Card, Version 2.1, January 1997.

[Con04] World Wide Web Consortium. Document Object Model (DOM), April 2004.

[Con08] World Wide Web Consortium. Extensible Markup Language (XML) 1.0 (Fifth Edition) W3C Recommendation, November 2008.

[Cor10] Dublin Core. The Dublin Core Metadata Element Set, October 2010.

[CTI10] Giovanni Campardo, F. Tiziani, and Massimo Iaculo. *Memory Mass Storage*. Springer, 2010.

[Dat04] C. J. Date. *An Introduction to Database Systems (Eight Edition)*. Pearson, Addison Wesley, 2004.

[Day98] Michael Day. Issues and Approaches to Preservation Metadata. In *Conf. Guidelines for Digital Imaging, University of Warwick*, 1998.

[Des90] Bipin C. Desai. *An Introduction to Database Systems*. West Publishing Company, 1990.

[Dev04] Hasan A. Deveci. Databases: Is Sui Generis a Stronger Bet Than Copyright? *International Journal of Law and Information Technology*, 12 No. 2:178–208, 2004.

[DF09] Angela Dappert and Adam Farquhar. Implementing Metadata that Guides Digital Preservation Services. In *iPress2009*, San Francisco, California, 5-6 October 2009.

[EN00] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems (Third Edition)*. Addison-Wesley, 2000.

[ESC93] Codd E.F., Codd S.B., , and Salley C.T. Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate. *Codd & Date, Inc*, 1993.

[Fed10] Digital Library Federation. *<METS> Metadata Encoding and Transmission Standard: Primer and Reference Manual*, version 1.6 revised edition, 2010.

[Fer06] Miguel Ferreira. *Introdução à Preservação Digital - Conceitos, estratégias e actuais consensos*. Escola de Engenharia da Universidade do Minho, 2006.

[FHY07] Adam Farquhar and Helen Hockx-Yu. PLANETS: Integrated Services for Digital Preservation. *International Journal of Digital Curation, Issue 2*, Volume 2:88–99, 2007.

[For02] Authenticity Task Force. Requirements for Assessing and Maintaining the Authenticity of Electronic Records. Technical report, InterPARES Project, Vancouver, Canada, 2002.

[fSDS02] Consultative Committee for Space Data Systems. *Reference Model for an Open Archival Information System (OAIS) - Blue Book*. Washington: National Aeronautics and Space Administration, 2002.

[GDO03] Encoded Archival Description Working Group, Network Development, and MARC Standards Office. Encoded Archival Description Tag Library, Version 2002. Technical report, Society of American Archivists and Library of Congress, 2003.

[Gra00] Stewart Granger. Emulation as a Digital Preservation Strategy. *D-Lib Magazine*, Volume 6 Number 10, October 2000.

[Gro02] EAD Schema Working Group. Encoded Archival Description (EAD), 2002.

[GW96] J. Garrett and D. Waters. Preserving Digital Information, Report of the Task Force on Archiving of Digital Information. Technical report, The Commission on Preservation and Access and The Research Libraries Group, Washington DC and Mountain View CA, 1996.

[Hac97] Douglas Hackney. *Understanding and Implementing Successful Data Marts*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA., 1997.

[HBH03] Wolfgang Hummer, Andreas Bauer, and Gunnar Harde. XCube: XML for Data Warehouses. In *Proceedings of the 6th ACM international workshop on Data warehousing and OLAP*, DOLAP '03, pages 33–40, New York, NY, USA, 2003. ACM.

[Hed97] Margaret Hedstrom. Digital Preservation: A Time Bomb for Digital Libraries. *Computers and the Humanities*, 31:189–202, 1997.

[Hen98] Tony Hendley. Comparison of Methods & Costs of Digital Preservation. Technical report, British Library Research and Innovation Centre, 1998.

[HL01] Margaret Hedstrom and Clifford Lampe. Emulation vs. Migration: Do Users Care? *RLG DigiNews*, 5 Num 6, 2001.

[HM09] Jason Hunter and Brett McLaughlin. JDOM, the Java Document Object Model, July 2009.

[Hun02] Jason Hunter. JDOM in the Real World - JDOM makes XML Manipulation in Java Easier than Ever. *Oracle Magazine*, September/October 2002.

[HW01] D. Holdsworth and P. Wheatley. Emulation, Preservation and Abstraction. *RLG DigiNews*, Volume 5, Number 4, August 2001.

[Ini09] Data Documentation Initiative. Data Documentation Initiative (DDI), October 2009.

[Inm92] W. H. Inmon. *Building the Data Warehouse*. John Wiley & Sons, New York, 1992.

[IP08] Swiss Federal Archives SFA Unit Innovation and Preservation. SIARD Format Description. Technical report, Federal Department of Home Affairs FDHA, Berne, 2008.

[JMP01] Mikael R. Jensen, Thomas H. M"ller, and Torben Bach Pedersen. Specifying OLAP Cubes on XML Data. In *Proceedings of the 13th International Conference on Scientific and Statistical Database Management*, SSDBM '01, pages 101–, Washington, DC, USA, 2001. IEEE Computer Society.

[KR02] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 2002.

[Lor01] Raymond A. Lorie. Long-Term Archiving of Digital Information. In *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, 2001.

[Lor02] Raymond A. Lorie. *The UVC: a Method for Preserving Digital Documents - Proof of Concept*. December 2002.

[Lor05] Raymond A. Lorie. *UVC: A Universal Virtual Computer for Long-Term Preservation of Digital Information*. IBM Research Division, February 2005.

[LPSW06] Carl Lagoze, Sandy Payette, Edwin Shin, and Chris Wilper. Fedora: An Architecture for Complex Objects and their Relationships. *International Journal on Digital Libraries*, Vol. 6 Num. 2:124–138, 2006.

[LSL+02] Kyong-Ho Lee, Oliver Slattery, Richang Lu, Xiao Tang, and Victor McCrary. The State of the Art and Practice in Digital Preservation. *Journal of Research of the National Institute of Standards and Technology*, Volume 107, Number 1:93–106, 2002.

[LvD05] Raymond A. Lorie and Raymond J. van Diessen. UVC: Long-Term Preservation of Complex Processes. *IS&T Archiving Conference*, Washington, DC:26–29, 2005.

[Mar96] EURIM European Informatics Market. Database Directive. EURIM Briefing No. 6, 1996.

[MMvPW06] E.R. Meinders, A.V. Mijiritskii, L. van Pieterson, and M. Wuttig. *Optical Data Storage - Phase-change media and recording*. Springer, 2006.

[MWS02] P. Mellor, P. Wheatley, and D. Sergeant. Migration on Request, a Practical Technique for Preservation. In *Research and Advanced Technology for Digital Libraries : 6th European. Lecture Notes in Computer Science,*, pages 516–526. Springer, Berlin / Heidelberg, September 2002.

[Net11] NetBeans. The NetBeans Platform, 2011.

[oC00] Library of Congress. MARC 21 Specifications for Record Structure, Character Sets, and Exchange Media, January 2000.

[oC10] Library of Congress. Metadata Object Description Schema (MODS), June 2010.

[oC11] The Library of Congress. Metadata Authority Description Schema (MADS) , Version 2.0, June 2011.

[Ora] Oracle. JDBC Overview.

[Ora02] Oracle. Oracle9i Database Concepts Release 2 (9.2) - The Data Dictionary, 2002.

[Ora03] Oracle. Oracle Database SQL Reference 10g Release 1 (10.1), Part Number B10759-01, 2003.

[Ora11] Oracle. Java Platform Standard Ed. 7, DRAFT ea-b141, 2011.

[PKW07] PKWARE. ZIP File Format Specification, Version: 6.3.2, Revised: September 28, 2007, 2007.

[PL98] Sandra Payette and Carl Lagoze. Flexible and Extensible Digital Object and Repository Architecture (FEDORA). *Second European Conference on Research and Advanced Technology for Digital Libraries. Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York*, Vol. 1513:41–59, 1998.

[PLA09] PLANETS. PLANETS: Tools and Services for Digital Preservation. PLANETS Product Sheet, 2009.

[Pre04] NISO Press. *Understanding Metadata*. NISO Press National Information Standards Organization, 2004.

[RDR10] Arif Ur Rahman, Gabriel David, and Cristina Ribeiro. Model Migration Approach for Database Preservation. In *The Role of Digital Libraries in a Time of Global Change, 12th International Conference on Asia-Pacific Digital Libraries, ICADL 2010, Gold Coast, Australia.*, pages 81–90. Springer Berlin / Heidelberg, 2010.

[RFC+07] J. C. Ramalho, M. Ferreira, R. Castro, L. Faria, F. Barbedo, and L. Corujo. XML e Preservação Digital. In *XATA - XML: Aplicações e Tecnologias Associadas 2007*, 2007.

[RFFC07] José Carlos Ramalho, Miguel Ferreira, Luís Faria, and Rui Castro. Relational Database Preservation through XML modelling. In *Extreme Markup Languages 2007*, 2007.

[Rot00a] Jeff Rothenberg. *An Experiment in Using Emulation to Preserve Digital Publications*. The Koninklijke Bibliotheek and RAND-Europe, 2000.

[Rot00b] Jeff Rothenberg. *Using Emulation to Preserve Digital Documents*. RAND-Europe, July 2000.

[Rus00] Kelly Russell. Digital Preservation and the CEDARS Project Experience. *New Review of Academic Librarianship*, Volume 6:139–154, 2000.

[She09] Xueming Shen. ZIP64, The Format for > 4G Zipfile, Is Now Supported. Xueming Shen's Oracle Blog, Apr 17 2009.

[Sin10] Pauline Sinclair. The Digital Divide: Assessing Organizations' Preparations for Digital Preservation. Planets White Paper, March 2010.

[Sla02] Jacqueline Slats. Practical Experiences of the Digital Preservation Testbed. In *DLM Forum, Barcelona*, 7 May 2002.

[SM98] Thom Shepard and Dave MacCarn. Universal Preservation Format - Background and Fundamentals. In *Sixth DELOS Workshop Preservation of Digital Information*, Tomar, Portugal, 1998.

[SMB94] C. M. Sperberg-McQueen and Lou Burnard. A Gentle Introduction to SGML, 1994.

[SWP03] Thornton Staples, Ross Wayland, and Sandra Payette. The FEDORA Project: An Open-source Digital Object Repository Management System. *D-Lib Magazine*, Volume 9 Number 4, April 2003.

[Tes03] Digital Preservation Testbed. From Digital Volatility to Digital Permanence: Preserving Databases. Technical report, Dutch National Archives and the Dutch Ministry of the Interior and Kingdom Relations, 2003.

[Thi02] Kenneth Thibodeau. Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years. In *The State of Digital Preservation: An International Perspective*. Documentation Abstracts, Inc. - Institutes for Information Science, 2002.

[Tho09] Hartwig Thomas. *SIARD Suite Manual*. Swiss Federal Archives, 19 May 2009.

[TJC06] Jim Taylor, Mark R. Johnson, and Charles G. Crawford. *DVD Demystified*. McGraw-Hill Companies, Inc., 3th edition edition, 2006.

[vdH07] Jeffrey van der Hoeven. Emulation for Digital Preservation in Practice: The Results. *The International Journal of Digital Curation*, Issue 2, Volume 2:123–132, 2007.

[vdHvW05] Jeffrey van der Hoeven and Hilde van Wijngaarden. Modular Emulation as a Long-term Preservation Strategy for Digital Objects, 2005.

[Ver03] Remco Verdegem. Databases Preservation Issues. In *Erpanet workshop on Long-term Preservation of Databases in Bern, Switzerland on 9 April 2003*. Digital Preservation Testbed, 2003.

[Whe01] Paul Wheatley. Migration: a CAMiLEON discussion paper. In *Ariadne, Issue 29*, Oct 2001.

[WT04] Frank L. Walker and George R. Thoma. A Web-Based Paradigm for File Migration. In *Proceedings of IS&T's 2004 Archiving Conference. Springfield, VA: IS&T: The Society for Imaging Science and Technology*, 2004.

[WTF03] Tim Weitzel, Thomas Tesch, and Peter Fankhause. A Scalable Approach to Processing Large XML Data Volumes. In *Americas Conference on Information Systems (AMCIS) 2003 Proceedings*, number Paper 315, 2003.

[WWHD00] Andrew Waugh, Ross Wilkinson, Brendan Hills, and Jon Dell'oro. Preserving Digital Information Forever. In *International Conference on Digital Libraries Proceedings of the fifth ACM conference on Digital libraries*, pages 175–184, 2000.

[ZvW08] Eld Zierau and Caroline van Wijk. The PLANETS Approach to Migration Tools. In *IS&T Archiving 2008*, Bern, Switzerland, 2008. Society for Imaging Science and Technology.

REFERENCES

Appendix A

DWXML - A Preservation Format for Data Warehouses

The DWXML schema detailed description reported at Section 5.4 is already a published work, presented at "XML: Aplicações e Tecnologias Associadas - XATA2011" workshop, on June 1 2011. The produced paper is included in the conference proceedings and is replicated in this appendix.

DWXML - A Preservation Format for Data Warehouses*

Carlos Aldeias, Gabriel David, and Cristina Ribeiro

Departamento de Engenharia Informática
Faculdade de Engenharia da Universidade do Porto
INESC Porto
Portugal
{carlos.aldeias,gtd,mcr}@fe.up.pt

Abstract. Data warehouses are used in many application domains, and there is no established method for their preservation. A data warehouse is structured by star or snowflake representations and can be grouped into data marts. A star is made up of a fact table that stores the facts, and dimensional tables that contextualizes the facts. There are also bridge tables used to resolve a many to many relationship between a fact table and a dimension table, or to flatten out a hierarchy in a dimension table. A snowflake is similar to a star but where the dimension tables have suffered a partial normalization, resulting in subdimensions. A data warehouse can be implemented in multidimensional structures or relational databases that represents the dimensional model concepts in the relational model. The focus of this work is on describing the dimensional model of a data warehouse and migrating it to an XML model, in order to achieve a long-term preservation format. This paper presents the definition of the XML structure that extends the SIARD format used for the description and archive of relational databases, enriching it with a layer of metadata for the data warehouse components. Data Warehouse Extensible Markup Language (DWXML) is the XML dialect proposed to describe the data warehouse. To acquire the relevant metadata for the warehouse and build the archive format, an application was produced that combines the SIARD format and the DWXML metadata layer.

Keywords: Database Preservation, DWXML, SIARD format

1 Introduction

The technological generation in which we live has gradually modified the method to create, process and store information, using compulsively digital means for this purpose. The institutions, enterprises and governments rely more and more

* This work is supported by FCT grant reference number PTDC/CCI/73166/2006.

on information systems that increase the availability and accessibility of information. These information systems typically require relational databases, transforming them into valuable assets for those entities.

However, rapid technological changes degenerate into rapid obsolescence of applications, file formats, media storage and even databases management systems (DBMS) [1]. If nothing is done, access to large chunks of stored information may become impossible and it be lost forever. So, it is important that entities which have major responsibilities in preserving information in digital form, become aware of this problem and join to initiatives all over the world, seeking for the best methodology for digital long-term preservation, and in particular for database preservation.

The present work is a development product of the DBPreserve[1] project, a research project funded by the portuguese Foundation for Science and Technology (FCT), in collaboration with INESC Porto, University of Minho and National Archives of Portugal (DGARQ), aiming at studying the feasibility of using data warehousing technologies to preserve complex electronic records, such as those constituting databases. DBPreserve project approaches the long-term preservation of relational databases issue with a new concept, a two step migration:

- A model migration from the relational model to the dimensional model, using data warehouse concepts for model simplification and efficiency increase [2];
- An XML migration from the dimensional model to an XML [3] format that represents the data warehouse, to ensure a long-term preservation format.

A data warehouse has star or snowflake representation, made up of fact tables and dimensional tables that adds context and meaning to the facts. When a dimension table is partially normalized, resulting in subdimensions, it is called a snowflake schema. A bridge table is used between a fact table and a dimension table or to flatten out a hierarchy in a dimension table. Data marts are subsets of a data warehouse.

Data Warehouse Extensible Markup Language (DWXML) is an XML dialect with the purpose of describing a Data Warehouse (DW) [1, 4, 5]. It has been defined and refined according to data warehouse's properties and tested using a case study of SiFEUP[2]. Its use in the project lies as a complement to the SIARD format [6] used for the description and archive of relational databases. This enrichment leverages past efforts to define an archive format suitable for data tables from databases and adds a layer of metadata for the data warehouse perspective.

2 Data Warehouse Preservation

Digital preservation has become more and more the focus for researching about what is the best strategy that is sustainable and efficient for the long-term preservation of digital objects [7]. Thibodeau's organization of digital preservation strategies relate them to their applicability and objective [8].

[1] `http://www.fe.up.pt/si/PROJECTOS_GERAL.MOSTRA_PROJECTO?P_ID=1349`
[2] Information System of Faculty of Engineering, University of Porto, Portugal

The Open Archival Information System (OAIS) Reference Model [9] introduces the appropriate terminology in the context of long-term preservation and defines the functional components necessary to implement an archive.

There are already many efforts and projects developed under the digital preservation scope. Projects such as CAMiLEON [10], InterPARES [11], FEDORA [12] or PLANETS [13, 14, 16] contributed to the study of requirements, strategies and proposals for preserving digital objects and ensure their authenticity.

Regarding complex digital objects, such as databases, projects like SIARD [6], Chronos [17] or RODA [18], analyzed in detail the preservation of relational databases. PLANETS project built a framework that also deals with Access, MS SQL Server and Oracle databases, as well as the SIARD format [19].

Data warehouses are often implemented using relational database technology, and thus they are made up of tables that store data. A deeper inspection leads to the finding of facts, dimensions, bridges tables, indexes, level keys and views. However, there are some key differences between a database used in an operational system and in a data warehouse.

W. H. Inmon defined a data warehouse as "a subject-oriented, integrated, nonvolatile, time variant collection of data in support of managements decisions" [4]. Data warehouses fulfill two major purposes: provide a single, clean and consistent source of data for decision support and unlink the decision platform from the operational system [1].

In a data warehouse the tables and joins are simple and de-normalized, in order to reduce the response time for analytical queries. For the characterization of a data warehouse additional metadata is required that defines the dimensional model and allows the data interpretation across different perspectives.

2.1 Data Warehouse Metadata

The structure of a data warehouse is referred to as a dimensional schema, where the fact tables are surrounded by dimensional tables, forming star schemas. A fact table is often located at the center of a star schema and consists of facts of a business process (e.g., measurements, metrics).

To understand the facts it is necessary to introduce the context and meaning of the dimensional model, achieved by the dimensions, representing the relevant vectors of analysis of the business process facts. The dimensions allow us to identify the how, what, who, when, where and why of something. Dimensions are usually represented by one or more dimensional tables. A dimensional table contains attributes in order to define and group the data for data warehouse querying.

The dimensions are characterized by a set of levels with defined hierarchies. Hierarchies are logical structures that use levels to organize and aggregate data, define navigation paths or establish a family structure [4, 5]. A common example is a time dimension, a hierarchy might aggregate data from the day level to the week level to the month level to the quarter level to the year level.

The figure 1 shows an example of a star schema related to a real world case study used in the project, a "Course Evaluation System", aiming to obtain general statistics about user satisfaction (anonymous students) in an academic environment scope, specifically on professor and class evaluation.

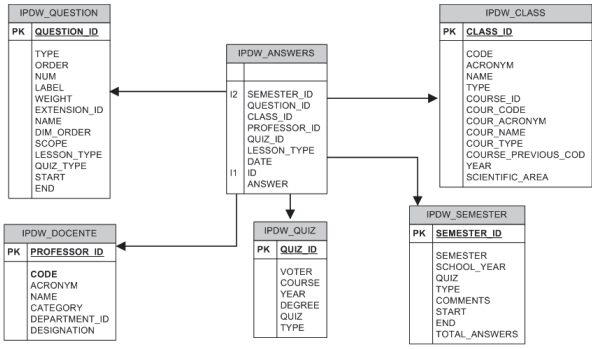

Fig. 1. Star schema example

In the center, a fact table contains the submitted answers (`IPDW_ANSWERS`). As dimensional tables, there are the question table (`IPDW_QUESTION`), the quiz table (`IPDW_QUIZ`), also the semester table (`IPDW_SEMESTER`), the class table (`IPDW_CLASS`) and the professor table (`IPDW_PROFESSOR`). Because the answers are anonymous, there is no relationship towards the students, who actually answered the questions. An important step in the data warehouse building process is to declare the dimensions. The next sample code shows the declaration of a dimension with the `CREATE DIMENSION` SQL statement [20] using Oracle.

Example of a dimension declaration

```
CREATE DIMENSION class_dim
  LEVEL class IS (IPDW_CLASS.CLASS_ID)
  LEVEL course IS (IPDW_CLASS.COURSE_ID)
  HIERARCHY class_rollup(
    class CHILD OF
    course)
  ATTRIBUTE class DETERMINES
    (IPDW_CLASS.CODE, IPDW_CLASS.ACRONYM,
     IPDW_CLASS.NAME, IPDW_CLASS.TYPE)
  ATTRIBUTE course DETERMINES
    (IPDW_CLASS.COUR_CODE, IPDW_CLASS.COUR_ACRONYM,
     IPDW_CLASS.COUR_NAME, IPDW_CLASS.COUR_TYPE,
     IPDW_CLASS.COURSE_PREVIOUS_COD);
```

This declaration defines a dimension (`class_dim`) with a hierarchy (`class_rollup` of two levels: the level `course` with `COURSE_ID` as level key, and a child level `class` with `CLASS_ID` as level key. This dimension uses the data from the table `IPDW_CLASS`. The `ATTRIBUTE` clause specifies the attributes that are uniquely determined by a hierarchy level. Thus it is possible to analyze the data in a more global perspective, through the `course` level, or get a more detailed overview using the `class` level.

Another data warehouse concept is a bridge table. A bridge table is used to resolve a many to many relationship between a fact table and a dimension table and is also used to flatten out a hierarchy in a dimension table [5].

Storing snowflake schemas and data marts is also needed. The snowflake schema is similar to the star schema, but dimensions are normalized into multiple related tables. A data mart is a subset of a data warehouse [5, 21].

2.2 Data Warehouse Preservation Format Proposal

The main objective of this study was to obtain a preservation format that suited the characteristics of a generic data warehouse. This format should allow the definition of the relevant metadata from the perspective of the data warehouse and archive the relevant metadata as well as the data from the tables in a format that would guarantee long-term preservation. The use of XML to the verification of these requirements appeared as the next option.

The study of the work already produced around the preservation of databases [6, 17, 18], including the model migration approach developed in the DBPreserve project [2], and on XML representation of a data warehouse [22, 23], resulted in the decision to complement the SIARD format, an XML based format for the archival of relational databases, in order to adapt it to the characteristics of the dimensional model used in data warehouses.

The SIARD format proved to be the most appropriate starting point for this representation given the inherent modularity of data warehouses, with independent stars sharing some dimensions. SIARD has a segmented structure of directories and files, unlike DBML [18] (Database Markup Language) presented at RODA, which represents everything in a single file, impairing the handling of data.

Thus, reusing the effort to define an archive format that stores the definition of the tables and their data, it is proposed to add a metadata layer for data interpretation according to the data warehouse perspective. So, given the simplicity of the dimensional model in terms of relationships between tables, it becomes possible to analyze the archived data with greater efficiency through simplified queries applied directly on the XML files using XQuery[3] and XPath[4].

3 Relational Database Preservation with SIARD

The Swiss Federal Archives (SFA) have developed an open storage format for relational databases called SIARD[5] (Software Independent Archiving of Relational Databases), as well as a set of conversion tools named the SIARD Suite [24], in order to convert relational databases (e.g., Access, Oracle and SQL Server) into the archival SIARD format, edit the SIARD format and reactivate an archived database, restoring from the SIARD Format to a database.

[3] `http://www.w3.org/TR/xquery`
[4] `http://www.w3.org/TR/xpath`
[5] Official site: `http://www.bar.admin.ch`

The SIARD format is a nonproprietary and published open standard, based on open standard (e.g., ISO norms Unicode, XML, SQL1999) and the industry standard ZIP. In May 2008, the European PLANETS project accepted SIARD format as the official format for archiving relational databases [6].

The SIARD format is a ZIP64 [25] uncompressed package based on an organizational system of folders, storing the metadata in the `header` folder and table data in the `content` folder. This organization is shown in figure 2.

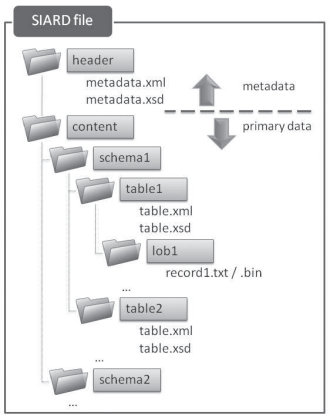

Fig. 2. Structure of the SIARD Archive File

For database's metadata characterization a single XML file is used that contains the entire structure of the database (schemas, tables, attributes, keys, views, functions...) and the corresponding XSD[6] schema for XML validation.

As to the primary data, each schema is stored in different folders and sequentially numbered, as well as the tables of each schema. The data from each table is stored in an XML file with simplified structure (only rows and columns) and its XSD. If there are Large Objects - LOB (BLOB - Binary Large Objects and CLOB - Character Large Objects), these data are stored in binary files or text, within a folder for each attribute of these types, being referred to its path in the respective XML of the table.

3.1 SIARD Suite

The SIARD project produced a set of tools - SIARD Suite[7] [24] - comprised of three components: the `SiardEdit`, a graphical user interface for migration and metadata processing; the `SiardFromDb`, a command line application for extracting and storing a database generating the SIARD file; and the `SiardToDb`, a command line application to reactivate a database from a SIARD file.

[6] `http://www.w3.org/XML/Schema`

[7] This application was gently sent by Johannes Bader from SIARD project

4 DWXML definition

Regarding the SIARD format extension for archiving data warehouses, the proposed XML bridges the gap to describe the dimensional model, adding a metadata file (`dw.xml`) and its schema definition (`dw.xsd`[8]). The figure 3 shows an excerpt of the extended SIARD format, bearing the description of a data warehouse.

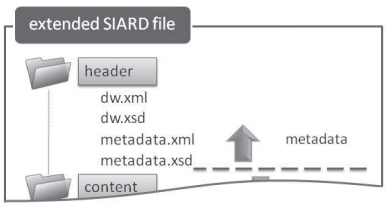

Fig. 3. DWXML added to the SIARD Archive File

This study characterizes the data warehouse as a set of stars and a set of dimensions, represented in tables and views organized in schemas. It is also envisaged a representation of data marts. The figure 4 caracterizes the DWXML basic structure and the `star` element.

The schemas, tables and views follow a similar representation to the SIARD format and are replicated in this description to permit the characterization of a data warehouse regardless of whether there is or not a package SIARD. However, this DWXML version does not contemplate the representation of the primary data in XML, since it is used in conjunction with the format SIARD, which already performs the primary data migration to XML format.

The attribute `version` represents the version of the DWXML definition. The `dwBinding` element supports the description of the DWXML file, the information related to the owner of the data, the credentials of the connection to the data warehouse and the names and versions of the applications involved in the DWXML creation, including the DBMS where the data warehouse was working.

4.1 Stars and Facts

A star is composed of a fact table and a set of rays which establish relationships to dimensions and possibly bridge tables. The `factTable` element references the respective table description in the `schemas` element, it indicates the columns responsible for the joins between fact tables and bridge tables or dimensions, it contains information about its granularity and about the facts. With respect to

[8] `https://www.fe.up.pt/si/wikis_paginas_geral.paginas_view?pct_pagina=42633`

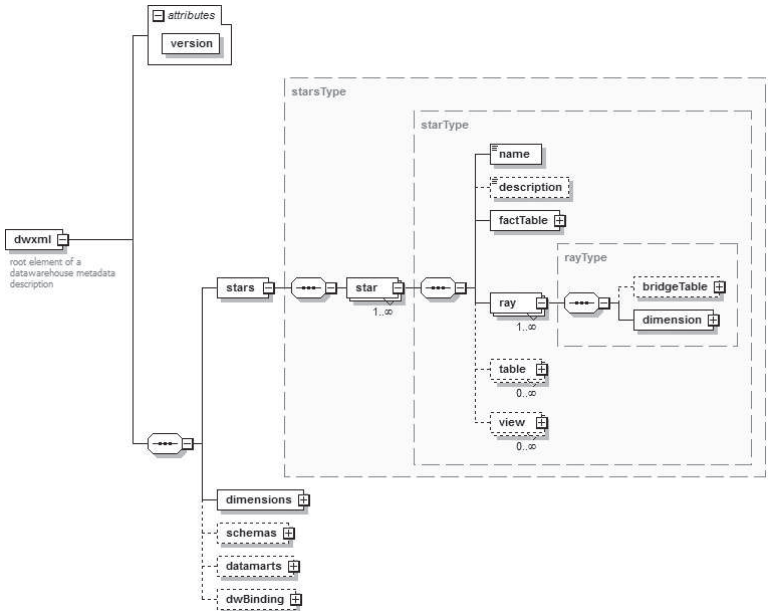

Fig. 4. DWXML schema showing the star element

the facts, they indicate the table column that represents them, as well as their measure type: non-additive, semi-additive or additive.

In a star, each `ray` element represents a relationship between the fact table and the dimension. If there is a many to many relationship between the fact table and the dimension table, it could be added up a bridge table. In this case, the `ray` element would be compose by a `bridgeTable` element that references the related table, followed by the `dimension` element that represents a reference to the dimension.

Example of a DWXML star definition

```
<?xml version="1.0" encoding="UTF-8"?>
<dwxml version="1.0" xsi:noNamespaceSchemaLocation="dw.xsd"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <stars>
    <star>
      <name>IPDW_ANSWERS_STAR</name>
      <description>Star related to the answers</description>
      <factTable>
        <schema>CALDEIAS</schema>
        <name>IPDW_ANSWERS</name>
        <facts>
          <fact>
            <name>ANSWER</name>
            <column>ANSWER</column>
            <measure>ADDITIVE</measure>
          </fact>
        </facts>
      </factTable>
      <ray>
        <dimension>
```

```
          <schema>CALDEIAS</schema>
          <name>IPDW_QUESTION</name>
        </dimension>
      </ray>
      <ray>
        ...
      </ray>
    </star>
  </stars>
  ...
</dwxml>
```

4.2 Dimensions

A key step in the process of the data warehouse creation is to declare the dimensions [20], so that the data dictionary [26] contains this metadata and enables its future extraction. It eases the process of identifying the dimensions, levels and hierarchies, as well as tables and views that support them. The figure 5 displays the `dimensions` element schema.

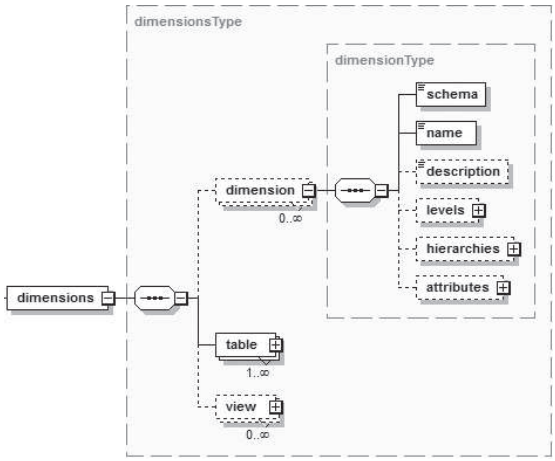

Fig. 5. The dimensions element schema

The metadata related to the dimensions is stored in separated `dimension` elements and allows the categorization and description of the facts and measures in order to support meaningful answers to the requested questions. Each `dimension` element describes the levels and respective level keys, the level hierarchies and the attributes defined by each level. The `tables` and `views` elements contain the reference to the tables and views described in the `schemas` element.

5 Application Architecture

The DBPreserve Suite, the application that supports the data warehouse migration process to the proposed preservation format, has the following general

requirements: to get the metadata describing the data warehouse, to integrate the component `SiardFromDb` that migrates the data warehouse to the SIARD format, to generate the DWXML and add it to the generated SIARD file and must have a graphical interface that helps the migration process and allows editing and retrieving of metadata by querying the primary data in XML format.

This application is composed by 5 major modules as shown in the overall architecture of the application in figure 6 and it has been developed using the NetBeans IDE 7.0 RC1 and Netbeans Platform[9], with support for Java 1.7[10], using the JDOM[11] library [27] for XML processing. The DBPreserve Suite has been tested in a case study that uses a data warehouse built on Oracle Database 11g Enterprise Edition Release 11.1.0.7.0 - 64bit Production[12].

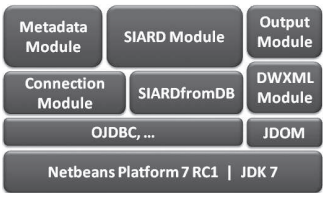

Fig. 6. DBPreserve Suite general architecture

The metadata extraction needed to complete the DWXML is done using a module that requests the metadata from the data dictionary [26] of the data warehouse. Through the analysis of the acquired metadata, a significant part of the metadata is automatically filled in, directly or by inference. Nevertheless, it is still necessary some manual input of small metadata details, such as objects' descriptions.

The SIARD Suite component that builds the SIARD format is integrated into the DBPreserve Suite via a thread responsible for the process that manages the execution of the command `SiardFromDb` [24], as well as the log of the migration execution. At this stage, the object to migrate is the relational database implementation of a data warehouse.

For the SIARD format encapsulation, the SIARD Suite uses a proprietary format to create the uncompressed ZIP64, that extends the ZIP format to overcome the 4 GB size limit in the standard ZIP. However, the access and the integration of DWXML into the SIARD format is performed using the Java 1.7 `java.util.zip` library which already supports ZIP64 format extensions defined by the PKWARE ZIP File Format Specification [25].

[9] `http://netbeans.org/features/platform/`
[10] `http://download.java.net/jdk7/docs/api/`
[11] `http://www.jdom.org/index.html`
[12] `http://www.oracle.com/us/products/database/enterprise-edition-066483.html`

The DWXML generation is performed by a Java representation of an XML document using JDOM [27]. JDOM has a straightforward, fast and lightweight API, optimized for Java programming.

The output module enables the access and display of the XML archived data throughout the data warehouse perspective and allows star level queries, using XQuery and XPath.

6 Conclusions and Future Work

This study resulted in a proposed file format for long-term preservation of data warehouses. The DWXML presented allows the characterization of the data warehouse metadata and seamlessly extends the SIARD format for this kind of databases. The developed application allows the control over the process of migrating the data warehouse and associated metadata to XML, according to DWXML and SIARD Format, as well as adding and editing associated metadata. Since this is an XML archive from a dimensional model, with simplified relationships, it is possible to query and extract the stored data with higher performance rather than using an XML from a relational model. As future work, there is the intention of untying the application from the SIARD Suite that makes the migration of primary data in the SIARD format with heavy costs in terms of time consumption, testing the performance improvements introduced by Java 1.7 and the use of JDOM in the XML processing. Another contribute to the enrichment of this application can be the reactivation of the data warehouse in a DBMS, in order to restore the data warehouse from the XML based archive format described.

References

1. C. J. Date. An Introduction to Database Systems (Eight Edition). Pearson, Addison Wesley, 2004.
2. Arif Ur Rahman, Gabriel David, Cristina Ribeiro. Model Migration Approach for Database Preservation. In The Role of Digital Libraries in a Time of Global Change, 12th International Conference on Asia-Pacific Digital Libraries, ICADL 2010, Gold Coast, Australia, pages 81-90. Springer Berlin / Heidelberg, 2010.
3. WorldWideWeb Consortium. Extensible Markup Language (XML) 1.0 (fifth edition) W3C Recommendation, November 2008.
4. W. H. Inmon. Building the Data Warehouse. JohnWiley and Sons, New York, 1992.
5. Ralph Kimball and Margy Ross. 2002. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (2nd ed.). John Wiley & Sons, Inc., NY, USA.
6. Swiss Federal Archives SFA Unit Innovation and Preservation. Siard Format Description. Technical Report, Federal Department of Home Affairs FDHA, Berne, 2008.
7. Miguel Ferreira. Introdução à Preservação Digital - Conceitos, estratégias e actuais consensos. Escola de Engenharia da Universidade do Minho, 2006.

8. Kenneth Thibodeau. Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years. In The State of Digital Preservation: An International Perspective. Documentation Abstracts, Inc. - Institutes for Information Science, 2002.
9. Consultative Committee for Space Data Systems. Reference Model for an Open Archival Information System (OAIS) - Blue Book. Washington: National Aeronautics and Space Administration, 2002.
10. Margaret Hedstrom, Clifford Lampe. Emulation vs. Migration: Do users care? RLG DigiNews, 5 Num 6, 2001.
11. Authenticity Task Force. Requirements for Assessing and Maintaining the Authenticity of Electronic Records. Technical report, InterPARES Project, Vancouver, Canada, 2002.
12. Carl Lagoze, Sandy Payette, Edwin Shin, ChrisWilper. Fedora: An Architecture for Complex Objects and their Relationships. International Journal on Digital Libraries, Vol. 6 Num. 2:124138, 2006.
13. Jeffrey van der Hoeven. Emulation for Digital Preservation in Practice: The Results. The International Journal of Digital Curation, Issue 2, Volume 2:123132, 2007.
14. Eld Zierau, Caroline van Wijk. The PLANETS Approach to Migration Tools. In IS&T Archiving 2008, Bern, Switzerland, 2008. Society for Imaging Science and Tech.
15. Angela Dappert, Adam Farquhar. Implementing Metadata that Guides Digital Preservation Services. In iPress2009, San Francisco, California, 5-6 October 2009.
16. Pauline Sinclair. The Digital Divide: Assessing Organizations' Preparations for Digital Preservation. PLANETS White Paper, March 2010.
17. Stefan Brandl, Peter Keller-Marxer. Long-term Archiving of Relational Databases with Chronos. In First International Workshop on Database Preservation - PresDB'07, 23 March 2007.
18. José Carlos Ramalho, Miguel Ferreira, Luís Faria, Rui Castro. Relational Database Preservation through XML Modelling. In Extreme Markup Languages 2007, 2007.
19. PLANETS: Tools and Services for Digital Preservation. PLANETS Product Sheet, 2009.
20. Oracle Database SQL Reference 10g Release 1 (10.1), Part Number B10759-01, `http://www.stanford.edu/dept/itss/docs/oracle/10g/server.101/b10759/statements_5006.htm`
21. Douglas Hackney. 1997. Understanding and Implementing Successful Data Marts. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
22. Wolfgang Hummer, Andreas Bauer, and Gunnar Harde. 2003. XCube: XML for Data Warehouses. In Proceedings of the 6th ACM International Workshop on Data Warehousing and OLAP (DOLAP '03). ACM, New York, NY, USA, 33-40. DOI=10.1145/956060.956067, http://doi.acm.org/10.1145/956060.956067
23. Jaroslav Pokorny. 2002. XML Data Warehouse: Modelling and Querying. In Proceedings of the Baltic Conference, BalticDB&IS 2002 - Vol.1, Hele-Mai Haav and Ahto Kalja (Eds.), Vol.1. Inst. of Cybernetics at Tallin Technical University 267-280.
24. Hartwig Thomas, Swiss Federal Archives SFA Unit Innovation and Preservation. SIARD Suite Manual. Federal Department of Home Affairs FDHA, Berne, 2009.
25. PKWARE Inc., .ZIP File Format Specification, Version: 6.3.2, Revised: September 28, 2007, `http://www.pkware.com/documents/casestudies/APPNOTE.TXT`
26. Oracle, Oracle9i Database Concepts Release 2 (9.2) - The Data Dictionary, `http://download.oracle.com/docs/cd/B1050101/server.920/a96524/c05dicti.htm`
27. Jason Hunter. JDOM in the Real World - JDOM makes XML Manipulation in Java Easier than Ever. Oracle Magazine, September/October 2002.

Appendix B

DBPreserve Suite GUI

The DBPreserve Suite application has several graphical interfaces which enable the access to the features of the application, helping the user in the migration process, metadata editing, schemas viewing and primary data browsing.

B.1 Application environment

Figure B.1 shows the desktop application with no project open. It is the first interface that the user gets. It has a menu bar to access all the actions of the application, such as to create a new archive project, open an archive project, build the SIARD format, create the DWXML and embed it into SIARD format, browsing the primary data and change preferences and options of the application. An archive project is the process that manages the overall migration process, from collecting the metadata from the data warehouse until the creation of the SIARD format and embeds the DWXML in it.

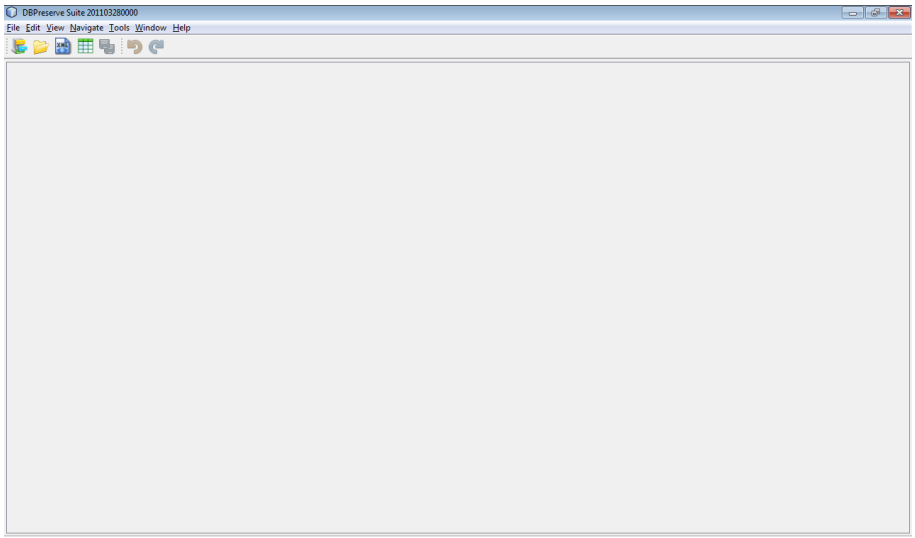

Figure B.1: Archive project initial interface

B.1.1 The toolbar

The toolbar has direct buttons which allows the access to some of the common actions of the DBPreserve Suite application. These functions are described below, ordered from left to right:

- New archive project (CTRL+N) - allows the creation of a new migration process to collect the metadata from a data warehouse to build an extended SIARD format with the relational and the dimensional model metadata, along with the primary data as well.
- Open an archive project (CTRL+O) - enables the user to browse the computer to the folder where an archive project is stored and opens it.
- Create the DWXML file (CTRL+D) - creates the DWXML file upon the imported data warehouse metadata and embed it into the SIARD format file.
- Extract data from XML (CTRL+E) - enables the browsing on the primary data, retrieving the data from the XML-based files.
- Save (CTRL+S) - saves all the changes to the metadata. No changes are allowed to the primary data.
- Undo (CTRL+Z) - undo the last changes.
- Redo (CTRL+R) - restores the last undo action.

B.1.2 Creating or opening an archive project

Figures B.2 and B.3 illustrates the interfaces to create a new archive project. The user should name the project, select the destination folder, add its description and setup the connection to the data warehouse.

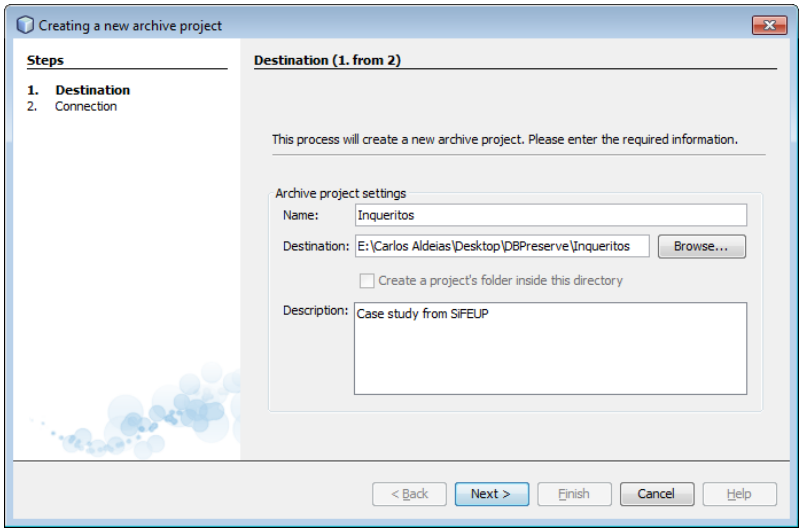

Figure B.2: New archive project interface - Destination

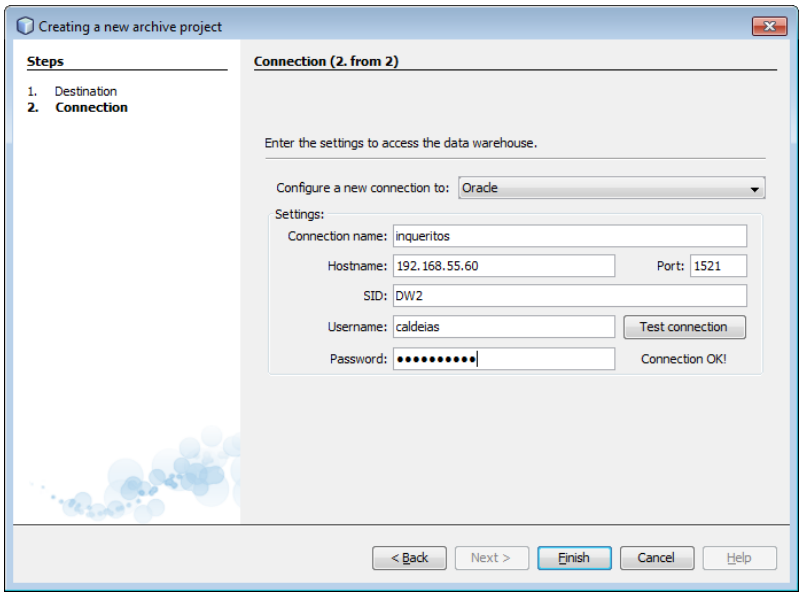

Figure B.3: New archive project interface - Connection

To open an archive project already created, the user should press the *Open an archive project* button in the toolbar and browse the computer to the folder where it is stored.

B.2 Archive project interface

After creating or opening an archive project, the user get the *Archive Project* interface which has a tabular panel representing the workflow of the migration project: at the begging the user sets or verifies the settings on the *Archive project settings* panel, then proceeds to the SIARD format creation on the *SIARD file generation* panel, after that the *DWXML proposal* panel allows the building a DWXML from the imported metadata and the *DWXML editing* panel allows the editing of the metadata.

B.2.1 Archive project setting panel

Figure B.4 displays the *Archive project settings* panel and the *Connection* explorer window where will be displayed the data tables according to their role in the dimensional model.

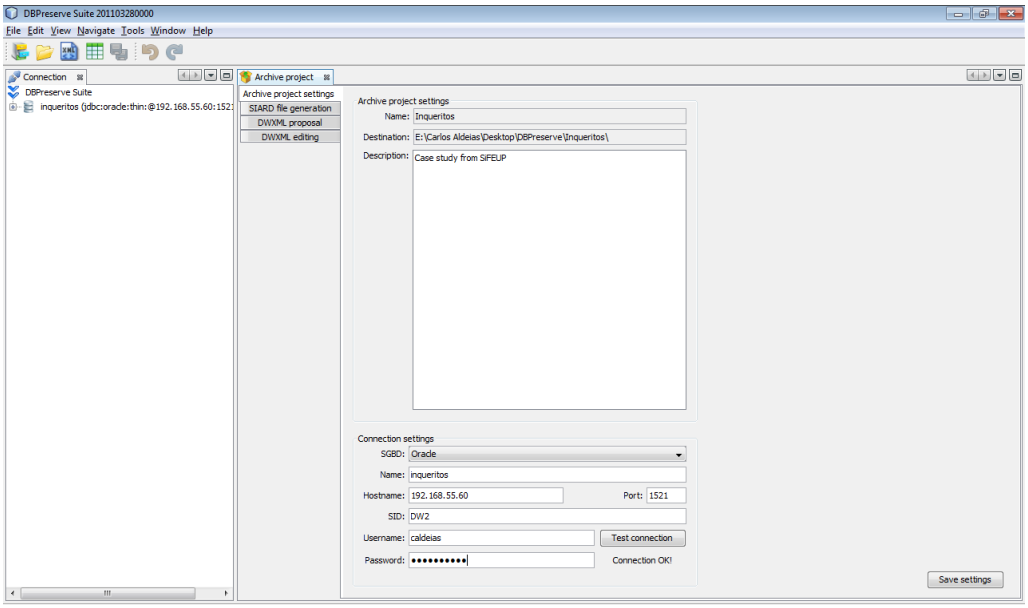

Figure B.4: Archive project interface - Settings panel

B.2.2 SIARD file generation panel

Figure B.5 represents the interface which integrates the `SIARDfromDB` tool for creating the SIARD format uncompressed ZIP64 file. The user can specify if this application builds only the metadata, leaving the primary data empty, or builds the metadata and the primary data as well. The log is shown to verify the progress of this migration.

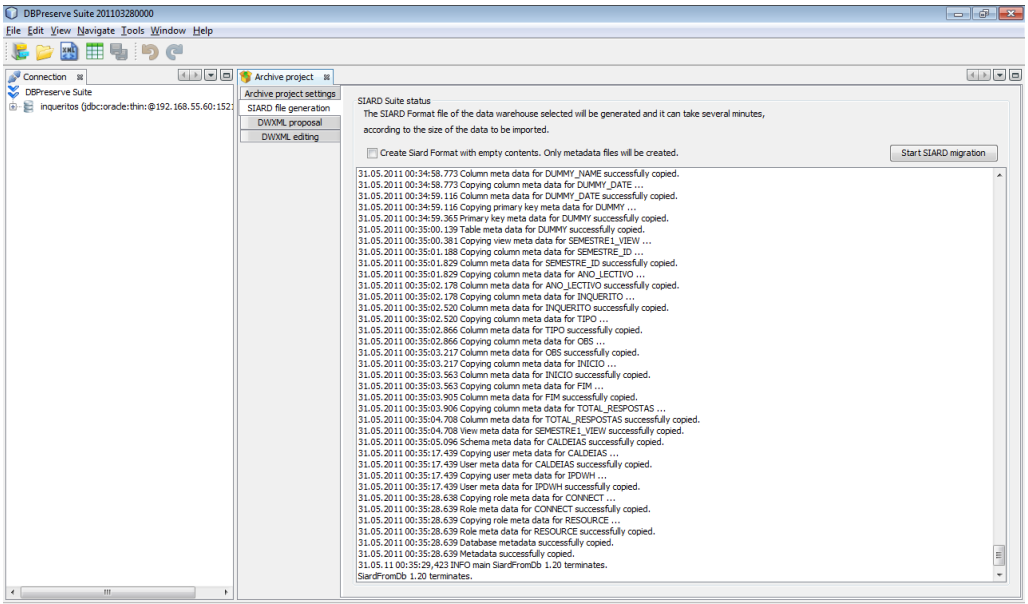

Figure B.5: Archive project interface - SIARD file generation panel

B.2.3 DWXML Proposal panel and Connection explorer

Figure B.6 shows the *DWXML proposal* panel. The user can specify which metadata objects to import, or if he wants to open an already built DWXML file or even get it from the extended SIARD format, if already present. The left side of figure displays the *Connection* explorer window. If the user chooses to import the metadata using the connection to the data warehouse, the tables will be sorted and hanging over the nodes according to their role in the dimensional model. If a table has an incorrect role, the user can reassign a new role just by dragging and dropping that table over the correct role node.

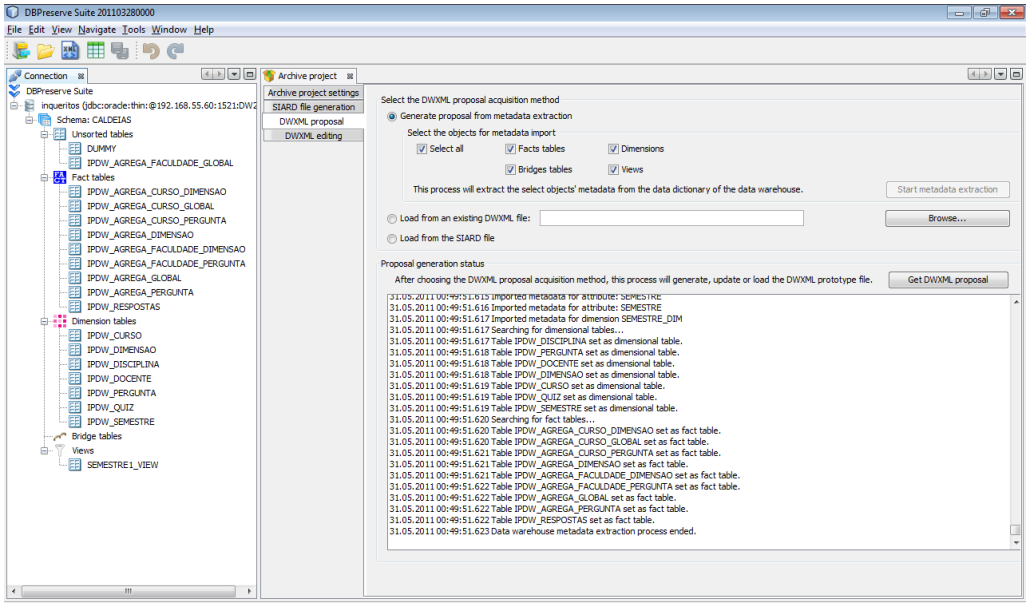

Figure B.6: Archive project interface - DWXML Proposal panel

When all the tables are assigned to their correct roles, the user can generate a DWXML proposal, showing the log to verify task progress. This will create a temporary DWXML memory representation and will trigger the opening of a *DWXML explorer* window and a *Diagram* window, as illustrated in Figure B.9.

B.2.4 DWXML editing panel

The migration of the metadata from the data warehouse does not do all the work in filling the DWXML description needed. So, the user gets several changing panels which are displayed under the *DWXML editing* panel, and shown according with the object selected in the *DWXML explorer* window, allowing the editing and completing of the metadata for full model description. Figure B.7 presents one of those interfaces, one of the most complexes, that allows the editing of a dimension, its levels, attributes and hierarchies.

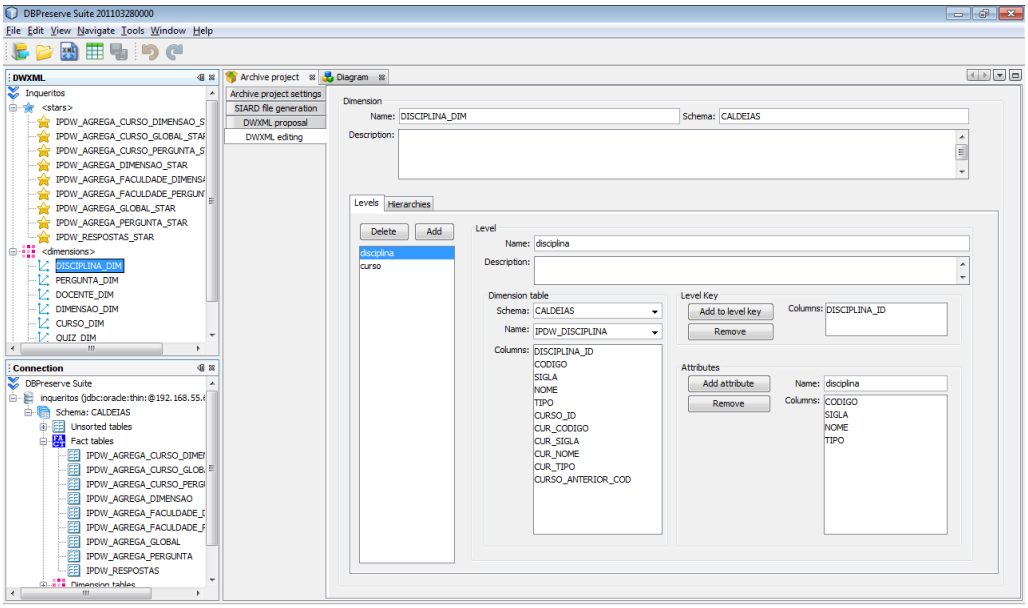

Figure B.7: Archive project interface - DWXML editing panel

B.3 DWXML file viewer window

When the metadata editing is finished, the user ends the SIARD format extension process by pressing the *Create the DWXML file* button in the toolbar. The application will generate the DWXML file (`dw.xml`) and will embed it into the `header` folder of the SIARD format file generated, as well its correspondent XSD schema (`dw.xsd`) for XML validation. The DWXML is shown in the *DWXML viewer* window (Figure B.8).

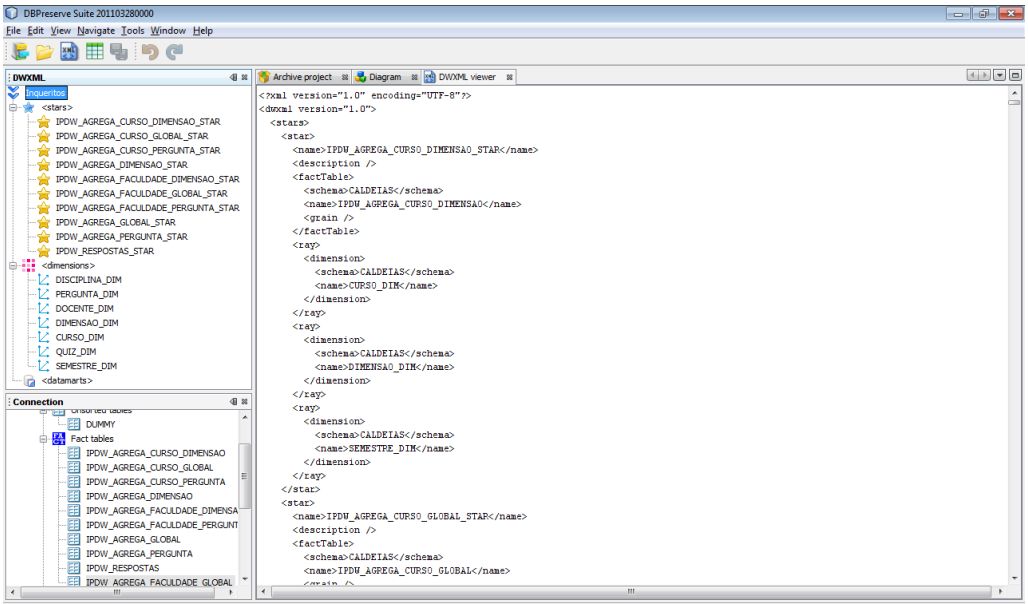

Figure B.8: DWXML file viewer window

B.4 DWXML explorer and Diagram window

The *DWXML explorer* window represents the XML nodes by showing the Star or Snowflake schemas and the dimensions. When the user selects a star or a dimension, the correspondent diagram is shown. This allows the user to get a rapid overview of the dimensional model. Figure B.9 displays the selected star schema, reflecting the fact and dimension tables that support it.

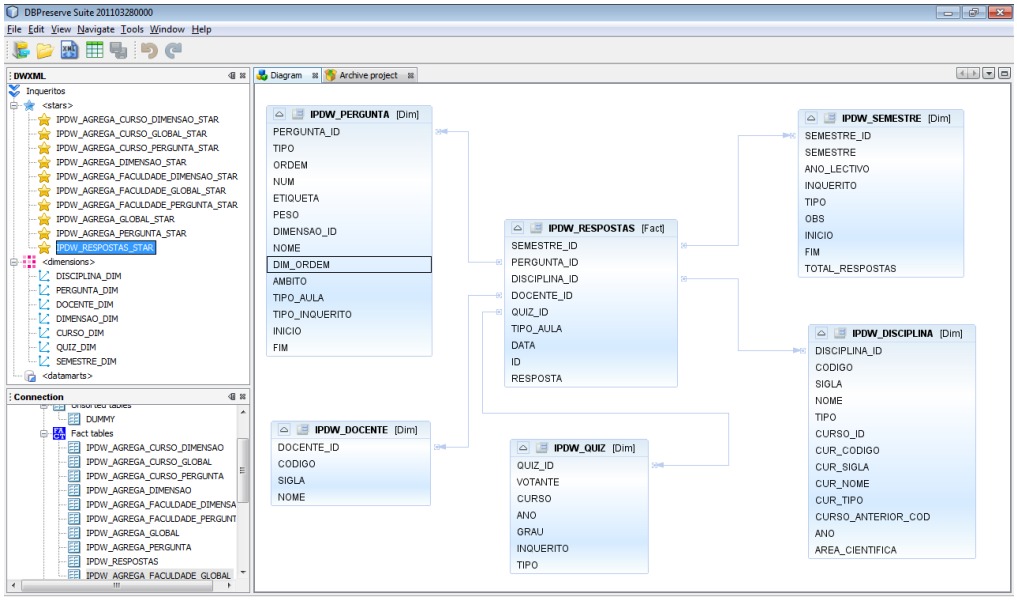

Figure B.9: DWXML explorer and Star Schema diagram window

Figure B.10 displays the levels and hierarchies of the selected dimension.

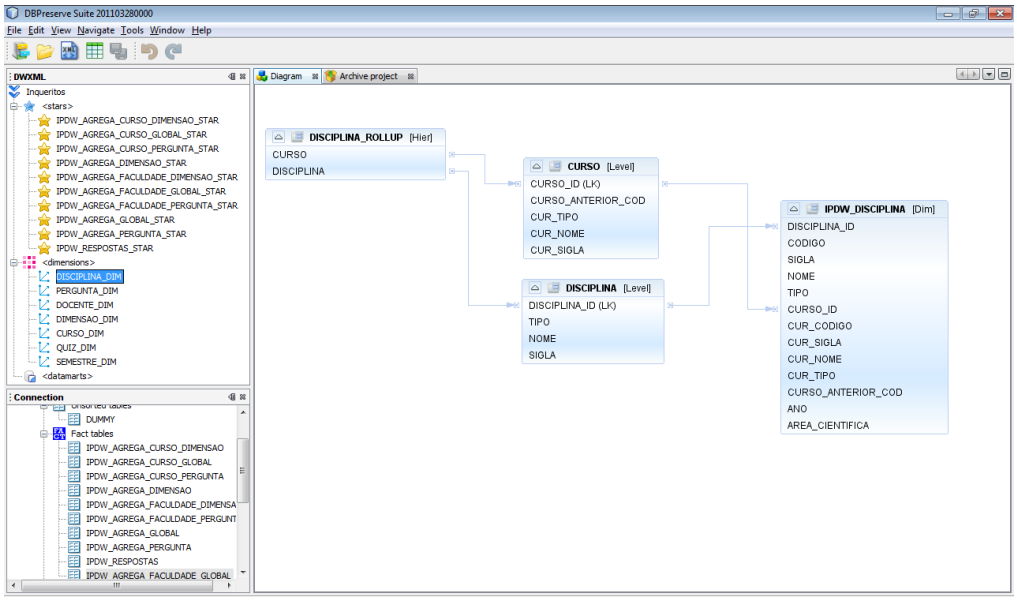

Figure B.10: Dimension representation diagram window

B.5 Data viewer window

In order to verify the access to the primary data in XML-based files, an interface was created which enables the reading of the primary data in the SIARD format and displays it for the user. Figure B.11 shows the primary data according to the object selected in the *DWXML explorer* window.

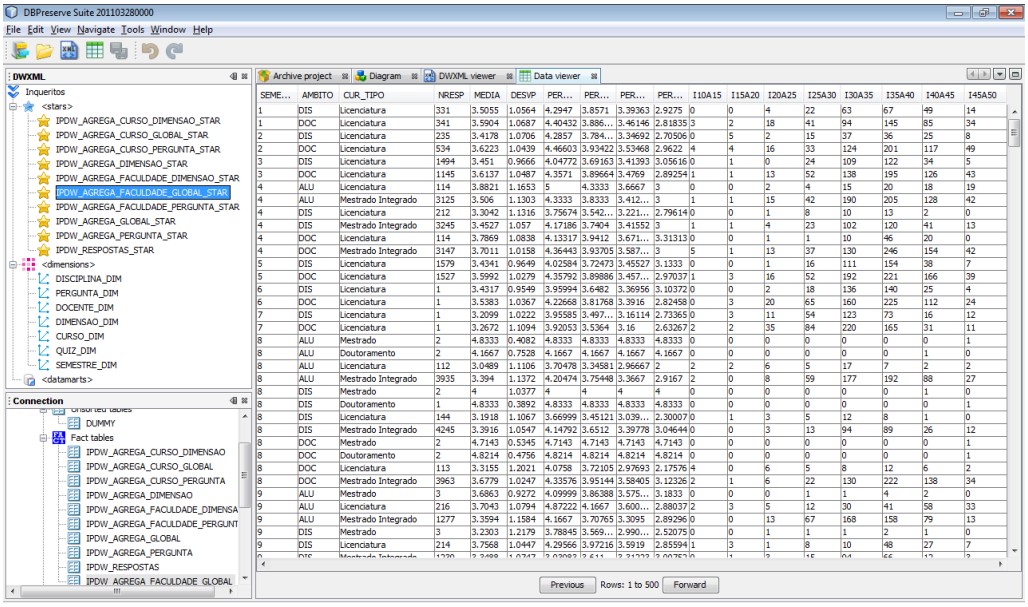

Figure B.11: Data viewer window

B.6 Preferences window

Figure B.12 illustrates the *Preferences window*. This *Preferences* option can be found at the *Tools* menu. Here, the user has to indicate the path to the SIARD Suite installation, in order to integrate that tool within the DBPreserve Suite application and enables the SIARD format generation.

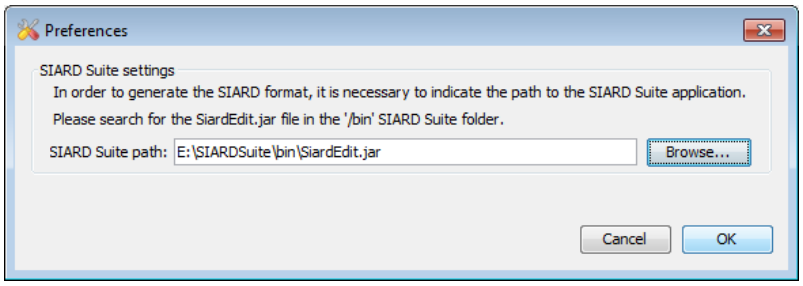

Figure B.12: Preferences window