Swiss Federal Archive: Long-term Archiving of Structured Data

The Swiss Federal Archives (BAR) are the "memory of the federal administration". According to the federal law on archiving (BGA, SR 152.1), the BAR's purpose is to secure, permanently store, develop, maintain the usability and provide availability of the permanently valuable files of the federal administration. According to the BGA, the BAR's archiving task is independent of the type of information medium of the documents.

Since 1982, the federal archives have also been archiving data digitally and today administers around six terabytes of digital archive data. This amount will increase by another nine terabytes in 2003 while growth of 20 terabytes per year is expected in the medium-term. The specialist department ARELDA (archiving of electronic digital data and documents) performs such archiving as well as supporting and consulting the information safeguarding teams of the federal archives and the departments of the federal administration. As a part of the E-government project ARELDA, the specialist department ARELDA also designs and realizes long-term solutions for long-term archiving of digital files. ARELDA is one of five key projects in the federal state's E-government strategy.

Schweizerisches Bundesarchiv
Archives fédérales suisses
Archivio federale svizzero
Archiv federal svizzer

Customer
Swiss Federal Archive

Industry
Public administrations

Project
SIARD (Software Invariant Archiving of Relational Databases)

Topics
Long-term archiving of relational databases

The solutions of the federal archives are based on the principle of "application-invariant archiving" in which the files are taken from their creation-specific original environments (software, hardware, storage, data and file formats; a certain loss of information and authenticity must be accepted) during the archiving process and transferred into open, standardized, generic and fully documented archive environments. There they are permanently retained in long migration cycles (at least 15 years per cycle). On principle, functionality (i.e. software, hardware) is not archived.

Nevertheless, digital archiving remains labor- and cost-intensive due to periodic demand for conversion and migration. The duration of the migration cycles and the "archiving data quality" (e.g. minute adherence to the archiving format specifications and the quality of the technical metadata) constitute the fundamental success factors which determine whether the digital archive can be financed and retained in the long-term.

The continuous takeover of large and heterogeneous amounts of digital documents therefore represents a great challenge. The federal archives places special importance on the archiving of databases. Their "application-invariant archiving" is extremely complex and labor-intensive since data often has to be taken from a multitude of different manufacturer-specific

(proprietary) database products owned by federal agencies. The SIARD project (Software Invariant Archiving of Relational Databases) was aimed at developing cost-saving and quality-enhancing solutions in this field.

What challenges had to be mastered?

Long-term archiving of databases in the federal archives has four main goals: The data to archive, stemming from different source systems, needs to (1) be kept as authentic (i.e. close to the original in form and content) as possible, (2) be "storeable" as long as possible (i.e. ten to 20 years) without permanent maintenance and migration effort, (3) remain understandable and intelligible with regard to their content and their context of meaning, creation and usage, (4) be usable anytime as conveniently as possible by the customers of the federal archives. Obviously, these targets conflict with one another significantly.

In the early phases of the project, the data was archived in "flat files", i.e. semantically and syntactically virtually unstructured pure text files with unlinked individual tables. This method resulted in the loss of logical links between the individual tables and views. Additional information, as well as explanations of key terms, code tables, record lengths, check constraints, etc., had to be detailed in accompanying paper documents with great effort, was prone to error and sometimes contradictory and incomplete. One of the main problems was the lack of unified standardization for separators, numeric number formats, etc. used in the "flat files". Databases archived under these conditions not only lack authenticity, they also permit only very laborious and limited use by the customers of the federal archives.

This project had three main goals: ensuring **enhanced data integrity**, i.e. documenting those metadata (keyword lists, check constraints, code tables, record lengths, measure units, etc.) necessary for understanding the data in a manner that avoids erroneous, missing and contradictory information. If possible, these technical metadata should not constitute a "foreign" supplement to the data, but already represent an integral part of the data structure encoding within the archive format. Another goal was to achieve a better standardization and an **increased efficiency of the process to deliver data from the operational system (which produced it) to the federal archives**. This includes automating the data export and documentation processes by controlling them in a standardized workflow. The third goal was to create **better and more flexible usability**: After a database has been archived (and thus made independent of specific database software), it should be reloaded into any relational database management system with an acceptable amount of manual effort (ideally automated).

What was realized?

One pre-requisite for solving these problems was the use of SQL-3 (ISO/IEC 9075) as an archive format for describing the logical database structure. The SQL format is open and fully documented, not manufacturer-specific yet database-related and widely supported by the industry. "Pure" (i.e. generic) SQL-3 had to be used, precisely conforming to the ISO standard, to allow a unique archive format and consistent migrations into new archive formats in the future.

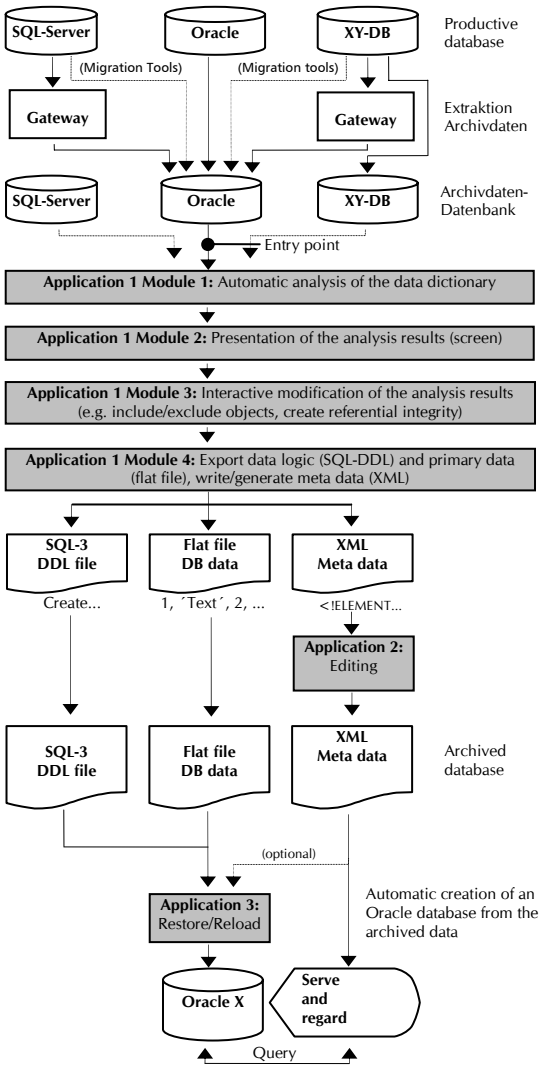
This requirement caused severe problems as most database system manufacturers do not comply fully with the ISO standard, i.e. supplement their products with manufacturer-specific features (e.g. proprietary data types). Plans called for the use of (i) SQL-3 for encoding the data logic (database structure), (ii) "flat files" for the actual primary data per database table and (iii) XML for encoding descriptive context metadata usually not stored in a database.

Before realizing the target solutions, the severity of the expected difficulties and losses of information and authenticity due to manufacturer-specific SQL "flavors" was examined by means of (1) an initial feasibility analysis and (2) the subsequent development of a software prototype.

The results of the feasibility analysis of the proposed solution were favorable. The prototype phase was constrained to the creation of the software-independent database archives and their reloading into Oracle8*i*. In addition, an extensive analysis document revealed the deviations of the Oracle SQL syntax from the ISO SQL3 standard. This document was later expanded with a focus on Microsoft SQL Server.

During the realization phase, the framework interface for the data dictionary analysis was extended to cover both Oracle and Microsoft SQL Server. In addition to these two "expert modes" for Oracle and Microsoft SQL Server, an additional "generic mode" provides a product-independent implementation of this interface with the help of the JDBC metadata API (database interface for the programming language Java). This mode has a reduced archiving scope as compared with the expert modes but fundamentally allows archiving from any database product for which a JDBC API is available (incl. Microsoft Access).

Recording the metadata: the application for describing the original context of meaning, creation and usage of the database was implemented, as well as the opportunity to integrate external documents (e.g. user manuals or development documentation in TIFF or PDF format) into the database archive. Additionally, the functionality of the reload tool was greatly enhanced. The interplay of the three resulting software tools is illustrated in the following figure:



Application 1: Archiving

Analysis of the database structure and its graphical representation as a tree structure in a GUI. Possible integrity violations and deviations from ISO standard SQL-3 are presented to the user. Various problems can be fixed by automatic and/or manual exclusion, modification and creation of objects. (All modifications are recorded in an activity log for traceability). It is now possible to create a database archive.

Application 2: Editing

High-level documentation of the created database archive: Entering additional information in predefined metadata fields (XML), e.g. clear text resolution of keywords , code lists, free text descriptions for table content, and additional information such as the delivering agency or the non-archiveable application logic (stored procedures). The final database archive cannot be built until all mandatory fields are filled in.

Application 3: Reloading

Reloading a database archive into an Oracle instance, providing standard query options for the primary data, incl. any SQL queries. All technical (low-level) metadata as well as the high-level descriptive metadata captured in application 2 is automatically displayed (XML). It is possible to reload several database archives into one instance and span queries over all of them. .

Technological highlights

All files that constitute a database archive are pure text files (UTF-16/Unicode). All XML files have dedicated XML schemes and thus can be validated. Various transformation tasks are performed using XSLT. As the applications have to run on various platforms (Windows, Linux, Solaris), they were developed using the programming language Java. The Java-propriety class libraries (Swing) were used for the GUI. Database connections are via JDBC. In addition to the expert modes for Oracle and MS SQL Server, further expert modes for archiving from other database products can be linked dynamically (without code changes).

What is the advantage for customers?

Compared to previous methods, the solution provides long-term archiving of data from various manufacturer-specific databases in a much more authentic, effective, comprehensive, and consistent way. Furthermore, the new archive format (pure ISO standard SQL-3) is standardized, fully documented and manufacturer-independent. This enables long-term 'shelf life' and durability of database archives at no great expense, as well as future migrations to new archive formats in the future. This significantly reduces long-term costs for continuous maintenance (i.e. conversions and migrations). Within the archival environment, migration cycles of at least 10 - 20 years are likely needed. These can be performed efficiently.

The solution enhances data integrity and data quality during archiving, i.e. erroneous, missing and contradictory information can be minimized. This solution automates and unifies the transfer of database data from the source systems of the federal agencies to the federal archives to the greatest extent possible, thus making it more cost-efficient.

This solution considerably eases and improves the usability of archived databases for the customers of the federal archives.

Systems

> Solaris 7 / 8
> Red Hat Linux from version 7
> Windows NT / 2000 / XP

Databases

> Oracle 7 / 8 / 9
> Microsoft SQL Server 7 / 2000
> Microsoft Access

Development tools and libraries

> Eclipse Platform 2.0
> J2SDK (Java 2 Software Development Kit) 1.4, incl. Swing
> JDBC (Java Database Connection) 3.0
> AXP (Java™ API for XML Processing) 1.2

Glossary

Code tables	Abbreviations (codes) and their descriptions
Data dictionary	All database metadata
Framework	Application framework, specific implementations are invoked from within the framework
GUI	Graphical User Interface
Keyword lists	Keyword tables
Class library (Swing)	Function library for GUIs
Context metadata	High-level metadata describing the context of meaning, creation and usage of the original database data
Objects	Database objects (schemas, tables, columns, ...)
Queries	Description of data query
SQL DDL	Structured Query Language / Data Definition Language DDL is part of SQL
Syntax	(Programming) language declaration
Technical metadata	Low-level metadata from the data dictionary describing the technical structure of the database as well as the individual database objects
Value range	Min./max. value
XSLT	eXtensible Stylesheet Language for Transformation – declaration language for transforming XML files into other formats