Model Migration Approach for Database Preservation*

Arif Ur Rahman[1,2], Gabriel David[1,2], and Cristina Ribeiro[1,2]

[1] Departamento de Engenharia Informática–Faculdade de Engenharia, Universidade do Porto
[2] INESC Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
{badwanpk,gtd,mcr}@fe.up.pt

Abstract. Strategies developed for database preservation in the past include technology preservation, migration, emulation and the use of a universal virtual computer. In this paper we present a new concept of "Model Migration for Database Preservation". Our proposed approach involves two major activities. First, migrating the database model from conventional relational model to dimensional model and second, calculating the information embedded in code and preserving it instead of preserving the code required to calculate it. This will affect the originality of the database but improve two other characteristics: the information considered relevant is kept in a simple and easier to understand format and the systematic process to preserve the dimensional model is independent of the DBMS details and application logic.

Keywords: database preservation, dimensional modeling.

1 Introduction

Organizations are increasingly relying on databases as the main component of their recordkeeping systems. However, at the same pace as the amount and detail of information contained in such systems grows, also grows the concern that in a few years most of it may be lost, when the current hardware, operating systems, database management systems (DBMS) and actual applications become obsolete and turn the data repositories unreadable. The paperless office increases the risk of losing significant chunks of organizational memory. In this paper we present an approach for preserving the information stored in relational databases for the future.

According to research in the area, the five characteristics of databases which must be preserved are context, structure, content, appearance and behaviour [15,17]. The **context** includes non-technical information giving answers to questions like who, when and why about the database as well as information on its technical features. The **contents** is the data stored in the database representing real-world facts. The **structure** of the database relates to the composition and logical hierarchy of the elements of a database, thus contributing to the meaning

* This work is supported by FCT grant reference number SFRH/BD/45731/2008.

G. Chowdhury, C. Khoo, and J. Hunter (Eds.): ICADL 2010, LNCS 6102, pp. 81–90, 2010.

of the data. **Appearance** is about screen forms used for entering and modifying data and about generated reports. It requires the presence of the user application designed to manipulate data, submit queries, and extract information. The **behavior** is the dynamic part of the system and, therefore, the most difficult to preserve. It includes the interaction control component and the code implementing the business rules. If the former can be seen as less relevant from a preservation viewpoint, the latter may contain important bits of information, in the form of functions to produce important derived results not explicitly stored in the database. This paper will discuss how the migration from relational to dimensional model impacts the preservation of these database characteristics.

Some aspects of preservation which need to be taken care of during the process of database preservation include integrity, intelligibility, authenticity, originality and accessibility [5]. **Integrity** refers to the completeness, correctness and consistency of data stored in the database. **Intelligibility** of a database concerns both the interpretation of the data formats and the understandability of the relationships between tables and their relation to the reality they represent. An intricate database model becomes hard to understand. The **authenticity** is the property which relates the preserved information with its source and is guaranteed by keeping record of the actors, tools and operations involved in a preservation process. **Originality** in terms of preserving the structure and functionality, should be kept into account but may conflict with other aspects like intelligibility or accessibility. Technical **accessibility** means that the data is kept in open formats and does not rely on vendor-specific software.

The approach to database preservation proposed in Section 3 is based on model migration. This operation changes the structure of the database in order to improve intelligibility and accessibility, the crucial problems identified above. In the process, we might decide to perform a data quality assessment and repair the data to reduce problems like missing values, inserting records on foreign key errors. However, the decision was to keep the data as it is, in order to preserve as much as possible the facts recorded, though in different format, even when they are affected by data quality problems. So, the goal is to preserve the actual level of integrity. The authenticity of the actual database is guaranteed by the inclusion of audit information qualifying the records. The authenticity of the preserved database requires the addition of audit information relating the preserved records to the original ones and the specification of the migration procedures, when they were executed and by whom, using which tools. Authenticity also benefits from metadata about the context of creation and use of the original database which should be recorded in the context component of the preserved database. Note however that the model migration approach is done at the expense of originality.

2 What to Preserve

The model migration approach provides a pre-processing of the database, which can be coupled with existing database preservation initiatives such as the

Software Independent Archiving of Relational Databases (SIARD) [4,14] or the Digital Preservation Testbed (DPT)[15]. SIARD is a non-proprietary published open standard. It is based on other open standards like Unicode, XML, SQL 1999 and the industry standard ZIP [14]. As it is based on open standards, it supports interoperability of the database contents in the long term.

Using the SIARD format, even if the database software through which the database was created is not available or not executable, the database will be accessible and usable. At the present it is possible to migrate Oracle, Microsoft SQL Server and Microsoft Access databases to SIARD format. A database in the SIARD format consists of two components namely metadata and the primary data. An uncompressed ZIP archive stores these components having the metadata in the folder header and the primary data in the folder content. Moreover, the archive also stores metadata about which primary data can be found where in the archive [14]. A SIARD database archive can be reloaded in the future to any RDBMS which supports standard SQL [5].

The model migration proposal concentrates on the archival format for the database contents. The archive should contain the original relational model and it may contain the original database file or an export file. The archive must also contain the new preservation model and the preserved contents according to the new model following the SIARD archive structure.

A similar approach could be applied to the DPT, modifying its central notion of preservation object. The DPT preservation object has five main components, namely the original database, an XML overview file of the database, applications, the preservation log file and metadata. Testbed suggests the preservation of the original database file (*.mdb or export file). The XML overview file represents an overview of the tables in the database, the relationships between the tables and the content and structure of the actual tables and views. The application component is for storage of queries, stored procedures, application code (if applicable), system documentation and user manuals. It is not meant to preserve the applications as a working entity. The preservation log file contains all the information about the preservation actions through which the database passes. The metadata component contains metadata for the authentic preservation. This is mainly contextual metadata.

For using the DPT with our proposal, the overview file has to contain the original relational model and the new model and also the original database file version possibly in XML or SQL DDL format.

It is clear from the research done that there can be no single way to preserve all kinds of databases. For our work we define a database as a combination of four components.

1. Data: The data is the contents stored in the tables of a database.
2. Schema: The schema of a database is the structure (data model) which is needed to understand the relationships among tables. Business rules which are partly structural also need to be preserved.
3. Context: The contextual information which is normally not included in the operational system.

4. Database Application: The application developed in a high-level programming language for retrieval, modification, and deletion of data in conjunction with various data-processing operations. This contains the appearance and behavioural aspects of the database. A part of the business rules maybe implemented in the behavioural aspect.

For preserving a database it is important to take into account the nature of its contents. The contents of some databases evolve with the passage of time e.g. the CIA World Factbook database [1,2] while others remain static as is the case of a population census database. The former needs a different approach than the latter for preservation. In this paper the focus will be on the latter.

It is very important to preserve the schema of a database for the understandability and usability of the information stored in it. In our approach as we suggest to migrate the database to a dimensional model, the structure of the preserved database is different from the structure of the operational system but it is easier to understand.

A database application can be preserved as a working entity by writing an emulator for it. An emulator is a program that runs on one computer and virtually re-creates a different computer. Therefore, through emulation we can use obsolete application on a recent computer [11]. But there are many problems associated with using emulation as a preservation strategy. For example it cannot be ensured that the computers in the future will be capable of executing an emulator of any older computer. Every time there is some change in the platform for which the emulator was developed, the emulator needs to be re-developed. Another approach to deal with the application component is to simply preserve the user manuals, queries and functions in textual format and not as a working entity [15]. In this paper we propose an alternative approach which calculates and explicitly stores the information embedded in the code (application logic). The goal is to keep just the data and make the information application-independent.

3 Database Migration for Database Preservation

Database migration is not a new concept, it has been studied and discussed in the past [8,9]. Database migration may take different forms including DBMS version evolution (Oracle 10 to Oracle 11), change in DBMS (Oracle to DB2) or change of the data model (hierarchal to relational data model). In this paper we propose the use of model migration from the relational model to the dimensional model as a step for preserving a relational database.

3.1 Dimensional Modeling

Dimensional modeling is a logical design technique that seeks to present the data in a standard framework which is intuitive, allows for high-performance access and is resilient to change [3,6]. Information is stored in tables of two natures: dimensions store detailed data about the entities or objects involved in a certain

relevant process (like clients, items being sold, employees); fact tables store the values representing real world facts (like quantities sold, amounts earned) and the relationship to the corresponding dimensions. A fact table surrounded by the related dimensions is called a star. Dimensions may be shared by different stars. Time and Location are commonly used dimensions.

The strengths of the dimensional model make it better for long-term preservation and access to the information. As discussed by Kimball [7] and Ponniah [10], report writers, query tools and user interfaces can all make strong assumptions about the dimensional model which makes the processing more efficient. Other features, as discussed by Torlone [16], include explicit separation of structure and contents, and hierarchies in the dimensions. The separation of the structure and contents helps in making it DBMS independent which is crucial for database preservation. The hierarchies in dimensions help in aggregating the data and result in faster access. In the past dimensional modeling has not been considered for database preservation.

3.2 Model Migration

The design of a database preservation process requires a proper balance of the aspects of originality, integrity, accessibility, intelligibility and authenticity in each major problem to be solved. Two issues to be considered are the complexity of the relational model of real-size information systems and the embedding in code of important knowledge from the application domain.

The complexity of the relational model may prove to be a serious stumbling block for preserving databases. Part of it comes from the requirement of redundancy elimination that transaction-oriented databases must follow in order to be efficient and consistent in capturing facts.

The preserved database is no longer used for transaction processing but instead for querying and decision making. Although it contains the facts of the original database, the change of usage brings a change of requirements. It is better if the data is preserved in a form that gives simpler and quicker access. This can be achieved by migrating the database from relational model to a dimensional model, as depicted in Figure 1 [12]. The operation will affect the originality of the database but will give a relief from the complexities of the relational model and improve intelligibility and accessibility, because the resulting model is much easier to understand and the queries on it are simpler to state.

The second problem is the fact that some results coming from the database are produced by functions embodying application-domain knowledge. Preserving code is a much more difficult problem than preserving data, because it requires the ability to preserve the engine able to run it, from the application to the DBMS or the underlying operating system. But discarding the code affects accessibility, as there is no technical way to reach the derived data it would be producing, and it affects also the integrity as chunks of the data are lost. The solution offered by migration is to include the facts and dimension attributes in the dimensional model to explicitly store the data in danger. In the data migration phase, also called ETL for extraction, transformation and loading according

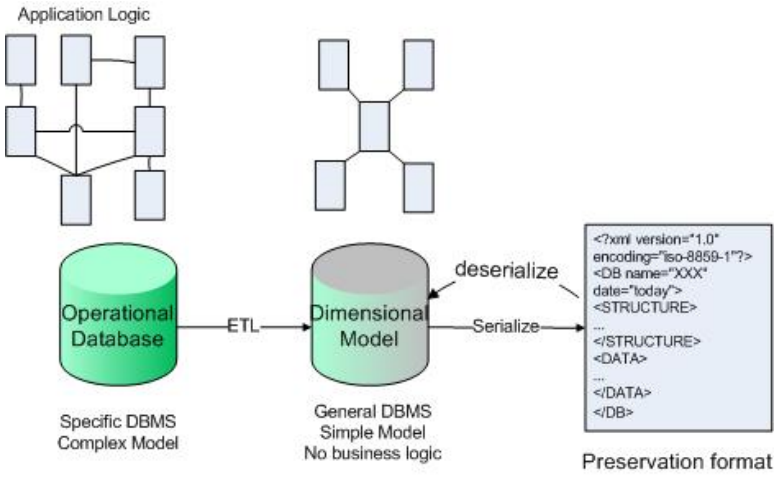

Fig. 1. Model Migration Approach for Database Preservation

to data warehouses terminology [7], the application code is run to produce the implicit values in it, which are then kept in the preserved database. It is assumed that when the preservation operation is performed, the original platform or a compatible one is still available.

In the migration process, parts of the relational model which are needed only to support the interaction at the data capture phase or which are not relevant may be dropped.

To further simplify the information stored in the database, it can be converted to XML format. This will make the information platform-independent which is very important for achieving its long-term preservation [13,14].

4 Case Study

The proof of concept for the ideas presented in this paper is a case study involving the database for the "Course Evaluation System" of the University of Porto. Students are invited to answer 31 questions about the course they are attending and teacher performance, with answers ranking from 1-5, with 1 the lowest and 5 the highest grade. Information about the identity of the student is not stored and the answers are anonymous. The operational system has a rather complex model, a part of which is shown in Figure 2. It is designed to capture the answers via dynamically built on-line forms. All the reports are calculated at query time using functions based on complex queries.

A report can be about the whole faculty, a program, a curricular year or a single course. The user may also choose a granularity level for a report which may be one of the following:

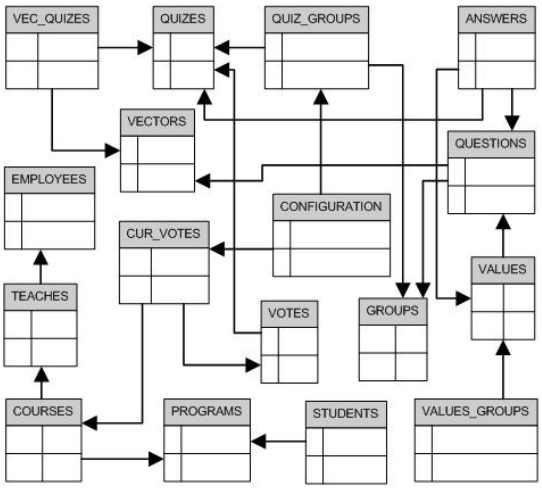

Fig. 2. Relational Model of the Operational System

- Question level: the report presents statistics about each individual question.
- Vector level: the questions are grouped into vectors, the report presents statistics about the vectors.
- Global level: the vectors are combined into a single result, the report presents statistics about the global results.

Before starting the migration process a thorough analysis of the operational system was done. The process was carried out in small steps which resulted in the dimensional model shown in Figure 3. The tables COURSES and PROGRAMS in the operational system are represented by the IPDW_COURSES dimension in the dimensional model which has two levels (courses and programs). The tables QUESTIONS and VECTORS are represented by the IPDW_QUESTIONS dimension also with two levels (questions and their aggregations in vectors). The questionnaire was modified six times since the inception of the system. The tables QUIZZES and QUIZ_GROUPS in the operational system store information about these different quizzes. The IPDW_SEMESTER dimension stores information about a semester and the QUIZ used for it. Though the answers of the students are kept anonymous, some information related to them is stored in the IPDW_QUIZ dimension in the dimensional model. IPDW_ANSWERS is the fact table, it is the de-normalized form of the ANSWERS and VALUES table. Reference keys to the corresponding dimensions along with the values from ANSWERS and VALUES tables are stored in it.

Tables GROUPS, VALUES_GROUPS, CONFIGURATION, and QUESTIONNAIRE_GROUPS were discarded as they were used for dynamically building the interface for the online data capture. In this process it was important not to damage the integrity of the data.

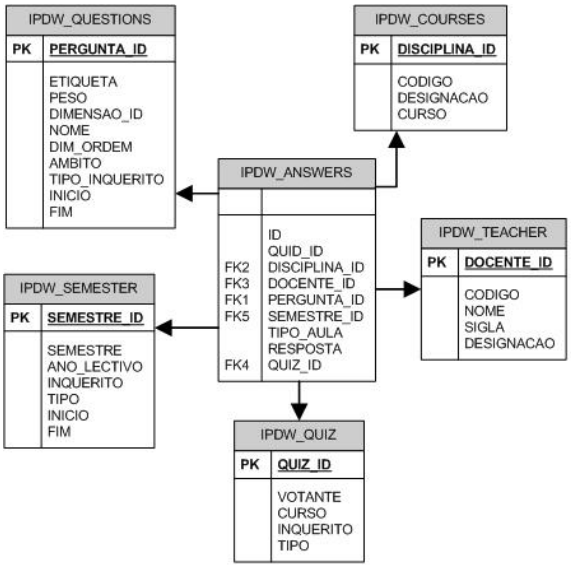

Fig. 3. Dimensional Model for the Operational System

In the next step the functions were executed and the results like averages, standard deviation, number of answers and percentiles for each granularity level were explicitly stored. For each granularity level we got a different star with the fact table storing the results coming from executing the functions and references to the corresponding dimensions. One of the stars is shown in Figure 3. The dimensions in this star are shared by others. At this stage the database became application-independent.

The dimensions in the dimensional model are systematic and easy to serialize and store in XML along with their structure and metadata.

If we compare the models (Figure 2 and Figure 3) it is obvious that the dimensional model is simpler and easier to understand and therefore more intelligible. The information which was embedded in code is now explicitly stored in the database and is readily accessible. Also there is no need to preserve the code for the future. After the process of migration is completed, the results coming from the migrated database were compared with the operational system to verify the authenticity of the information.

5 Conclusion

This paper proposes a model migration approach for database preservation. For migrating the relational model of the operational system a thorough understanding of the original system is required. Before migration it should be decided what is to be kept for the future and what can be discarded. This work is similar to the

evaluation, elimination and description work an archivist must perform before archiving a set of documents.

6 Future Work

This is a work in progress and we are currently engaged in making the process of migration easier. As the proposed approach involves migration of an operational system from a relational model to a dimensional model, we are working on defining some generic transformation rules for the process. The rules will guide the team involved in a migration process.

Another aspect that requires further research is metadata. A specification of the metadata needs, both for keeping the database system context and for describing the preservation process is still required.

References

1. Buneman, P., Cheney, J., Tan, W.-C., Vansummeren, S.: Curated databases. In: PODS 2008: Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, New York, NY, USA, pp. 1–12 (2008)
2. Buneman, P., Müller, H., Rusbridge, C.: Curating the CIA world factbook. International Journal of Digital Curation 4(3) (2009)
3. Connolly, T.M., Begg, C.: Database Systems: A Practical Approach to Design, Implementation, and Management. Addison-Wesley Longman Publishing Co., Inc., Boston (2001)
4. Heuscher, S.: Technical aspects of SIARD. ERPANET (2003)
5. Heuscher, S., Järmann, S., Keller-Marxer, P., Möhle, F.: Providing authentic long-term archival access to complex relational data. In: Ensuring Long-Term Preservation and Adding Value to Scientific and Technical Data. European Space Agency (2004)
6. Imhoff, C., Galemmo, N., Geiger, J.G.: Mastering Data Warehouse Design: Relational and Dimensional Techniques. Joe Wikert (2003)
7. Kimball, R., Reeves, L., Thornthwaite, W., Ross, M., Thornwaite, W.: The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses. John Wiley & Sons, Inc., New York (1998)
8. Meier, A.: Providing database migration tools - a practicioner's approach. In: VLDB 1995: Proceedings of the 21st International Conference on Very Large Databases, pp. 635–641. Morgan Kaufmann Publishers Inc, San Francisco (1995)
9. Meier, A., Dippold, R., Mercerat, J., Muriset, A., Untersinger, J.-C., Eckerlin, R., Ferrara, F.: Hierarchical to relational database migration. IEEE Softw. 11(3), 21–27 (1994)
10. Ponniah, P.: Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals. John Wiley & Sons, Inc., Chichester (2001)
11. Dutch Archives Testbed Project. Digital preservation testbed white paper emulation: Context and current status. Technical report, Dutch National Archives, Digital Preservation Testbed Project, ICTU, Nieuwe Duinweg 24-26, 2587 AD Den Haag, (June 2003)

12. Rahman, A.U., David, G., Ribeiro, C.: Model migration approach for database preservation. In: 5th International Digital Curation Conference, London, December 2-4 (2009)
13. Ramalho, J.C., Ferreira, M., Faria, L., Castro, R.: Relational database preservation through xml modeling. In: Extreme Markup Languages, Montreal Quebec, Department of Informatics, University of Minho, Portugal (August 2007)
14. SFA. SIARD format description. Technical report, Swiss Federal Archives, Berne (September 2008)
15. Digital Preservation Testbed. From digital volatility to digital permanence: Preserving databases. Technical report, National Library of Australia, Dutch National Archives (2003)
16. Torlone, R.: Conceptual multidimensional models. In: Multidimensional Databases, pp. 69–90. Idea Group, USA (2003)
17. Wilson, A.: Significant properties report. Technical report, InSPECT (April 2007)