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.

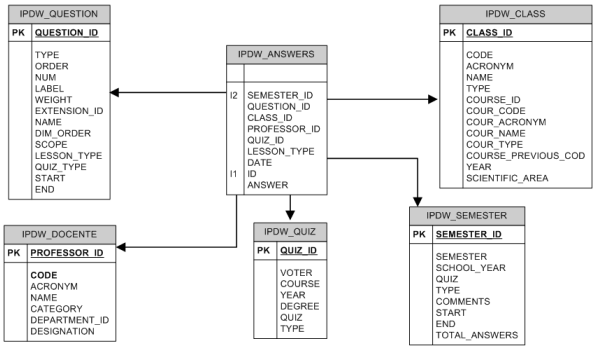

Fig. 1. Star schema example

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.

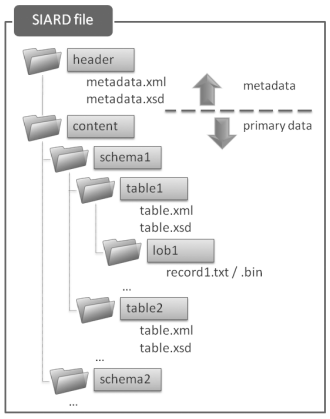

Fig. 2. Structure of the SIARD Archive File

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.

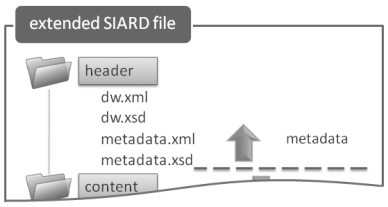

Fig. 3. DWXML added to the SIARD Archive File

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`

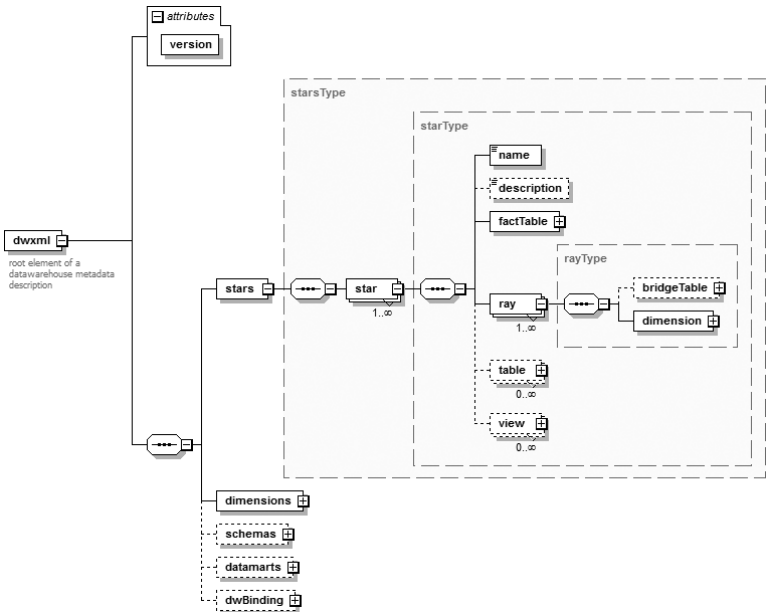

Fig. 4. DWXML schema showing the star element

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.

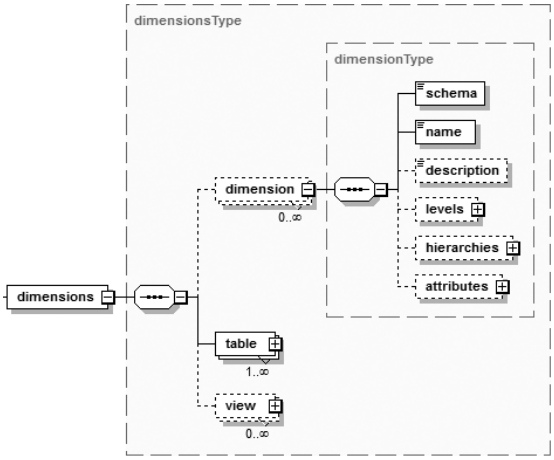

Fig. 5. The dimensions element schema

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].

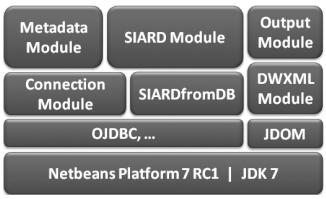

Fig. 6. DBPreserve Suite general architecture

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.